



Project Number: 693349

Standard and protocols for visual materials

**Giuseppe Airò Farulla, Pietro Braione, Marco Indaco,
Mauro Pezzè, Paolo Prinetto**

CINI, Aster SpA

Version 1.2 – 15/11/2017

Lead contractor: Universitat Pompeu Fabra
Contact person: Josep Quer Departament de Traducció i Ciències del Llenguatge Roc Boronat, 138 08018 Barcelona Spain Tel. +34-93-542-11-36 Fax. +34-93-542-16-17 E-mail: josep.quer@upf.edu
Work package: 3
Affected tasks: 3.5

Nature of deliverable¹	R	P	D	O
Dissemination level²	PU	PP	RE	CO

¹ R: Report, P: Prototype, D: Demonstrator, O: Other

² **PU:** public, **PP:** Restricted to other programme participants (including the commission services), **RE** Restricted to a group specified by the consortium (including the Commission services), **CO** Confidential, only for members of the consortium (Including the Commission services)

COPYRIGHT

© COPYRIGHT SIGN-HUB Consortium consisting of:

- UNIVERSITAT POMPEU FABRA Spain
- UNIVERSITA' DEGLI STUDI DI MILANO-BICOCCA Italy
- UNIVERSITEIT VAN AMSTERDAM Netherlands
- BOGAZICI UNIVERSITESI Turkey
- CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE France
- UNIVERSITE PARIS DIDEROT - PARIS 7 France
- TEL AVIV UNIVERSITY Israel
- GEORG-AUGUST-UNIVERSITAET GÖTTINGEN Germany
- UNIVERSITA CA' FOSCARI VENEZIA Italy
- CONSORZIO INTERUNIVERSITARIO NAZIONALE PER L'INFORMATICA Italy

CONFIDENTIALITY NOTE

THIS DOCUMENT MAY NOT BE COPIED, REPRODUCED, OR MODIFIED IN WHOLE OR IN PART FOR ANY PURPOSE WITHOUT WRITTEN PERMISSION FROM THE SIGN-HUB CONSORTIUM. IN ADDITION TO SUCH WRITTEN PERMISSION TO COPY, REPRODUCE, OR MODIFY THIS DOCUMENT IN WHOLE OR PART, AN ACKNOWLEDGMENT OF THE AUTHORS OF THE DOCUMENT AND ALL APPLICABLE PORTIONS OF THE COPYRIGHT NOTICE MUST BE CLEARLY REFERENCED

ALL RIGHTS RESERVED.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 693349.

History of changes

VERSION	DATE	CHANGE	REVIEWER(S)
1.0	17/05/2017	Initial version.	Paolo Prinetto, Josep Quer, Jordina Sánchez Amat
1.1	25/07/2017	Added Section 5.c; Typos fixing; Adapted to the project's template.	Paolo Prinetto, Josep Quer, Jordina Sánchez Amat
1.2	15/11/2017	<p>Document partially revised w.r.t. its original structure. In particular, hereby follows the most important changes made:</p> <ul style="list-style-type: none"> • Removed overlap with deliverable D3.2; • Added a description over the issues and possible solutions related to archiving and streaming SIGN-HUB products, linked to what is already stated within deliverable D3.2; • Section 2 has been revised: old sections 2.b and 2.c have been removed, while new section 2.a now deals with annotation formats specifically of interest for the needs of the SIGN-HUB project; • Section 3 has been deepened to now be more centered on the needs of the project, to cover the requirements for formats suitable for preservation and formats suitable for distribution; • Removed too generalist content in old Section 5. 	Pietro Braione, Mauro Pezzè, Jordina Sánchez Amat

INDEX

1. Introduction	5
2. Video Metadata and Metadata Processing	6
a. Representation of Video Metadata	6
b. Text Data for Video Processing	7
i. Video Annotations.....	8
ii. Video Subtitles.....	11
3. Analysis on Video Coding Format	13
a. Codec: Compression/Decompression	15
i. MPEG Standards	15
ii. H.264	15
iii. H.265.....	16
iv. Motion JPEG 2000	16
v. AV1	17
vi. FFV1	17
vii. WebM.....	18
viii. Ogg Theora	18
ix. Xvid.....	19
x. Windows Media Video and RealVideo	19
xi. Libavcodec	19
xii. Discussion on the codecs	20
b. Video Container Format	21
xiii. AVI	21
xiv. MP4.....	21
xv. QuickTime	21
xvi. FLV	22
xvii. RealMedia	22
xviii. Matroska	22
xix. Ogg	22
xx. Discussion on Video Containers	23
c. Lossy Video Compression	25
d. Lossless Video Compression	28
e. Representing and Encoding 3D Graphics Assets	31
i. MPEG-4 Animation Framework eXtension (AFX)	31
ii. COLLADA	32
iii. X3D.....	32
iv. Considerations	32
4. Guidelines to Produce and Publish High-Quality Video Content	33
a. Naming Conventions	34
b. Content Uploading on Online Digital Platforms for Researchers	36
c. Recommendations for the SIGN-HUB project	39
5. Conclusion	43
Bibliography	44

1. Introduction

This deliverable report includes the results of activities which have been undertaken within the WP3 *Digital Infrastructures*. The main objective of the WP3 is the development of the web platform that will host the contents to be produced by WP2 and WP4. Specifically, this document is connected to task T3.5 (i.e., *Standardization and management of video material and multimedia content*).

The goal of the deliverable is to give an overview of existing standards to represent, analyze, store and retrieve digital media content (specifically videos) and to give specifications to the content providers for WP2 to allow the homogeneous and scalable storage, findability, and accessibility of the content within the SIGN-HUB digital platform. As a result of the study, a list of standards, conventions, protocols, formats and software has been proposed and studied, posing particular emphasis on the requirements that will apply to future contents within the project.

The structure of the deliverable is the following:

- Section 2 the fundamentals standards, conventions and procedures to represent documents and written text, with emphasis to video metadata;
- Section 3 describes the fundamentals technologies that are enforced to represent video multimedia content and virtual avatars;
- Section 4 gives guidelines on how to publish video multimedia content to the web;
- Section 5 briefly concludes the deliverable and summarizes its findings and the guidelines exposed to the members of the SIGN-HUB consortium.

2. Video Metadata and Metadata Processing

a. Representation of Video Metadata

Linguistics analyzes human language as a system for relating sounds (or signs in sign languages) and meaning [1]. As so, they are committed in documenting, describing and analyzing entities, properties, phonemes, and other aspects of languages, semantics, grammars, and many other characteristics of modern, and ancient, languages. Language description is a complex task, usually requiring many years of work in deepening the language concerned, to equip the linguists with enough data to write a reference grammar. Further, the task of documentation requires the linguist to collect a substantial corpus in the language in question. For spoken languages, this may also consist of texts and recordings; on the other hand, video recording, which outcomes can be stored in an accessible format within open repositories and used for further research, is almost the only usable tool when it comes to sign languages [2].

During years of investigation on a given sign language (or a set of languages), it is possible that linguists acquire thousands, or even more, of video contents, which are then to be processed, analyzed, and stored to make them available to the scientific community. However, as video is complex digital data, modern computers and software are typically unable to analyze it automatically and to categorize it properly, thus requiring effort from the linguists (or, in any case, video editors) to put together the video with a set of ordered information useful to retrieve, at a glimpse, the most important information about the video itself. Indeed, in any case, automatic extraction of features and information from digital video content is not of interest for the SIGN-HUB project. Information to be collected within the SIGN-HUB project comprises, for instance, date and time when the video was captured, its length, the actors involved, the sign language under investigation and the particular considerations or conclusions that can be drawn from seeing the video. By means of this extra information, the tasks of ordered video storing, information retrieval and automated analysis is greatly simplified. A video that is accompanied from such extra information is typically referred to as *annotated*, and the extra information is known as *metadata*.

Metadata is all the textual information that accompanies a video and helps search engines and human agents better understand the content. However, in the true sense of the word, metadata is of outmost relevance when it comes to video. As Wikipedia puts it, "metadata is particularly useful in video, where information about its contents (such as transcripts of conversations and text descriptions of its scenes) are not directly understandable by a computer, but where efficient search is desirable". While metadata is often associated with on-page textual information (e.g., titles and descriptions), it can and should be leveraged to provide much deeper information. From a cognitive perspective, it has an important role in learning and instruction. As part of guided noticing it involves highlighting, naming, and labelling and commenting aspects of visual representations to help focus learners' and scientists' attention on specific visual aspects. This is especially important when scientists and experts interpret visualizations in detail and explain their interpretations to others.

Metadata is additional data that provides information about other data. It is often referred to as *bits about bits*. The first typical example one can think of about metadata is represented from the card catalogs in libraries, which provide informative data (e.g., author's name, title, year, physical location in the library) about other data (a book). But metadata is also used to describe digital data. This is even more true in the modern world, where information has become increasingly digital. For instance, a web page may include metadata, i.e., meta tags, with keywords and a brief description about the page. Meta tags are inserted at the very top of a web page and are not displayed to the user who reads the page on a browser. The information is instead read by search engines which use it to classify the web page, or better its content.

Summarizing, metadata is itemized information about a file. It is attached to the file and can include copyright information, author, keywords, file size, media format, people, place, and

even more information [3]. A main purpose of metadata is to facilitate in the discovery of relevant information, more often classified as resource discovery. Metadata also helps organize electronic resources, provide digital identification, and helps support archiving and preservation of the resource.

Having briefly highlighted its importance, this Section deals with techniques, languages, standards, and protocols related to video metadata, with particular emphasis to the cases of video annotation and metadata of importance for linguists.

b. Text Data for Video Processing

Video metadata has great potential. It is today hard to imagine a platform providing the opportunity of accessing video contents without their metadata, while it is widely recommended to any platform delivering media content, and especially video, to manage and keep up-to-date the most appropriate and precise possible set of metadata for its content. Description provided from metadata enables easier and quicker searches among video contents and provide users more in-depth information about the desired video.

This information is in form of a text because it is text that machines can read, parse and process. With the words *text processing*, we typically refer to the process of automate the creation or editing of electronic texts and metadata, as opposed to the same manipulation done manually. Text processing involves automated editing commands to produce and change textual content. Textual content come in standardized character sets which contains also control and formatting characters (e.g., the newline or the tabulation characters), which arrange text. Nevertheless, text processing is a process resulting of manually administered actions (macros, filtering mechanisms, content creation...) which affect most basically the visual characters (or graphemes) within the text rather than these invisible characters, which however are the most useful to define the formatting options [4].

Text processing is performed resorting to a software known as editor, which typically is used also for text visualization. An editor essentially invokes an input stream and directs it to the text processing environment, which is either a command shell or a visual environment. The resulting output is the desired text; this text is also eligible for further text processing in a second moment. Many editors give the opportunity to modify the text while preserving unaltered originals. This opportunity is very important for text processing in fields dealing with sensible data, such as law-related procedures [5].

The document lifecycle represents the comprehensive process a document goes through from beginning to end – from creation and distribution, to storage and ongoing document manipulation and modification. Document manipulation and modification tools and processes play an important role in how documents are revised, edited and reused. Text processing, and especially automated processing, is also performed as a medium for textual analysis. This technique is useful for researchers working in cultural, sociological and linguistic studies, to gather and classify information about the authors, the context and the topics of the texts analyzed [6], [7].

Although metadata can be associated with any digital document (e.g., again, a book or a web page), it is when it accompanies videos that it proves to be the most useful. Metadata for video is particularly helpful because videos do not include keywords nor any description, beside the title, as text, and text is the easiest form of information that any machine can analyze. So, metadata is the keywords that computers and search engines use to search for videos. The standard format for metadata for video is the Exchangeable Information File Format (EXIF). The metadata tags defined in the EXIF standard cover a broad spectrum, ranging from date and time information to camera settings, content description, information about the author, and also some non-text information such as preview thumbnail. However, this particular class of metadata is quite general and not of particular interest for linguistics research and digital humanities. Therefore, although we strongly recommend the usage of metadata to accompany the digital video content for what concerns digital content providing within SIGN-HUB (particularly for Work Package WP2), and advise for the need of identifying as soon as possible a

metadata schema and a standard for metadata representation, we also state that such a choice cannot be taken independently from the choice of which repository to select for ensuring both long-term data persistence of the digital video content to be produced within SIGN-HUB and also data findability, accessibility, reusability, and interoperability (as for the FAIR principles). Thus, the identification of (services for) digital platforms and repositories is needed, and then a negotiation phase with their managers to identify a match between the needs of the project and the features they support.

Considering all this, the discussion on which metadata schema and standard to use that may be of more interest for the SIGN-HUB project (e.g., OLAC or CMDI from the CLARIN project), together with the discussion on which repository to use, have been merged and can be found within deliverable D3.2. Just to mention briefly, for the sake of completeness, it is important to remember that:

- OLAC (the Open Language Archives Community, www.language-archives.org) is an international partnership of institutions and individuals who are creating a worldwide virtual library of language resources by: (i) developing consensus on best current practice for the digital archiving of language resources, and (ii) developing a network of interoperating repositories and services for housing and accessing such resources. The OLAC metadata set is based on the Dublin Core metadata set and uses all fifteen elements defined in that standard. To provide greater precision in resource description, OLAC follows the Dublin Core recommendation (being actually an enriched version of it, specifically targeting Sign Languages) for qualifying elements by means of element refinements or encoding schemes. The qualifiers recommended by DC are applicable across a wide range of resources. However, the language resource community has a number of resource description requirements that are not met by these general standards. In order to meet these needs, members of OLAC have developed community-specific qualifiers, and the community at large has adopted some of them as recommended best practice for language resource description.
- CLARIN's CMDI (Component Metadata, www.clarin.eu/content/component-metadata) provides a framework to describe and reuse metadata blueprints. Description building blocks ("components", which include field definitions) can be grouped into a ready-made description format (a "profile"). Both are stored and shared with other users in the Component Registry to promote reuse. Each metadata record is then expressed as an XML file, including a link to the profile on which it is based. The CMDI approach combines architectural freedom when modeling the metadata with powerful exploration and search possibilities over a broad range of language resources. As of now, there are two supported versions of CLARIN's component metadata framework: CMDI 1.1 and CMDI 1.2. They are not interchangeable, but CMDI 1.1 metadata can easily be converted into CMDI 1.2. Considering the project's needs, it would be advisable to create CMDI 1.2 metadata, enabling some new features such as enhanced description of relations between resources, and documentation on component, element and attribute level.

i. Video Annotations

Annotations are typically used to convey information about a resource or associations between resources. Simple examples include a comment or tag on a single web page or image, or a blog post about a news article.

ELAN (tla.mpi.nl/tools/tla-tools/elan) is a professional tool for the creation of complex annotations on video and audio resources. It has a tier-based data model that supports multi-level, multi-participant annotation of time-based media. It is applied in humanities and social sciences research (e.g., language documentation, sign language, and gesture research) for the purpose of documentation and of qualitative and quantitative analysis. With ELAN, a user can add an unlimited number of annotations to audio and/or video streams. An annotation can be a sentence, word, gloss, comment, translation or a description of any feature observed in

the media. Annotations can be created on multiple layers, called tiers. Tiers can be hierarchically interconnected. An annotation can either be time-aligned to the media or it can refer to other existing annotations. The textual content of annotations is always in Unicode and the transcription is stored in an XML format. ELAN provides several different views on the annotations, each view is connected and synchronized to the media played. Up to 4 video files can be associated with an annotation document. Each video can be integrated in the main document window or displayed in its own resizable window. ELAN delegates media playback to an existing media framework, like Windows Media Player, QuickTime or JMF (Java Media Framework). As a result, a wide variety of audio and video formats is supported and high-performance media playback can be achieved. ELAN is written in the Java programming language and the sources are available under a GPL 3 license. It runs on Windows, Mac OS X and Linux. Its main features include:

- navigate through the media with different step sizes
- easy navigation through existing annotations
- waveform visualization of .wav files
- support for template documents
- input methods for a variety of script systems
- multi-tier regular expression search, within a single document or in a selection of annotation documents
- support for user definable Controlled Vocabularies
- import and export of Shoebox/Toolbox, CHAT, Transcriber (import only), PRAAT, and CSV/tab-delimited text files
- export to interlinear text, HTML, SMIL, and subtitles text
- printing of the annotations.

The ELAN software is ELAN used massively in WP2 for video annotation. Its annotation structure is described here: www.mpi.nl/tools/elan/EAF_Annotation_Format.pdf. Summarizing. This online document describes EAF, the ELAN Annotation Format (also known as the EUDICO Annotation Format), on the basis of the main XML elements and attributes of the EAF schema. EAF is the format used for serializing objects that are part of the Abstract Corpus Model. In addition, some prose descriptions are added explaining how ELAN interprets certain elements and what assumptions are made when reading an EAF file. These annotations are first created as XML structures to make the representation independent of any video format and to make it processable. These descriptions form the additional requirements that have to be met to make a document a valid, editable ELAN document.

Thus, for the purposes of SIGN-HUB project, it is necessary to identify platforms and repositories that support the management and distribution of EAF files and schemas (as CLARIN, for instance, does). For instance, from these annotations within SIGN-HUB it could be possible to create video subtitles for specific presentations. Within SIGN-HUB, the various partners (especially from WP2) are going to adopt different solutions for annotations subtitling. In light of this, this present document can only provide mild indications on the most appropriate standard or software to use as each unit might have its own preferred solution (including highly professional software). What is relevant for SIGN-HUB is that the final format of the file must be compatible with the on-line streaming capabilities offered by the repository that will be chosen for the management and distribution of digital video content. Here the most important base constraints are:

- Annotations on the same tier cannot (time-wise) overlap;
- A mix of alignable and reference annotations on the same tier is not allowed.

Figure 1 shows a (simplified) UML class diagram of the EAF document model (and, since an EAF document is a fairly straightforward serialization of objects in an ELAN transcription, of the ELAN transcription objects). Annotations are contained in Tier objects, a tier can be associated

with a parent tier and there are two distinct types of annotations, alignable and reference annotations.

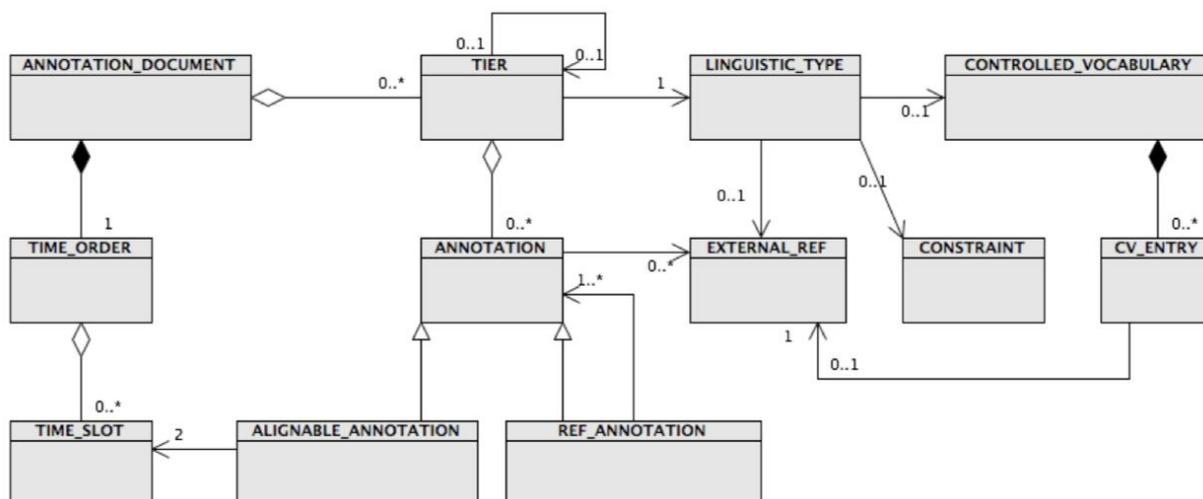


Figure 1 - Simplified UML class diagram of the EAF document model

Apart from ELAN, other standards include the following ones.

Sign Stream (www.bu.edu/asllrp/SignStream/3/) seems offering similar features with respect to ELAN, although through a less intuitive user interface.

EXMARaLDA (Extensible Markup Language for Discourse Annotation, exmaralda.org) is a set of free software tools for creating, managing and analyzing spoken language corpora. It consists of a transcription tool (comparable to tools like Praat or Transcriber), a tool for administering corpus meta data and a tool for doing queries (KWIC searches) on spoken language corpora. EXMARaLDA is used for doing conversation and discourse analysis, dialectology, phonology and research into first and second language acquisition in children and adults. EXMARaLDA is based on the open standards XML and Unicode and programmed in Java.

FOLKER (the FOLK EditoR, exmaralda.org/de/folker-de), which was developed for the project "Research and Teaching Corpus Spoken German" at the Institute of German Language. The data model for this tool represents an adaptation of the EXMARaLDA system, which is specifically tailored to the needs of the FOLK project. FOLKER offers the following functions:

- support of the cGAT conventions for minimal transcripts;
- three different views of the transcript (as a segment, score or post view);
- audio signal in the form of an oscillogram;
- interoperability with other tools (ELAN, Praat, and of course EXMARaLDA);
- in addition, OrthoNormal can also be used for the orthographic normalization of FOLK transcripts in literary transcription.

PRAAT (praat.org) is a free computer software package for the scientific analysis of speech in phonetics. It was designed, and continues to be developed, by Paul Boersma and David Weenink of the University of Amsterdam. It can run on a wide range of operating systems, including various versions of Unix, Linux, Mac and Microsoft Windows (2000, XP, Vista, 7, 8, 10). The program supports speech synthesis, including articulatory synthesis. Basically, Praat is a computer program with which users can analyze, synthesize, and manipulate speech, and create high-quality pictures for your articles and thesis.

It is also worth to mention the specification of the Web Open Annotation Data Model standard managed by the W3C consortium, which describes a structured model and format to enable annotations to be shared and reused across different hardware and software platforms. Common use cases can be modeled in a manner that is simple and convenient, while

at the same time enabling more complex requirements, including linking arbitrary content to a particular data point or to segments of timed multimedia resources. The Web Annotation Data Model provides an extensible, interoperable framework for expressing annotations such that they can easily be shared between platforms, with sufficient richness of expression to satisfy complex requirements while remaining simple enough to also allow for the most common use cases, such as attaching a piece of text to a single web resource. An annotation is considered to be a set of connected resources, typically including a body and target, and conveys that the body is related to the target. The exact nature of this relationship changes according to the intention of the annotation, but the body is most frequently somehow *about* the target. This perspective results in a basic model with three parts: annotation, body, and target. The full model supports additional functionality, enabling content to be embedded within the annotation, selecting arbitrary segments of resources, choosing the appropriate representation of a resource and providing styling hints to help clients render the annotation appropriately. Annotations created by or intended for machines are also possible, ensuring that the Data Web is not ignored in favor of only considering the human-oriented Document Web. The Web Annotation Data Model does not prescribe a transport protocol for creating, managing and retrieving annotations. Instead it describes a resource oriented structure and serialization of that structure that could be carried over many different protocols. The related annotation-protocol specification describes a recommended transport layer, which may be adopted separately. The specification provides a specific JSON format for ease of creation and consumption of annotations based on the conceptual model that accommodates these use cases, and the vocabulary of terms that represents it.

Nevertheless, ELAN represents the standard *de facto*, and also it is already in use by the partners in WP2, and so the digital platform of the SIGN-HUB project should mainly focus on this.

ii. Video Subtitles

A particular class of text-based information that it is possible to include within a video content is subtitle, giving real-time information about what is represented by the video in the exact moment the subtitle appears.

There are many reasons to add subtitles to a video. One of these is, surely, the fact that they make video with speech accessible to people who are deaf or in any case experience any hearing impairment. Nevertheless, they can also be used to convey meaning to videos showing signing characters, that could not be otherwise understood from an audience that does not know any sign language. A platform aiming at delivering media content to a mixed audience of both proficient and non-proficient subjects in sign languages should carefully provide subtitles for all its video contents. A screenshot of an example video is shown in Figure 2.

Subtitle Workshop (subworkshop.sourceforge.net/download.php) is a free application for creating, editing, and converting text-based subtitle files. It supports converting between, creating, opening, editing, and saving over 60 subtitle formats via the Subtitle API library, as well as saving subtitles in a custom user-defined file format, through an User-friendly, multi-language, customizable interface (including customizable shortcuts and translation mode).

Jubler (jubler.org) is a tool to edit text-based subtitles. It can be used as an authoring software for new subtitles or as a tool to convert, transform, correct and refine existing subtitles. The most popular subtitle formats can be used. Preview of the subtitles in real-time or in design time, spell checking, translation mode and styles editing are some of the main features. It is open source under a liberal (GNU) public license. It is written in Java 5.0 to be multi-platform.

Aegisub (aegisub.org) is a free, cross-platform open source tool for creating and modifying subtitles. Aegisub makes it quick and easy to time subtitles to audio, and features many powerful tools for styling them, including a built-in real-time video preview.

AHD Subtitles (ahd-subtitles-maker.webnode.com) is a powerful tool, but for Windows only, to create the most common text-based subtitle formats easily. It is a free comprehensive tool for subtitle editing and creation, with an extensive range of editing options and an impressive list of supported subtitle file formats. Thus, the program allows to create subtitle streams, trans-

late existing ones, or simply to change the format of your subtitle file to fit any of the most widely used subtitle editing tools.

Subtitle Edit (subtitle-edit.en.softonic.com), which is a free subtitle editor that uses an intuitive, clear interface as the basis for modifying .srt subtitles. It offers an intuitive, customizable interface.

Another free system is Aegisub (www.aegisub.org), which provides very basic functions only. There seem to be online tutorials available, which might be helpful.



Figure 2 - An example of a subtitled video with a signing character (screenshot from <https://www.youtube.com/watch?v=MY9c5y5Wgws>)



Figure 3 - An example of a video with two contemporaneous subtitle tracks (taken from <http://subs2srs.sourceforge.net>)

3. Analysis on Video Coding Format

As already stated in this document, video files, and visual materials in general, are multimedia content that needs particular care and strategies to be produced, stored, retrieved and analyzed; a common base point behind all these operations is given by the techniques enforced to represent content itself. For this reason, it is important to provide an overview about how video can be represented and which techniques are typically used in the modern digital world.

A video coding format is a content representation format for storage or transmission of digital video content [29]. This definition holds both in the cases when the video is meant to be stored within a file (as any document or other multimedia content could be) or to be transmitted over a digital communication channel in form of a bitstream (literally, a continuous flow of bits). In both cases, however, videos representation can be really heavy (in terms of employed bits), and this is especially the case for modern high-quality video and their high-, or ultra high-, definition standards.

To express this concept, it is useful to leverage on an example. A typical modern high quality video (i.e., a video that could be easily found on YouTube or similar video sharing platforms) is made up of frames (i.e., static images) with a resolution of 1920x1080 pixels (also known as 1080p), meaning that each frame is composed of more than 2 million pixels; frames are then interlaced at a temporal frequency of 50Hz (i.e., during a second of video there are 50 different frames which are interchanged, and in the USA this number rises up to 60) and each pixel is typically represented by means of at least 10 bits. Taking all this into account, the necessary multiplications show that to represent a single second of the video almost a billion of bits is needed. Such representation would be surely impractical, for modern technology, for storing and/or sharing any video that is longer than very few seconds. And things get worst when thinking that today a video with a frame resolution of 1920x1080 is considered as (almost) outdated, as newer resolution standards which are even more demanding in terms of bits are rising (just think of the 3840x2160 pixels format, also known 4k). Today, despite of a considerable technological advance, Internet connection rates still average in the range of a few megabits per second; a comparison with rates required from some video representation is reported in Table 1.

Table 1 - Video Resolution Format Comparison (adapted from <http://www.logitech.com/assets/45120/logitechh.pdf>)

Format	Horizontal Pixels	Vertical Lines	Pixels	Megabits per second
QVGA	320	240	76800	37
VGA	640	480	307200	147
720p	1280	720	921600	442
1080p	1920	1080	2073600	995
4k	3840	2160	8294400	3980

In addition, the examples below show the differences in storage space required when lossy, lossless and uncompressed video codecs are used:

- uncompressed (e.g., v210) 10-bit -> approx. 100GB per hour of video;
- lossless compression (FFV1 and JPEG 2000) 10-bit -> approx. 45-50 GB per hour of video;
- lossy compression:

- MPEG 2 (50 Mbps) -> approx. 25 GB per hour of video;
- DV (DV25) -> approx. 12 GB per hour of video;
- MPEG 2 (DVD quality) -> approx. 3.6 GB per hour of video.

It is obvious that, having considered these numbers, one should stop and ask himself how it is possible, after all, that millions and millions of videos content is everyday shared over the Internet, and that a one-hour lasting video can occupy less than 1GB (meaning, less than eight billion of bits, which is remarkable considering that one hour is made of 3600 seconds) when stored as file into a regular mass media storage support. This is so because, in the last years, the development of new and more demanding, but also impressive, video format has been accompanied by a parallel development of always more effective compressing techniques. Although significant gains in storage, transmission, and processor technology have been achieved in recent years, it is primarily the reduction of the amount of data that needs to be stored, transmitted, and processed that has made widespread use of digital video a possibility. Today, *video coding* itself has become a synonym for *video compression*. Advances in compression technology more than anything else have made possible the delivery of video contents and channels to our homes, desktop computers and mobile platforms. This is true for any kind of files, since compressed files require less disk space, can be easier attached to email, and require less time for network streaming, but it is more evident with video, which tend to be very heavy multimedia objects as already shown.

Compression is a reversible conversion of data to a format that requires fewer bits, so that the data can be stored or transmitted more efficiently. Compression is analogous to folding a letter before placing it in a small envelope so that it can be transported more easily and cheaply. Compressed data, like the folded letter, is not easily read and must first be decompressed, or unfolded, to restore it to its original form. Although always being reversible, this process may lead to lose data to some extent, as it will be clear from the remaining of this Section. In particular, if the inverse of the process, known as decompression, produces an exact replica of the original data, then the compression is said to be lossless. Compression is said to be lossy, instead, when it does not allow reproduction of an exact replica of the original data. Thus, lossy compression allows only an approximation of the original content to be generated after decompression, and it is typically more effective in terms of reducing the requirements for space needed for representation than lossless one.

The success of multimedia content compression depends largely on the content itself, as some data is inherently more compressible than other. Generally, with multimedia content it is possible to enforce modern coding and compression techniques to reduce redundancy within the content itself, that is later restored (with the possible introduction of errors or approximations) during decompression. The greater the redundancy within the data, the more successful the compression of the data is likely to be. Digital video content contains a great deal of redundancy and thus is very suitable for compression. In fact, it is usual for the sequence of frames composing a video to contain spatial and temporal redundancy that video compression algorithms attempt to eliminate or at least to encode in a smaller size. Similarities can be encoded by, e.g., storing only differences between frames, or by using perceptual features of human vision. For example, small differences in color are for human eyes more difficult to perceive than are changes in brightness, and so might be discarded in some lossy compression.

In the remaining of this Section, modern standards for video compression are discussed, and more insights on lossy and lossless video compression are given.

a. Codec: Compression/Decompression

A device (intended as both software or hardware) that compresses data is said to be an encoder or coder, whereas a device that decompresses data is known as a decoder. A device that acts as both a coder and decoder is known as a codec, which is an encoding tool that processes video and stores it in a stream, or flow, of bytes. Codecs use algorithms to effectively shrink the size of video files, and then decompress it when required (e.g., when a user opens a video file). There are dozens of different types of codecs, and each uses a different technology to encode and shrink a video file for the intended application. The most spread and known are hereby listed and discussed.

i. MPEG Standards

The need to set, and update, standards for compression and transmission of multimedia content, and video in particular, is such that in 1988 was established the Moving Picture Experts Group (known as MPEG), which is a working group of authorities that was formed by ISO and IEC on the purpose [30]. At the same time this acronym describes a whole family of international standards for the compression of audio-visual digital data or, more precisely, a set of parts within a single standard. The most known parts are MPEG-1, MPEG-2 and MPEG-4, which are also formally known as ISO/IEC-11172, ISO/IEC-13818 and ISO/IEC-14496 [31]. Their most important aspects are summarized as follows.

The MPEG-1 standard was released on 1993. It aims to provide moderate video quality with a bandwidth of 1,5 Mb/s, which allowed to play a video in real time from a 1x CD-ROM. The frame rate in MPEG-1 is locked at 25 frames per second, half of the one enforced from modern standards. Quite remarkably, MPEG-1 was designed to allow a fast forward and backward search and a synchronization of audio and video, and to require low computation times for encoding and decoding operations.

Two years later the MPEG-2 standard was released, which allowed a higher quality with a slightly higher bandwidth. MPEG-2 is compatible to MPEG-1. Later it was also used for High Definition Television (HDTV) and DVD. MPEG-2 is more scalable than MPEG-1 and can play the same video in different resolutions and frame rates, still maintaining backward compatibility, and it is still today the de-facto standard for digital video represented within DVD.

MPEG-4 is the most famous standard developed from the MPEG group. It was released 1998 and it provided lower bit rates (10Kb/s to 1Mb/s) with a good video quality. It was a major development from MPEG-2 and was designed for the use in interactive environments, such as multimedia applications and video communication. MPEG-4 uses further coding tools with additional complexity to achieve higher compression factors than MPEG-2. In addition to more efficient coding of video, MPEG-4 moves closer to computer graphics applications. It enhances the MPEG family with tools to lower the bit-rate individually for certain applications. It offers a better reusability of the contents as well as copyright protection [32]. Several new higher-efficiency video standards have been included over years; MPEG-4 Part 10, known as Advanced Video Coding (MPEG-4 AVC, also known as H.264), is the most well-known and spread among the examples.

ii. H.264

H.264 is currently one of the most commonly used formats for the recording, compression, and distribution of video content. The intent behind the development of the H.264 codec was to create a standard capable of providing good video quality at substantially lower bit rates than previous standards, without increasing the complexity of design so much that it would be impractical or excessively expensive to implement. Typically, a codec based on the H.264 standard compresses a digital video file so that it only uses half the space of MPEG-2, to represent video with the same quality. An additional goal of its developers was to provide enough flexibility to allow the standard to be applied to a wide variety of applications on a wide variety of networks and systems. The H.264 standard can be viewed as a family of standards com-

posed of many different profiles. A specific decoder decodes at least one, but not necessarily all profiles. The decoder specification describes which profiles can be decoded.

The standard defines sets of capabilities, which are referred to as profiles, targeting specific classes of applications. These are declared as a profile code (`profile_idc`) and a set of constraints applied in the encoder. This allows a decoder to recognize the requirements to decode that specific stream. Profiles for non-scalable 2D video applications include the following for high-definition video content:

- High Profile: The primary profile for broadcast and disc storage applications, particularly for high-definition television applications (for example, this is the profile adopted by the Blu-ray Disc storage format and the DVB HDTV broadcast service);
- Progressive High Profile: Similar to the High profile, but without support of field coding features;
- Constrained High Profile: Similar to the Progressive High profile, but without support of B (bi-predictive) slices.
- High 10 Profile: Going beyond typical mainstream consumer product capabilities, this profile builds on top of the High Profile, adding support for up to 10 bits per sample of decoded picture precision;
- High 4:2:2 Profile: Primarily targeting professional applications that use interlaced video, this profile builds on top of the High 10 Profile, adding support for the 4:2:2 chroma subsampling format while using up to 10 bits per sample of decoded picture precision;
- High 4:4:4 Predictive Profile: This profile builds on top of the High 4:2:2 Profile, supporting up to 4:4:4 chroma sampling, up to 14 bits per sample, and additionally supporting efficient lossless region coding and the coding of each picture as three separate color planes.

iii. H.265

Still, despite the impressive achievement of H.264 with regards to its predecessors, modern advances in technology gave made it possible to produce even more performing codecs. So, between 2004 and 2010, several research groups have worked to deliver High Efficiency Video Coding (HEVC), also known as H.265. This is a video compression standard, one of several potential successors to the widely-used H.264. In comparison to this last one, H.265 offers about double the data compression ratio at the same level of video quality, or substantially improved video quality at the same bit rate. It supports resolutions up to 8192×4320, including 8K UHD.

In most ways, H.265 is an extension of the concepts already expressed from the H.264 codec. H.265 was developed by the Joint Collaborative Team on Video Coding (JCT-VC). The JCT-VC brings together image and video encoding experts from around the world, producing a single standard that is now approved by two standards bodies: the Video Coding Experts Group (VCEG) that has published the H.265 standard as ITU-T H.265, and the MPEG, which published the H.265 standard as ISO 23008-2. Surely, for a modern platform aiming at delivering video content, H.264 and even more H.265 codecs represent a valid choice given their performances, their spread support and large development community.

iv. Motion JPEG 2000

Motion JPEG (M-JPEG or MJPEG) is a video compression format in which each video frame or interlaced field of a digital video sequence is compressed separately as a JPEG image. Originally developed for multimedia PC applications, M-JPEG is now used by video-capture devices such as digital cameras, IP cameras, and webcams, as well as by non-linear video editing systems. It is natively supported by the QuickTime Player, the PlayStation console, and web browsers such as Safari, Google Chrome, Mozilla Firefox and Microsoft Edge. Specifically, Motion JPEG 2000 (MJ2 or MJP2) is a file format for motion

sequences of JPEG 2000 images and associated audio, based on the MP4/QuickTimeformat. Filename extensions for Motion JPEG 2000 video files are .mj2 and .mjp2, as defined in RFC 3745. MJ2, first defined by Part 3 of the ISO Standard for JPEG 2000 ISO/IEC 15444 in November 2001 (ISO/IEC 15444-3:2002) as a standalone document, has later been defined by ISO/IEC 15444-3:2007, ISO/IEC 15444-3:2007/Amd 1:2010, additional profiles for archiving applications, and by ISO/IEC 15444-12 which defines the JPEG 2000 base media format, which contains the timing, structure, and media information for timed sequences of media data.

Motion JPEG2000 was always intended to coexist with MPEG. Unlike MPEG, MJ2 does not implement inter-frame coding; each frame is coded independently using JPEG 2000. This makes MJ2 more resilient to propagation of errors over time, more scalable, and better suited to networked and point-to-point environments, with additional advantages over MPEG with respect to random frame access, but at the expense of increased storage and bandwidth requirements.

v. AV1

AOMedia Video 1 (AV1) is an open, royalty-free video coding format designed for video transmissions over the Internet. It is being developed by the Alliance for Open Media (AOMedia), a consortium of leading firms from the semiconductor industry, video on demand providers, and web browser developers, founded in 2015. It is the primary contender for standardization by the video standard working group NetVC of the Internet Engineering Task Force (IETF). The group has put together a list of criteria to be met by the new video standard. It is meant to succeed its predecessor VP9 and compete with HEVC/H.265 from the Moving Picture Experts Group. AV1 can be used together with the audio format Opus in a future version of the WebM format for HTML5 web video and WebRTC.

The purpose of AV1 is to be as good as possible under royalty-free patent licensing. Crucial to this objective is therefore to ensure, during development, that it does not infringe on patents of competing companies. This contrasts to its main competitor HEVC, for which IPR review was not part of the standardization process. The latter practice is stipulated in ITU-T's definition of an open standard. The case of HEVC's independent patent pools has been characterized by critical observers as a failure of price management. Under patent rules adopted from the World Wide Web Consortium (W3C), technology contributors license their AV1-connected patents to anyone, anywhere, anytime based on reciprocity, i.e. as long as the user does not engage in patent litigation. As a defensive condition, anyone engaging in patent litigation loses the right to the patents of all patent holders. It aims for state of the art performance with a noticeable compression efficiency advantage at only slightly increased coding complexity. The efficiency goal is 25% improvement over HEVC. AV1 is primarily intended for lossy encoding, although lossless compression is supported as well. It is specifically designed for real-time applications (especially WebRTC) and higher resolutions (wider color gamuts, higher frame rates, UHD) than typical usage scenarios of the current generation (H.264) of video formats where it is expected to achieve its biggest efficiency gains. It is therefore planned to support the color space from ITU-R Recommendation BT.2020 and 10 and 12 bits of precision per color component.

vi. FFV1

FFV1, which stands for "FF video codec 1", is a lossless intra-frame video codec. It can use either variable length coding or arithmetic coding for entropy coding. The encoder and decoder are part of the free, open-source library libavcodec in the project FFmpeg since June 2003. FFV1 is also included in ffdshow and LAV Filters, which makes the video codec available to Microsoft Windows application that support system-wide codecs over Video for Windows (VfW) or DirectShow. FFV1 is particularly popular for its performance regarding speed and size, compared to other lossless preservation codecs, such as M-JPEG2000.

The "Österreichische Mediathek" has also developed DVA-Profession, a Free Software solution for archive-suitable mass video digitization, mainly using FFV1 as video encoding throughout the whole workflow, without transcoding. Additionally, they have initiated the development of "FFV1.3" (=version 3 of FFV1) together with Michael Niedermayer (FFmpeg), Peter Bu-

bestinger and Dave Rice. FFV1.3 contains improvements and new features such as support for multi-threaded encoding/decoding, error resilience and integrity validation by CRC checksums, storing of display aspect ratio (DAR) and field order. It was tested for over 1 year, and officially released stable for production in August 2013. In August 2016, support for 48bit/16bpc (=bits per component) in RGB was added. Before that, 16bpc in FFV1 were only supported in YCbCr and RGB was limited to 14bpc.

Within the video archiving domain, the interest in FFV1 is increasing, defining FFV1 always more as suitable for long-term preservation of digital content. Companies are also picking up FFV1 support. For example, NOA (formerly "NOA Audio Solutions"), announced support for the FFV1 in their product line in July 2013 and KEM-Studiotechnik released a film-scanner with FFV1 output in November 2013.

vii. WebM

WebM is a media file format (filename extension is typically .webm). It is primarily intended to offer a royalty-free alternative to use in the HTML5 video and the HTML5 audio tags. It has a sister project WebP for images. The development of the format is sponsored by Google, and the corresponding software is distributed under a BSD license. The WebM container is based on a profile of Matroska. WebM initially supported VP8 video and Vorbis audio streams. In 2013 it was updated to accommodate VP9 video and Opus audio.

VP8 is an open and royalty free video compression format owned by Google and created by On2 Technologies as a successor to VP7. In May 2010, after the purchase of On2 Technologies, Google provided an irrevocable patent promise on its patents for implementing the VP8 format, and released a specification of the format under the Creative Commons Attribution 3.0 license. That same year, Google also released libvpx, the reference implementation of VP8, under the revised BSD license. Opera, Firefox, Chrome, and Chromium support playing VP8 video in HTML5 video tag. Internet Explorer officially supports VP8 with a separate codec. According to Google VP8 is mainly used in connection with WebRTC and as a format for short looped animations, as a replacement for the Graphics Interchange Format (GIF). VP8 can be multiplexed into the Matroska-based container format WebM along with Vorbis and Opus audio. The image format WebP is based on VP8's intra-frame coding. VP8's direct successor, VP9, and the emerging royalty-free internet video format AV1 from the Alliance for Open Media (AOMedia) are based on VP8.

VP9 is a successor to VP8 and competes mainly with MPEG's High Efficiency Video Coding (HEVC/H.265). At first, VP9 was mainly used on Google's popular video platform YouTube. The emergence of the Alliance for Open Media, and its support for the ongoing development of the successor AV1, led to growing interest in the format. In contrast to HEVC, VP9 support is common among web browsers. The combination of VP9 video and Opus audio in the WebM container, as served by YouTube, is supported by roughly $\frac{3}{4}$ of the browser market (mobile included) as of early 2017, thanks to only two significantly popular video capable browsers lacking VP9 support: the discontinued Internet Explorer (unlike its successor Edge) and Safari (both desktop and mobile versions) which remains the last H.264 holdout among web browsers. Android has supported VP9 since version 4.4 KitKat, though hardware acceleration varies. Parts of the format are covered by patents held by Google. The company grants free usage of its own related patents based on reciprocity, i.e. as long as the user does not engage in patent litigations.

viii. Ogg Theora

Theora is a free lossy video compression format. It is developed by the Xiph.Org Foundation and distributed without licensing fees alongside their other free and open media projects, including the Vorbis audio format and the Ogg container (filename is typically .ogv). The libtheora video codec is the reference implementation of the Theora video compression format being developed by the Xiph.Org Foundation. Theora is derived from the formerly proprietary VP3 codec, released into the public domain by On2 Technologies. It is broadly comparable in design and bitrate efficiency to MPEG-4 Part 2, early versions of Windows Media Video,

and RealVideo while lacking some of the features present in some of these other codecs. It is comparable in open standards philosophy to the BBC's Dirac codec.

Theora is a variable-bitrate, DCT-based video compression scheme. Like most common video codecs, Theora also uses chroma subsampling, block-based motion compensation and an 8-by-8 DCT block. Pixels are grouped into various structures, namely blocks, super blocks, and macroblocks. Theora supports intra-coded frames and forward-predictive frames, but not bi-predictive frames which are found in H.264 and VC-1. Theora also does not support interlacing, or bit-depths larger than 8 bits per component. Theora video streams can be stored in any suitable container format, but they are most commonly found in the Ogg container with Vorbis or FLAC audio streams. This combination provides a completely open, royalty-free multimedia format. It can also be used with the Matroska container.

ix. Xvid

Xvid is a video codec library following the MPEG-4 video coding standard, specifically MPEG-4 Part 2 Advanced Simple Profile (ASP). Xvid is contemporaneously a different implementation and primary competitor of the DivX codec: in contrast with the DivX codec, which is proprietary software developed by DivX Inc., Xvid is free software distributed under the terms of the GNU General Public License. This also means that unlike the DivX codec, which is only available for a limited number of platforms, Xvid can be used on all platforms and operating systems for which the source code can be compiled.

Xvid implements a high-quality compression that allows users to achieve very little bulky compressed video with a very reasonable quality loss. Therefore, for instance, this format makes it possible to store a complete reasonably high-quality movie on a regular CD-ROM (650-700 millions of Bytes ca.).

x. Windows Media Video and RealVideo

Windows Media Video (WMV) is the name of a family of video codecs and their corresponding video coding formats developed by Microsoft. It is part of the Windows Media framework. The first version of the format, WMV 7, was introduced in 1999, and was built upon Microsoft's implementation of MPEG-4. The original video compression technology known as WMV, was originally designed for Internet streaming applications, as a competitor to RealVideo. RealVideo is a suite of proprietary video compression formats developed by RealNetworks. It was first released in 1997 and it is supported on many platforms, including Windows, Mac, Linux, Solaris, and several mobile phones.

xi. Libavcodec

Libavcodec is a free and open-source library of codecs that provides a generic encoding/decoding framework. It contains multiple encoders and decoders for audio, video and subtitle streams, and several bitstream filters. The shared architecture makes it suitable for implementing robust and fast codecs as well as for experimentation purposes. Libavcodec contains decoder and sometimes encoder implementations of several proprietary formats, including ones for which no public specification has been released. As such, a significant reverse engineering effort is part of libavcodec development. Having such codecs available within the standard libavcodec framework gives several benefits over using the original codecs, most notably increased portability, and in some cases also better performance, since libavcodec contains a standard library of highly optimized implementations of common building blocks. However, even though libavcodec strives for decoding that is bit-exact to the official implementation, bugs and missing features in such reimplementations can sometimes introduce compatibility problems playing back certain files.

xii. Discussion on the codecs

Finally, the following Table summarizes Pros and Cons of the most important codecs here reviewed.

Table 2 - Video codecs

Codecs (encodings)	Pros	Cons
V210	<ul style="list-style-type: none"> ○ No loss of quality ○ Widely supported by software tools ○ Used by several archives 	<ul style="list-style-type: none"> ○ Proprietary, but well supported by open source and other tools ○ Results in big files
JPEG2000	<ul style="list-style-type: none"> ○ ISO non-proprietary standard ○ Smaller file than uncompressed video ○ Adopted by important AV archives ○ Lossless compression possible 	<ul style="list-style-type: none"> ○ Complex ○ Not widely supported in editing software ○ Not ideal for YUV material
FFV1	<ul style="list-style-type: none"> ○ Open source ○ Lossless compression ○ Smaller file than uncompressed video 	<ul style="list-style-type: none"> ○ Only used by a small amount of collections ○ Not popular in the broadcast field
D10 & AVC/H264	<ul style="list-style-type: none"> ○ Results in considerably smaller files than lossless codecs ○ Widely adopted by broadcast archives ○ Widely supported by tool chains 	<ul style="list-style-type: none"> ○ Lossy compression ○ Not widely adopted by heritage collections

There is no consensus to date among the archival community as to which file format or codecs should be used for preservation purposes for digital video. The previously proclaimed encodings were Motion JPEG 2000 (lossless) and uncompressed video. FFV1 has turned out to be a viable addition to that choice and was therefore recently added as a suitable option for preservation encoding.

For persistence of media streams, we recommend to not use lossy compressed formats. From lossless compressed files, other formats can be produced all the time dependent on the needs and converters are available. For the content providers and end users of SIGN-HUB, we recommend video to be recorded at least for a minimum of 720x486 pixels at 30 frames per second, with full-range color (at least 8 bits per channel with 10 bits per channel preferred), with audio tracks of digital video recordings at least 48KHz. It is suggested to avoid the following digital video files as they are known to have insufficient quality for archival retention:

- Files created for "streaming" broadcast (e.g. RealAudio, Windows Media) - these files sacrifice quality for file size. They are often reduced from high-resolution files by removing high and low frequency ranges and compressing the remaining signal;
- Reference files of lower quality than original (e.g. for web site use) - again, this sacrifice quality for file size;
- Files that have been transcoded. In other words, it is not acceptable to render previously encoded content through another codec.

We recommend recording new videos and storing information with the highest performant encoding possible being able to guarantee outcomes in very high definition (e.g., the highest quality possible for the digital camera used while recording). In fact, the need here is to store the highest quantity possible of the information within the videos for their further processing.

b. Video Container Format

Representation of any multimedia content, and video is not an exception, given any of the codec previously discussed would be useless without a proper environment to store it. A container exists solely for bundling all the audio, video, and codec files into one organized package. In addition, the container often contains extra information such as chapter for DVD movies, metadata, subtitles, and/or additional audio files such as different spoken languages tracks. Multimedia content is conveyed through containers, or file formats.

Good design normally dictates that a file extension should enable the user to derive which program will open the video file from the file extension. However, file formats are not always one-to-one related to codecs. For example, a file format such as Windows Media Video (.wmv files) typically contains data that is compressed leveraging on the Windows Media Video codec only. On the other hand, a file format such as Audio Video Interleaved (AVI, .avi files) can contain data that is compressed by any of several different codecs, including the MPEG-2, DivX, or Xvid codecs; in addition, AVI files can also contain data that is not compressed by any codec.

xiii. AVI

AVI is a multimedia container format introduced by Microsoft in 1992 as part of its Video for Windows software. AVI files can contain both audio and video data in a file container that allows synchronous audio and video playback. Like the DVD video format, AVI files theoretically support multiple streaming audio and video, although these features are seldom used. While being still today primarily associated with Windows codecs, recent AVI files might also be compressed with other codecs (e.g., Xvid). Its main drawback is that it does not provide a standardized way to encode aspect ratio information, with the result that players cannot select the right one automatically (though it may be possible to do so manually).

AVI files can be created with no compression, resulting in extremely large file sizes, but with no loss of quality from the input video to the saved file. This also requires no codecs to be installed, either for saving or playback. This is generally not recommended as extremely large file are unmanageable to store, and more importantly, to deliver.

xiv. MP4

MP4 stands for MPEG-4 Part 14, being in fact a standard specified as a part of MPEG-4 under the ISO 14496-14:2003 and the more general ISO 14496-12:2004. It is directly based upon the QuickTime File Format. MP4 is a digital multimedia container format most commonly used to store video and audio. Like most modern container formats, it allows streaming over the Internet. The only official filename extension for its files is .mp4, but many have other extensions are in use nowadays (e.g., .m4a and .m4p).

MP4 files can contain metadata as defined by the format standard, and in addition, can contain Extensible Metadata Platform (XMP) metadata, and are supported by most of modern operating systems and browsers. This explains why it has become standard de-facto for web-platform video delivery, and its usage is mostly recommended for modern web-sites willing to deliver video contents to users.

xv. QuickTime

QuickTime File Format (QTFF) is a multimedia file format used natively by the QuickTime framework. The format specifies multimedia container files containing one or more tracks, each of which stores a particular type of data: audio, video, or text (e.g., metadata such as subtitles). Each track either contains a digitally-encoded media stream (using a specific format) or a data reference to the media stream located in another file. Tracks are maintained in a hierarchical data structure consisting of objects called atoms. An atom can be a parent to other atoms or it can contain media or edit data, but it cannot do both.

The ability to contain abstract data references for the media data, and the separation of the media data from the media offsets and the track edit lists means that QuickTime is particularly suited for editing, as it can import and edit in place (i.e., without data copying) abstraction that is lacked from many other file formats.

Because both the QuickTime and MP4 container formats can use the same MPEG-4 formats, they are mostly interchangeable in a QuickTime-only environment. However, the MP4, being an international standard, has more support. This is especially true on hardware devices, many devices (particularly gaming platforms and DVD players) include an MP4 parser, but not one for QTFF.

xvi. FLV

Flash Video is a container file format used to deliver video over the Internet using Adobe Flash Player. There are two different video file formats known as Flash Video: FLV and F4V, the first being more popular. Both formats are supported in Adobe Flash Player and developed by Adobe Systems, even though FLV was originally developed by Macromedia. The most recent public releases of Flash Player also support H.264 codec.

Flash Video is viewable on most operating systems via the Adobe Flash Player and web browser plugin or one of several third-party programs. It used to be very spread and used for web-based streaming video, its providers including VEVO and Yahoo! Video, although, most notably, all the Apple's iOS-based devices do not support the Flash Player plugin and so require other delivery methods.

xvii. RealMedia

RealMedia is a proprietary multimedia container format created by RealNetworks. Its files are marked from the usage of the .rm extension. It is typically used in conjunction with RealVideo and RealAudio and it is used for streaming content over the Internet. RealMedia is a type of file designed by RealNetworks, and can be played with the proprietary RealPlayer. RealPlayer is available for Windows, Mac OS, Mac OS X and Linux.

xviii. Matroska

The Matroska Multimedia Container is an open and free container format; this means for personal use it is free to use and that the technical specifications describing the bitstream are open to everybody, even to companies that would like to support it in their products. The source code of the libraries developed by the Matroska Development Team is licensed under GNU L-GPL.

Matroska files can hold an unlimited number of video, audio, picture, or subtitle tracks in one file. It is intended to serve as a universal format for storing common multimedia content, like movies or TV shows. Matroska is similar in concept to other containers like AVI or MP4, but it is entirely open in specification, with implementations consisting mostly of open source software, which makes this container a valid alternative to solutions like MP4. Matroska file extensions are usually .mkv for video (with subtitles and audio) and .mks for subtitles only.

xix. Ogg

Ogg is a free, open container format maintained by the Xiph.Org Foundation. The creators of the Ogg format state that it is unrestricted by software patents and is designed to provide for efficient streaming and manipulation of high quality digital multimedia. Its name is derived from "ogging", jargon from the computer game Netrek. The Ogg container format can multiplex a number of independent streams for audio, video, text (such as subtitles), and metadata. In the Ogg multimedia framework, Theora provides a lossy video layer. The audio layer is most commonly provided by the music-oriented Vorbis format or its successor Opus. Lossless audio compression formats include FLAC, and OggPCM.

Before 2007, the .ogg filename extension was used for all files whose content used the Ogg container format. Since 2007, the Xiph.Org Foundation recommends that .ogg only be used for Ogg Vorbis audio files. The Xiph.Org Foundation decided to create a new set of file extensions and media types to describe different types of content such as .oga for audio only files, .ogv for video with or without sound (including Theora), and .ogx for multiplexed Ogg.

As of August 4, 2011, the current version of the Xiph.Org Foundation's reference implementation, is libogg 1.3.0. Another version, libogg2, has been in development, but is awaiting a re-write as of 2008. Both software libraries are free software, released under the New BSD License. Ogg reference implementation was separated from Vorbis on September 2, 2000. Because the format is free, and its reference implementation is not subject to restrictions associated with copyright, Ogg's various codecs have been incorporated into a number of different free and proprietary media players, both commercial and non-commercial, as well as portable media players and GPS receivers from different manufacturers.

The "Ogg" bitstream format, spearheaded by the Xiph.Org Foundation, has been created as the framework of a larger initiative aimed at developing a set of components for the coding and decoding of multimedia content, which are available free of charge and freely re-implmentable in software. The format consists of chunks of data each called an "Ogg page". Each page begins with the characters, "OggS", to identify the file as Ogg format. A "serial number" and "page number" in the page header identifies each page as part of a series of pages making up a bitstream. Multiple bitstreams may be multiplexed in the file where pages from each bitstream are ordered by the seek time of the contained data. Bitstreams may also be appended to existing files, a process known as "chaining", to cause the bitstreams to be decoded in sequence. A BSD-licensed library, called "libvorbis", is available to encode and decode data from "Vorbis" streams. Independent Ogg implementations are used in several projects such as RealPlayer and a set of DirectShow filters. Mogg, the "Multi-Track-Single-Logical-Stream Ogg-Vorbis", is the multi-channel or multi-track Ogg file format.

xx. Discussion on Video Containers

Finally, the following Table summarizes Pros and Cons of the most important containers here reviewed.

Table 3 - Video containers

Containers (wrap-pers)	Pros	Cons
AVI	<ul style="list-style-type: none"> o Simple o Widely used o Widely supported 	<ul style="list-style-type: none"> o Proprietary, but well supported by open source and other tools o Can't wrap complex objects
Quicktime (MOV, .QT)	<ul style="list-style-type: none"> o Widely used o Widely supported o Can wrap complex objects 	<ul style="list-style-type: none"> o Proprietary, but well supported by open source and other tools
MXF	<ul style="list-style-type: none"> o SMTPE non-proprietary standard o Can wrap complex object o Widely adopted by the audio-visual community (especially also by the broadcast sector) 	<ul style="list-style-type: none"> o Interoperability problems between manufacturers and profiles o Lacks a specific profile for preservation. o Not yet widely supported by software tools
Matroska	<ul style="list-style-type: none"> o Open Source o Can wrap complex objects 	<ul style="list-style-type: none"> o Not widely adopted by archives or the broadcast sector o Lacks a specific profile for preservation

In this, although being proprietary, we recommend using when possible the AVI video container, since it is well supported by open source projects and suitable for long-term preservation.

c. Lossy Video Compression

As already stated, video compression has two important benefits. Firstly, it makes it possible to use digital video transmission and storage environments that would not otherwise support uncompressed video, since they tend to be very demanding in terms of resources needed for their representation. For example, current Internet bandwidth is sometimes insufficient to handle uncompressed video in real time; a DVD can only store on average a few seconds of uncompressed television-quality video and so DVD-Video storage would be almost useless without video and audio compression.

Secondly, video compression enables more efficient use of transmission and storage resources. Even if, theoretically speaking, a very high bitrate transmission channel is available, then it is a more attractive proposition to send very high-resolution compressed video or multiple compressed video channels than to send a single uncompressed stream. While data reduction (compression in any form) is a main goal of transform coding, it also allows other goals. Through coding, in fact, it is also possible to represent a content in a better domain for manipulating or otherwise editing it. For instance, equalization of audio is most naturally expressed in the frequency domain rather than in the time one. From this point of view, video encoding is not essentially about discarding data, but rather about providing a better representation [33].

An information-carrying signal may be compressed by removing redundancy from it. In a lossless compression system, statistical redundancy is removed so that the original signal can be perfectly reconstructed at the receiver. Unfortunately, modern lossless methods can only achieve a modest amount of compression of image and video signals. Most practical video compression techniques are based on lossy compression, in which greater compression is achieved with the penalty that the decoded signal is not identical to the original: researchers in this field are working to produce always better algorithms and techniques, to achieve efficient compression whilst minimizing the distortion and errors introduced.

In the whole world of information technology, lossy compression is the class of data encoding algorithms that use inexact approximations and partial data discarding to represent multimedia content requiring a reduced amount of resources (disk space or bandwidth) for their fruition. These techniques are used to reduce data size for storage, handling, and transmitting content. Since some part of the content is inevitably lost after the compression process, this class of techniques is also known as irreversible compression, which is a slightly less negative term but conveys the same idea.

The advantage of lossy methods over lossless methods is that in some cases a lossy method can produce a much smaller compressed file than a lossless method, while still meeting the requirements of the application in terms of quality of the outcome. The compression ratio (that is, the size of the compressed file compared to that of the uncompressed file) that modern lossy video codecs can achieve is almost always remarkable: video can be compressed immensely, with a compress ratio that can grow up to 100:1 with little visible quality loss. Lossy methods are most often used for compressing videos intended for human interpretation and that tend to be bulky, since human minds can easily very minor errors or inconsistencies introduced from compression and human eyes can easily be tricked. Ideally, lossy compression is transparent to human: it produces different content which differences are, however, imperceptible for human eyes. Most video compression technologies, such as MPEG, use a lossy technique, and work by removing redundant or unnecessary information.

Lossy compression suffers from a main drawback, known as generation loss: repeatedly compressing and decompressing a video content will cause it to progressively lose quality (with a concept like repeatedly and recursively taking photocopies of a printed document, causing the outcome to degrade gradually but relentlessly). It holds the same when converting a video from a lossy codec to another, while converting the same video to a lossless codec would cause nothing but having a video with the same quality (since information is lost and cannot be restored) but requiring much more resources to be handled.

Manufacturers sometimes label “technically lossy” compression schemes as “visually lossless”, because the difference between the compressed video and the original is supposed to be imperceptible to the (common) human eye. Despite its name, “visually lossless” is a compression method in which a part of the data is permanently discarded. For this reason, “visually lossless” is also sometimes more accurately defined as “near-lossless compression”. For the purposes of this document, the term “lossy” is used also to refer to “visually lossless” compression.

Lossy video compression is of particular interest for the SIGN-HUB project for what concerns content distribution and specifically video streaming; in fact, the SIGN-HUB project aims at providing end users with a plethora of videos forming a digital archive of sign languages, which are meant to be fully accessible and interoperable. With modern technology and network capabilities, it would be meaningless to resort to lossless video compression for video streaming. In this, HTML5 is representing the modern standard-de-facto for video streaming for web browsers.

The revolution of HTML5 is an exciting web developers and designers from all around the world. The latest specifications support dozens of attributes and elements for building high quality websites. Some of the newest features of HTML5 include tags to assist with the structure of your web page, such as article and section and multimedia tags for helping you in adding video and audio formats. In the past years, Flash-based media players were sufficient for streaming videos on the Web. However, the advancement of modern standards has resulted in the inclusion of HTML5 video, which opens doors for several new opportunities. HTML5 is the first version which is providing a standard method to embed video by using the video tag. As it can be embedded in the browser directly, so now there is no need of third party plug-in, such as Silverlight and Flash. HTML5 video has created unique and new ways for creating interactivity between the video, the user, and other page elements. By the use of HTML 5 video tag, it is easy to add video to Web pages. However, although it appears easy on the surface, still there are many things that you should understand to get the video up and running. If you want to use HTML5 video on your website, then the first step is to choose a codec or codecs that you will use. Currently, there are three codecs that support HTML5:

- WebM;
- Ogg/Theora;
- H.264 or MP4.

The support for the HTML video tag is evolving constantly. HTML video is not supported by every browser and the browser which supports HTML5 needs a different video format. Some browsers, such as Internet Explorer versions do not support HTML5 videos and require flash video. According to the latest data, more than 80 percent of the market supports HTML5 video. However, Internet Explorer 8 is the leading non-HTML5 browser on the desktop, whereas the non-HTML line on mobile mainly comprises of proxy browsers, such as Nokia Xpress, UC Browsers and Opera Mini.

In particular, according to Wikipedia, only Firefox and Opera employ libraries for built-in decoding. In practice, Internet Explorer and Safari can also guarantee certain format support, because their manufacturers also make their multimedia frameworks. At the other end of the scale, Konqueror has identical format support to Internet Explorer when run on Windows, and Safari when run on Mac, but the selected support here for Konqueror is the typical for GNU/Linux, where Konqueror has most of its users. In general, the format support of browsers is much dictated by conflicting interests of vendors, specifically that Media Foundation and QuickTime support commercial standards, whereas GStreamer and Phonon cannot legally support other than free formats by default on the free operating systems that they are intended for. Figure 4 shows which video formats are supported by a given user agent.

Browser	Operating System	Theora	H.264 (MP4)	HEVC (MP4)	VP8 (WebM)	VP9 (WebM)
Android browser	Android	Since 2.3 ^[44]	Since 3.0 ^[44]	No ^[45]	Since 2.3 ^[44]	Since 4.4 ^[46]
Chromium	Unix-like and Windows	Since r18297 ^[47]	Via FFmpeg ^{[48][49]}	No ^[50]	Since r47759 ^[51]	Since r172738 ^[52]
Google Chrome	Unix-like, Android, macOS, iOS, and Windows	Since 3.0 ^{[53][54]}	Since 3.0 ^{[54][a]}	No ^[45]	Since 6.0 ^{[56][57]}	Since 29.0 ^[b]
Internet Explorer	Windows	Via OpenCodecs	Since 9.0 ^[60]	No ^[45]	Via OpenCodecs	No
	Windows Phone	No	Since 9.0 ^[61]			
	Windows RT	No	Since 10.0 ^[61]			
Microsoft Edge	Windows 10	No ^[62]	Since 12.0 ^[63]	Needs hardware decoder ^[c]	Only via MSE Since 16.0 ^[66]	Needs hardware decoder ^[67] Only via MSE ^[68] Since 15.0
	Windows 10 Mobile		Since 13.0 ^[69]			
Konqueror	Unix-like and Windows	Needs OS-level codecs ^[d]				
Mozilla Firefox	Windows 7+	Since 3.5 ^[71]	Since 21.0 ^[e]	No ^[45]	Since 4.0 ^{[74][75]}	Since 28.0 ^{[76][77]}
	Windows Vista		Since 22.0 ^[78]			
	Windows XP and N editions		Since 46.0 ^[79]			
	Linux		26.0 (via GStreamer) ^[f] 43.0 (via FFmpeg) ^[82]			
	Android		Since 17.0 ^[83]			
	macOS		Since 34.0 ^[84]			
Opera Mobile	Android, iOS, Symbian, and Windows Mobile	Since 13.0	Since 11.50	No ^[86]	Since 15.0	Since 16.0
Opera	macOS, Windows, Linux	Since 10.50 ^[87]	Since 24.0 ^[88]		Since 10.60 ^{[89][90]}	Yes
Safari	iOS	No	Since 3.1 ^{[91][92]}	Since 11 ^[93]	No	No
	macOS	Via Xiph QuickTime Components				
Web	Linux and BSD	Needs OS-level codecs ^[g]				

Figure 4 - Status of video format support in each web browser

While H.264/MPEG-4 AVC contains patented technology, and requires licenses from patent holders and limited royalties for hardware, Google has irrevocably released the VP8 patents it owns under a royalty-free public license. According to a comparison of VP8 (encoded with the initial release of libvpx) and H.264 conducted by StreamingMedia, it was concluded that "H.264 may have a slight quality advantage, but it's not commercially relevant" and that "Even watching side-by-side (which no viewer ever does), very few viewers could tell the difference". They also stated that "H.264 has an implementation advantage, not a technology advantage." Google's claims that VP8 offers the "highest quality real-time video delivery" and Libvpx includes a mode where the maximum CPU resources possible will be used while still keeping the encoding speed almost exactly equivalent to the playback speed (realtime), keeping the quality as high as possible without lag. On the other hand, a review conducted by streaming-media.com in May 2010 concluded that H.264 offers slightly better quality than VP8.

d. Lossless Video Compression

Lossless compression is a class of data compression algorithms that allows the original data to be perfectly reconstructed after the decompression procedure. Lossless compression is used in cases where deviations from the original data could be deleterious and errors in data reconstruction are inadmissible. Typical examples are executable programs, text documents, and source code, i.e., data that would suffer from errors and which content could be lost even after only errors on a few Bytes is introduced.

For what concerns video, this class of algorithms is rarely and scarcely enforced, mainly when the video itself is not intended for human audience but for some automated analysis from a machine, that would surely suffer from loss of information, or for some selected professional use case. Lossless coding of video sequences is today important for a reduced number of applications ranging from digital cinema, post-production, archiving and, finally, medical applications. For example, medical imaging applications often generate very resources-demanding sequences of strongly related images: compression is clearly desirable, both for storage and remote medical applications, but when dealing with human life (just think to modern advances in cancer treating through medical imaging) up to the last bit of information is to be preserved. Finally, one of the digital cinema's main requirements is lossless video compression, due to the strict needs during acquisition, post-production, archiving and distribution. The maximum quality possible has to be preserved during all the steps in the chain from the acquisition to the film theaters, and generation loss must be avoided [34].

Most lossless compression programs work on two steps in sequence. Within the first step they generate a statistical model for the input data, while within the second one they use this model to map input data to bit sequences in such a way that frequently encountered data is represented in a shorter way than other, less probable to occur, data. The primary encoding algorithms used to produce bit sequences are Huffman coding and arithmetic coding. Arithmetic coding achieves compression rates close to the best possible for a statistical model, which is given by the information entropy, whereas Huffman compression is simpler and faster but produces poor results.

There are two primary ways of constructing statistical models: in a static model, the data is analyzed and a model is constructed, then this model is stored with the compressed data. This approach is simple and modular, but has the disadvantage that the model itself can be expensive to store, and that it forces using a single model for all data being compressed, and so performs poorly on files that contain heterogeneous data. Adaptive models dynamically update the model as the data is compressed. Both the encoder and decoder begin with a trivial model, yielding poor compression of initial data, but as they learn more about the data, performance improves. Most popular types of compression used in practice now use adaptive coders.

Quite interestingly, lossless data compression algorithms cannot guarantee compression for all input multimedia content. In fact, it is possible to demonstrate that, for any lossless data compression algorithm, there will be an input data set that does not get smaller when processed by the algorithm, and for any lossless data compression algorithm that makes at least one file smaller, there will be at least one file that it makes larger. Basic information theory says that there is an absolute limit in reducing the size of any data. When data is compressed, its entropy increases, and it cannot increase indefinitely. Considering this, modern software and codec perform some analysis of the input data prior to compression to evaluate if compression itself is useful for that particular file or if it should be avoided. As an intuitive example, think about a compressed ZIP file: while being smaller than the original file, repeatedly compressing the same file will not reduce the size to nothing. Most compression algorithms can recognize when further compression would be pointless and would in fact increase the size of the data.

Not many lossless video formats are in use today. MagicYUV is a high-performance, ultra-fast, mathematically lossless video codec for recording, archiving, video post-production and uncompressed-quality playout at high resolutions. It provides faster-than-real time performance for FullHD and 4K resolutions on modern PC hardware. MagicYUV is a good intermediate for-

mat for video post-production and delivery format of high-quality playout, reducing bandwidth without quality degradation, when it is necessary to keep the video material bit-by-bit identical to the original.

For long-term preservation of digital video sustainable container formats as well as audio/video codecs are necessary. There is no consensus to date among the archival community as to which file format or codecs should be used for preservation purposes for digital video. The previously proclaimed encodings were Motion JPEG 2000 (lossless) and uncompressed video. FFV1 has turned out to be a viable addition to that choice and was therefore recently added as a suitable option for preservation encoding. With compression ratios comparable to JPEG 2000 lossless and its lower computing requirements, it is already being used by professional archives as their long-term storage codec. Especially archives where the collections do not feature extensive broadcast materials but rather consist of, oral history-, ethnographic-recordings and the likes, "favored the lossless FFV1 encoding" in communications with the "Federal Agencies Digitization Guidelines Initiative" (FADGI) team. As of 2015, standardization of FFV1 through the Internet Engineering Task Force (IETF) is work in progress as part of the European PREFORMA Project, as well as implementation of a conformance checker for FFV1/PCM in a Matroska (MKV) container. Details of FFV1's standardization plan have been prepared by MediaArea (authors of MediaInfo) as part of their conformance checking tool "Media CONCH". It is also listed as a format option for long-term preservation of moving images on sites of the U.S. Library of Congress', State Records NSW and others. The Society of American Archivists has published a paper in August 2014, suggesting only FFV1 as preservation codec for video. The Digital Preservation project at the U.S. Library of Congress identified AVI and Matroska as common container formats for FFV1. Other institutions known to use FFV1 are:

- Austrian Museum of Modern Art (MuMoK): For their collection of art videos;
- Filmarchiv Austria (Austrian Film Archive): For their telecined DigiBeta copies of some films;
- Österreichische Mediathek (Austria's national audio/video archive).

Lossless video compression is of particular interest for the SIGN-HUB project for what concerns content preservation and specifically long-term archiving of video content; in fact, the SIGN-HUB project aims at providing researchers with a plethora of videos forming a digital archive of sign languages, which are meant to be fully findable and usable for many decades in the future, even possibly enabling research scenarios which are not even known by today. It is therefore important to ensure that no feature nor information of the video stored is lost.

The archiving community has defined sets of principles to evaluate the sustainability factors of file formats for long-term archiving. One example of these evaluation tools is the list created by the Library of Congress for its own collection. The criteria considered by PACKED vzw during its research in the framework of VIAA are largely based on this list and others developed by others, such as the InterPares project or the United Kingdom's National Archives. They comprise:

- **the quality:** the quality of the file should be high enough to anticipate future use and to avoid risk of quality loss over time;
- **the openness:** there should be no restriction on the use and reuse of the file such as licenses that can constitute a threat to the adoption and support of the format;
- **the adoption:** the format should be widely used by archives or different domains and have sufficient support in existing tools;
- **the transparency:** the file format should be easy to analyze;
- **the durability:** the format shouldn't be expected to become obsolete or need transcoding too quickly. Backward compatibility in the short term should be ensured;
- **the functionality:** the file should be able to deal with complex objects;
- **the handling:** the file should be handled easily, efficiently and without (or very limited) risk of error and threat to workflows.

While the criteria are clear, a format combining all the listed requirements such as openness (important for cultural heritage institutions) and efficient handling (critical for the broadcast sector), is not yet apparent. The result of this is that the choice of the archiving format is always a compromise, whereby different types of institutions do not necessarily prioritize the same criteria.

There seem to be only two options for one to be sure that the best quality is captured digitally from the original video source: an uncompressed or lossless codec. As seen also, uncompressed video requires a lot of storage capacity and the capability to handle large amounts of data efficiently. For the same amount of storage needed for one uncompressed copy, lossless encoding with FFV1 or lossless JPEG2000 allows one to store around one and a half to two copies. Although lossless compression results in smaller files, it may require more processing power due to the algorithms used in lossless compression. In a digitization project, financial means and storage capacity are often considered the crux of decision-making. Compressed format such as MPEG-2/D10 or lossy JPEG2000 allow even more reduced storage space, but represent a big risk for the (future) quality of archival material.

FFV1 is the only open source codec that could be used by an audio-visual archive, but it is still in the process of gaining acceptance within the archival community and unknown by a large part of it. Not many archives are keen on being pioneers in using long-term preservation formats. When a more widely used combination of FFV1 and Matroska become available and are further supported by software tools and manufacturers, it might be the best option for collections that can't rely on specific proprietary hardware and software to use their files. However, its limited adoption compared to JPEG2000 and MPEG-2/D10 makes it difficult to convince any heritage institutions and broadcast archives to choose it as their long-term preservation format.

Also, small research project often cannot afford to have the necessary software to edit material in MXF/JPEG2000 or to transcode it, it is crucial. The Institut National de l'Audiovisuel, the Library of Congress, the Library and Archives of Canada, and the National Archives of Australia are amongst the biggest audio-visual archives in the world digitizing their material with a lossless JPEG2000 codec wrapped in MXF. This adoption amongst big institutions constitutes a community with an amount of material and resources important enough to think that there will always be a solution to migrate and access these archives in the future. While there are still disparities in the MXF profiles they use, the work being done on developing the AS-07 specifications to make a common MXF profile for audio-visual archives is encouraging given the number of large institutions and manufacturers involved in it. However, this profile is still a work in progress that started back in 2009, and there is still no fixed time frame as to when it will be ready.

To prepare the digital archive infrastructure properly, costs for (amongst others) storage, transcoding hardware and network requirements need to be evaluated before the digitization process has even started. Uncompressed and lossless compressed formats generate bigger files than lossy compression and therefore require more storage space and processing power. Costs will accumulate throughout the years, but storage and processing capacity also becomes cheaper every year according to Moore's law. For an important digitization process, costs will probably decrease before the end of the digitization is completed. From an archival and conservation point of view, storage costs shouldn't be a decisive criterion, and not one to favor over quality aspects and long-term sustainability. In the future, there is a big chance that the additional cost for storing lossless compressed files will become a lesser concern than the quality of the digitized content. Use cases that we can't even imagine today might require a very high-quality video file. Lossy compression is a risky path that could lead to important quality problems and the safest decision is to avoid any type of information loss. If the digitization process has to be remade because the quality standards chosen were too low, the costs would be far greater than the supplementary investment needed to store lossless video files today. More importantly, there is a big chance that renewed digitization would no longer be possible because of the advanced deterioration of the carriers, the obsolescence of the necessary playback equipment and the lack of skilled operators.

e. Representing and Encoding 3D Graphics Assets

During the last decade, animated 3D content made a spectacular break-through into the world of digital multimedia. Application domains include general public, entertainment, educational as well as professional products and services with high socio-economic impact [35]. This is the case of 3D virtual avatar employed for signing multimedia content that would be otherwise inaccessible for deaf and hearing-impaired people and for giving support to teaching languages.

In the context of the virtual reality environment, an avatar is a graphical representation of the human form. The human body is accompanied with a wide amount of gestures which need to be represented to impart the avatar with the necessary realism. Despite continuous research and development of new technologies, combined with growing expectations, there is still a need for standards for creating, and more importantly representing, 3D multimedia content and virtual avatar.

In the last decade, several efforts have been made to develop a unique data format for interchanging assets between authoring tools. In the category of open standards, X3D and COLLADA are the best known, the latter probably being the most adopted by current tools, although still far from being the solution to all the challenges posed from 3D graphics production. While the COLLADA concentrates on representing 3D objects or scenes, X3D pushes the standardization further by addressing user interaction as well thanks to an event model in which scripts, possibly external to the file containing the 3D scene, may be used to control the behavior of its objects. One of the characteristics of both X3D and COLLADA is their data representation based on XML.

Textured, animated polygonal meshes are the building block for any 3D virtual figure. Nowadays they are supported by two families of ISO standards: Virtual Reality Modeling Language (VRML-97 and its successor, X3D), and MPEG-4. In particular, MPEG-4 contains several technologies for efficient streaming of compressed multi-textured polygonal 3D meshes.

This brief section is introduced in this document since we envisage that sign language content can be produced also by virtual actors (i.e., avatars); therefore, it would be desirable for any repository and platform developed within the SIGN-HUB project to be already compatible with the standards and formats described in the following; nevertheless, this is not a strict requirement for the SIGN-HUB project.

i. MPEG-4 Animation Framework eXtension (AFX)

Since 1999, when the first version of MPEG-4 was published, the MPEG committee has had a continuous activity on defining standards for representing 3D graphics assets and scenes. MPEG-4 offers a rich set of tools that may be classified with respect to data type. The most generic, called BIFS (Binary Format for Scene), allows compressing a scene graph expressed in a similar way to VRML-97.

VRML-97 is a text file format to represent vertices, edges, textures, shininess, transparency and related properties of 3D polygons. URLs can be associated with graphical components so that a web browser might fetch a webpage or a new VRML-97 file from the Internet when the user clicks on the specific graphical component. Animations, sounds, lighting, and other aspects of the virtual world can interact with the user or may be triggered by external events such as timers. VRML-97 files are commonly called *worlds* and have the *.wrl extension.

The definition of BIFS as a part of the system information is one of the biggest steps forward of MPEG-4 with respect to its predecessors. It gives a flexible way to manipulate various types of audio-visual media in an MPEG-4 interactive scene, to schedule events, to coordinate multimedia objects both in the temporal and spatial domains, and to process interactivity. BIFS is a very efficient, binary encoded version of an extended set of VRML97 nodes. The downside of being a generic tool is that BIFS cannot completely exploit the redundancy of specific data such as meshes or animation curves. To overcome this, MPEG-4 defines methods for different kinds of graphics primitive, e.g., shapes, textures and animations [36].

The penetration of COLLADA in the market, as well as the adoption of XML for the new version of VRML, X3D, made the MPEG committee issue the new Part 25 of the MPEG-4 set of specifications, with the goal of offering an architectural model able to accommodate a third-party XML description of the scene graph with (potential) binarization tools and with MPEG-4 3D graphics compression tools. The advantages of such an approach are the use of powerful specific compression tools and the generality of graphics primitives' representation.

The MPEG-4 Animation Framework eXtension (AFX) is defined through the ISO/IEC 14496-16, which contains components for rendering geometry, textures, volumes, and animation organized around AFX conceptual organization of models for computer games and animation. AFX offers a unified standardized framework including a rich set of 3D tools, most of them already well-supported in the industry. Furthermore, advanced features such as compression, streaming, and seamless integration with other audiovisual media, allow building high quality creative cross media applications [37] [38].

ii. COLLADA

COLLADA (COLLABorative Design Activity) is an interchange file format for interactive 3D applications. It is managed by the nonprofit technology consortium, the Khronos Group, and it has been standardized by the ISO 17506.

COLLADA defines an open standard XML schema for exchanging digital assets among various graphics software applications that might otherwise store their assets in incompatible file formats. COLLADA documents that describe digital assets are XML files, usually identified with the .dae (Digital Asset Exchange) filename extension.

COLLADA supports all the features that modern 3D interactive authoring applications and DCC (digital content creation) tools need to exchange and fully preserve asset data and meta data. The COLLADA feature set is expanding to incorporate technologies such as packaging programmable shader effects and controlling real-time physics engines. COLLADA enables powerful content creation pipelines that can automatically condition and scale 3D geometry and texture assets for real-time playback on a wide diversity of platforms.

iii. X3D

ISO 19775-1:2008 defines eXtensible 3D (X3D), a royalty-free open standards file format and run-time architecture to represent and communicate 3D scenes and objects using XML. It provides a system for storing, retrieving and reproducing real time graphics content embedded in applications, all within an open architecture to support a wide array of domains and user scenarios. X3D features extends the ones of VRML-97, providing also enhanced APIs.

X3D is designed for applications where 3D models and behaviors can best illustrate the spatial relationships and interactive features that would be otherwise difficult to show. Example applications quite remarkably include training for Arabic language and cultural interaction. X3D provides a wide variety of capabilities used by the 3D community for rendering, texturing, modeling, animation, and user interactivity. Other advanced X3D components include geospatial positioning, interchangeable humanoid animation bodies, and the IEEE Distributed Interactive Simulation (DIS) network protocol [39].

iv. Considerations

X3D seems the most interesting standard file format revised in this brief section, since it is both open and provides extended APIs. It is thus envisaged for the SIGN-HUB digital platform to evaluate the possibility of interaction with it for supporting future possible developments concerning 3D virtual signing characters; nevertheless, again, this is not a strict requirement for the SIGN-HUB project.

4. Guidelines to Produce and Publish High-Quality Video Content

This section summarizes guidelines that should be addressed by the content providers involved within the SIGN-HUB project (with particular regards to Working Package WP2) to ensure that the digital video content they produce is compliant with the most recent and highest standards for long-term video preservation and video accessibility.

a. Naming Conventions

Although it may sound too simplistic, and even if data findability is practically ensured via metadata rather than via file names, still a naming convention to represent and store digital video content, i.e., the structure of folders and file names that will be used to organize data, should be enforced throughout the whole project to maintain the digital archive consistent and to ease the tasks of data harvesting and batch processing. In particular, following a naming convention will benefit the project with consistency, convenience, and clarity.

File names should be made up of made up of 3 separate but equally important parts, plus clearly an extension that should directly make clear which data format are we dealing with:

The first item should be a clear but short description about the content of the file, targeting “who” or “what” is represented in the file, using clear keywords and conveying enough meaningful information to someone unfamiliar; it could be standardized among the partners of the SIGN-HUB project by specifying the task the video corresponds to (first two letters for grammar, atlas, assessment, interviews), the author of the video (initial of the name and of the surname), and information from the participant in form of gender (first letter for female or male), age (2 digits), from Hearing family (Y/N), and also if possible if in conversation activity (Y/N). If the file is within a series, additional words or numbers may be used to identify the file within its series: for instance, an interview might have the same content, but recorded from 2 different perspectives, one with the interviewee and the other video with both the interviewee and the interviewer, could be represented as

- firstitem-1-YYYYMMDD-HHmm-Version.format
- firstitem-2-YYYYMMDD-HHmm-Version.format;
- The second item should be a clear indication about when the digital content has been produced-digitized (or retrieved, for instance from the Internet): it should be a full date in format YYYYMMDD-HHmm, with 0 padding used to cover for single digit number and year first to be able to sort every single file chronologically; if (some of) this information is missing, X padding may be used too;
- The third item should be a versioning reference, to be able to reconstruct changes and versioning to the file.

One clear example may be: Project-Experiment-Scientist-YYYYMMDD-HHmm-Version.format (concretely: Atlantis-LakeMeasurements-Smith-20180113-0130-v3.csv). It is of outmost importance to be consistent throughout the whole SIGN-HUB digital archive. Additionally, for multimedia digital content such as video, a fourth element could be added to the file name to clearly state which codec and settings have been used to produce that file.

As a rule of thumb, the following constraints should apply:

- Keep file names short, but comprehensible, using commonly recognized acronyms or abbreviations where appropriate
- Order the elements in a file name in the most appropriate way to retrieve the record, with the most important element first
- Avoid repetition and redundancy
- If it is important that documents can be sorted by date, number or name, to ensure correct sort order, numbers below 10 should be given in two digits, and personal names with surname followed by initials without spaces
- Avoid non-alphanumeric characters, such as ? ; : / \ < > * & \$ £ + =. Hyphens may be used
- When saving items such as digital photographs and scanned images, the title should be changed from the system-generated number to a something meaningful
- A description of the application should not be included in the document title, as it should be apparent from the file extension

- Words describing the form or format of a document, such as 'draft', 'letter', 'presentation', 'spreadsheet', should not be used within file names.

b. Content Uploading on Online Digital Platforms for Researchers

Given the important necessities of the researching communities, in the recent years many online platforms have been developed to ease the collaborative exchange and management of data, often a crucial task in large and ambitious research projects. Some of existing digital platforms for researchers are legacies of successful European-funded projects. For the purposes of SIGN-HUB, concerning long-term preservation of digital video content, resorting to an existing platform would represent a considerable speed-up. Nevertheless, the pricing model should be evaluated, as well as the features offered for accessibility (e.g., video streaming), and the amount of storage space offered (SIGN-HUB aims at producing several dozens of TB of material).

One example of the case is DARIAH³. It is a digital infrastructure for arts and humanities which was born to support digital research as well as the teaching of digital research methods. DARIAH should, up to date, provide digital tools to ease the task of data sharing among researchers. However, it is not clear whether it is suitable and/or optimized for visual multimedia content and how people outside the DARIAH member group can access data on the platform. Contacts between SIGN-HUB and DARIAH have started to identify whether DARIAH may represent a valid solution for implementing, within SIGN-HUB, a repository ensuring not only long-term data persistence of video content but also their findability, accessibility, interoperability, and reusability (as for the FAIR principles and as for what is better detailed in deliverable D3.2).

Another platform thought for researchers is CLARIN⁴. Actually, CLARIN was born mainly for the challenges raised from the community of researchers in linguistic fields. It provides a wide research infrastructure that was initiated from the vision to make all digital language resources and tools from all over Europe and beyond accessible through a single sign-on online environment for the support of researchers in the humanities and social sciences. For what concerns data storage and sharing, the platform offers two services. The first is a data storage service⁵, but it seems to be not managed directly from the platform rather from one of the project partners. The second is a search service specifically thought for metadata⁶, which however offers also the possibility to upload the actual data in addition to the metadata. These drawbacks may pose major issues when a large amount of visual data content should be produced in an agile way and released to a broad audience in an easy and user-friendly way. Contacts between SIGN-HUB and CLARIN have started, and negotiations are in an advanced state, to identify whether DARIAH may represent a valid solution for implementing, within SIGN-HUB, a repository ensuring not only long-term data persistence of video content but also their findability, accessibility, interoperability, and reusability (as for the FAIR principles and as for what is better detailed in deliverable D3.2).

Another promising digital platform for researchers seems to be EUDAT⁷. EUDAT platform was born after a project aiming at enabling European researchers from any research discipline to preserve, find, access, and process data in a trusted environment, conceived as a network of collaborating, cooperating centers. As of today, EUDAT offers common data services, supporting multiple research communities as well as individuals, through a geographically distributed, resilient network of 35 European organizations. These shared services and storage resources are distributed across 15 European nations and data is stored alongside some of Europe's most

³ <http://www.dariah.eu/>

⁴ <http://www.clarin.eu/>

⁵ <https://www.clarin.eu/content/depositing-services>

⁶ <https://lindat.mff.cuni.cz/repository/xmlui/page/deposit>

⁷ <https://www.eudat.eu/>

powerful supercomputers. Covering both access and deposit, from informal data sharing to long-term archiving, and addressing identification, discoverability and computability of both long-tail and big data, EUDAT's services address the full lifecycle of research data.

The most interesting service is named B2SHARE⁸, and it is advertised as an enabler for sharing small-scale research data from diverse contexts. In particular, it is a web-based service for publishing, storing and sharing small data sets, intended for European scientists. The service stores the data at a trusted repository with national backing, to provide a professionally managed and supported IT environment. However, it is thought with the purpose of allowing registered users to upload small scientific data collections, thus being probably inadequate to store large collection of visual materials and multimedia content; in addition, it cannot guarantee long-term persistence of their locally stored data and lack means to easily share such data with colleagues worldwide. This represents the main drawback of research-oriented platform: with respect to commercial ones, they can offer much fewer certainty in terms of data persistence over time. So, it is still not possible to state if EUDAT will be of relevance for the purposes of the SIGN-HUB project. For instance, video streaming does not seem to be a standard service offered by B2SHARE. Nevertheless, contacts between SIGN-HUB and EUDAT have started to identify whether EUDAT may represent a valid solution for implementing, within SIGN-HUB, a repository ensuring not only long-term data persistence of video content but also their findability, accessibility, interoperability, and reusability (as for the FAIR principles and as for what is better detailed in deliverable D3.2).

With the idea of providing the content providers within the SIGN-HUB project, particularly concerning Working Package WP2, with a generic example and guidance on how to submit digital data content to an on-line repository, in the following a small guide on how to submit digital content to EUDAT B2SHARE is given. Users must register to upload and share data. To upload video on EUDAT, a registered user should sign in to the platform and the B2SHARE service, under the "Deposit" section. Then, he/she should select and upload the video resource by means of the drag and drop interface shown in Figure 5.

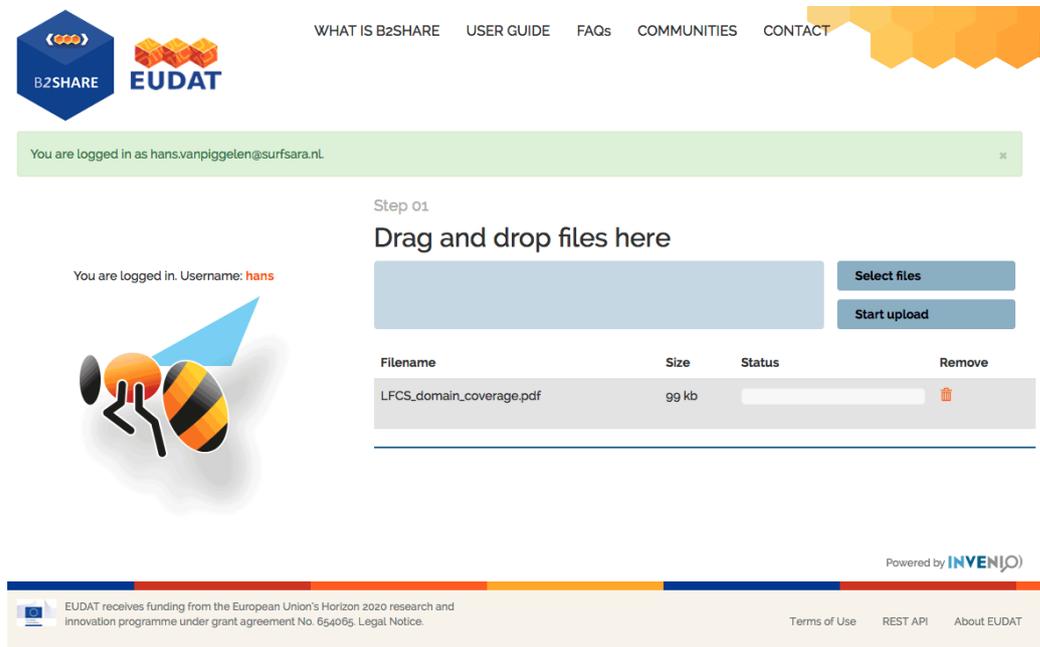


Figure 5 - Drag&Drop interface on B2SHARE [adapted from <https://b2share.eudat.eu/docs/b2share-guide/>]

⁸ <https://www.eudat.eu/services/b2share>

As a second step, the user should indicate a domain or project-specific metadata set to describe the video resource, by means of the interface shown in Figure 6.

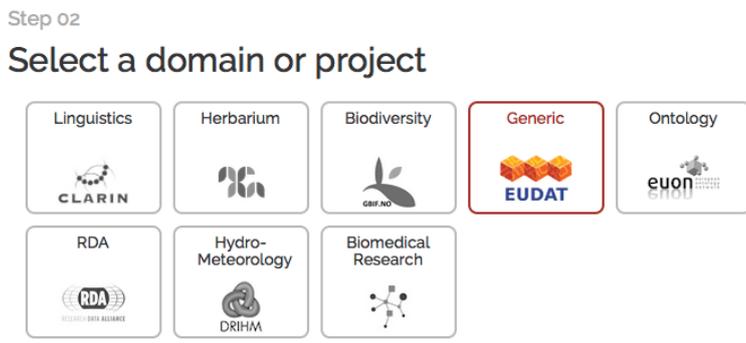


Figure 6 - Domain selection interface on B2SHARE [adapted from <https://b2share.eudat.eu/docs/b2share-guide>]

As a third and final step, the user should fill in the metadata form, and to choose the license. Metadata is useful to allow data indexing and retrieval; the license influences how data is distributed and reusable from the research community and the general audience. Licenses are selected from a menu, as in the screenshot in Figure 7.

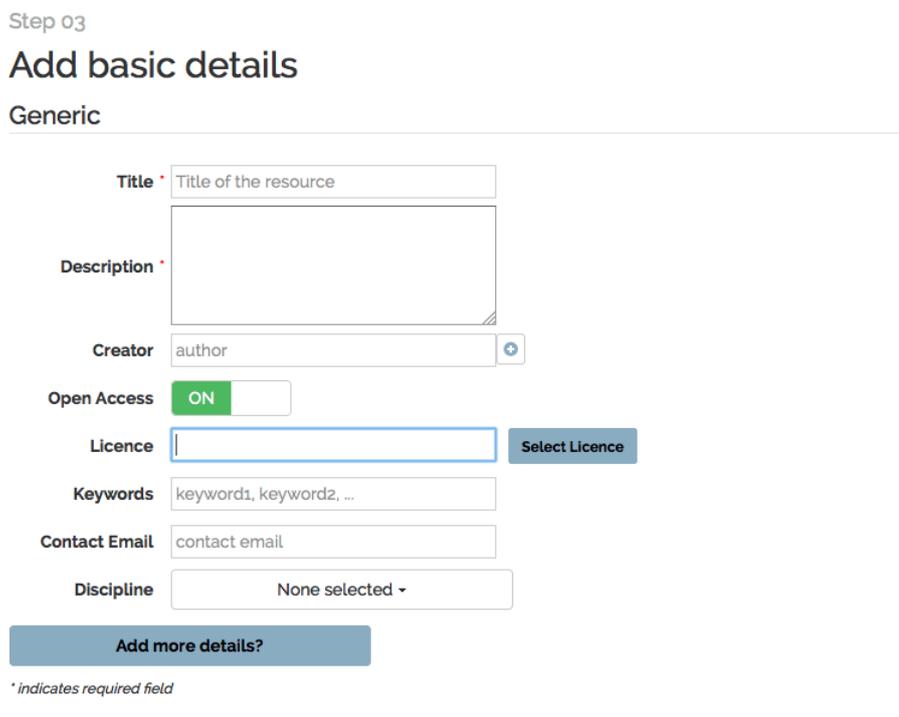


Figure 7 - Metadata interface on B2SHARE [adapted from <https://b2share.eudat.eu/docs/b2share-guide>]

c. Recommendations for the SIGN-HUB project

Summarizing what has been stated throughout this document, concerning digital content and video files in particular, the needs of the SIGN-HUB project are double fold.

On the one side, it is needed to create a digital archive of video to preserve the cultural heritage of signing communities spread all over Europe-the world. With this respect, researchers and content providers are strongly encouraged to produce high-quality content, by:

- digitizing old content (e.g., tapes) with the highest quality hardware and software possible;
- recording video with the highest quality options provided by their cameras.

On the other side, it is needed to make videos within this digital archive accessible through web browsers and thus HTTP-base streaming. With this respect, content providers are strongly encouraged to produce lower-quality versions of their videos, to make the task of streaming easier.

For persistence of media streams, it is always wise to not use lossy compressed formats since one does not know what future generations will want to do. From lossless compressed files, other formats can be produced all the time dependent on the needs and converters are available.

For video streaming, one will be confronted with the requirements of streaming server technology and bandwidths. For pure presentation purposes, it does not matter whether H.264 or H.265 are being used for example, since the occasional user will hardly see that in H.265 static areas are replaced by stored patterns, i.e. the material is not authentic in the true sense.

In this, the streaming technology is changing rapidly: Flash was the standard up to very few years ago, being now instead summoned by HTML5. So, the SIGN-HUB partners should consider to select a repository implementing the digital archive where to host two versions for each of the video files:

- one raw master file for archiving with high-quality, lossless if possible (Motion JPEG 2000, MPEG lossless, H264 lossless);
- derived from that one a second file, of lower quality, that can be easily streamed via HTML5 (e.g., H264/mp4): this second file may be generated on-the-fly or pre-generated, and this choice depend on the features offered by the chosen repository.

To evaluate the suitability of digital audio and video formats for records that need to be retained over the long-term (i.e. over multiple generations of technology), the researchers in SIGN-HUB should consider the following aspects:

- the format should be publicly and openly documented;
- the format is non-proprietary;
- the format is in widespread use;
- the format is self-documenting;
- the format can be opened, read, and accessed using readily-available tools.

In this, it is difficult for this document to issue formal recommendations for digital audio and digital video file due to the rapidly evolving nature of digital audio and video formats and the lack of any open, national or international consensus standards for the creation and preservation of digital audio and video.

Nevertheless, we can state that we recommend video to be recorded at least for a minimum of 720x486 pixels at 30 frames per second, with full-range color (at least 8 bits per channel with 10 bits per channel preferred), with audio tracks of digital video recordings at at least 48KHz. It is suggested to avoid the following digital video files as they are known to have insufficient quality for archival retention:

- Files created for "streaming" broadcast (e.g. RealAudio, Windows Media) - these files sacrifice quality for file size. They are often reduced from high-resolution files by removing high and low frequency ranges and compressing the remaining signal;
- Reference files of lower quality than original (e.g. for web site use) - again, this sacrifice quality for file size;
- Files that have been transcoded. In other words, it is not acceptable to render previously encoded content through another codec.

Concerning the production and optimization of videos for their storage, thus being interested in maintaining their highest definition possible, which for the purposes of the project is of interest to store local copies of the multimedia content (e.g., interviews to elderly signers) for further research and analyses, we recommend storing information with the highest performant encoding possible being able to guarantee outcomes in very high definition. In fact, the need here is to store the highest quantity possible of the information within the videos for their further processing.

Lossless video compression is of particular interest for the SIGN-HUB project for what concerns content preservation and specifically long-term archiving of video content; in fact, the SIGN-HUB project aims at providing researchers with a plethora of videos forming a digital archive of sign languages, which are meant to be fully findable and usable for many decades in the future, even possibly enabling research scenarios which are not even known by today. It is therefore important to ensure that no feature nor information of the video stored is lost. The choice on which codec and format to use, at the end, should be negotiated with the service providers (e.g., CLARIN) and it is demanded to the in-depth description given in deliverable D3.2. Nevertheless, it is safe to admit that Motion JPEG 2000 and FFV1, which is also royalty free and used in many digital archives for humanities and linguistics, are good choices for the purposes of the project.

Concerning the production and optimization of videos for their streaming within the online platform of the SIGN-HUB project, which for the purposes of the project is of interest to publish online content and browse it, we recommend storing information with the highest performant encoding possible ensuring the best compromise between the size and the quality of the encoded result. In fact, the need here is to transmit the lowest quantity of the information within the videos allowing the proper video reproduction and not compromising the overall user experience, using the lowest possible bandwidth.

Lossy video compression is of particular interest for the SIGN-HUB project for what concerns content distribution and specifically video streaming; in fact, the SIGN-HUB project aims at providing end users with a plethora of videos forming a digital archive of sign languages, which are meant to be fully accessible and interoperable. With modern technology and network capabilities, it would be meaningless to resort to lossless video compression for video streaming. In this, HTML5 is representing the modern standard-de-facto for video streaming for web browsers. H.264 is currently one of the most commonly used formats for the recording, compression, and distribution of video content. The intent behind the development of the H.264 codec was to create a standard capable of providing good video quality at substantially lower bit rates than previous standards, without increasing the complexity of design so much that it would be impractical or excessively expensive to implement. The choice on which codec and format to use, at the end, should be negotiated with the service providers (e.g., CLARIN) and it is demanded to the in-depth description given in deliverable D3.2. Nevertheless, it is safe to admit that implementing video streaming with H.264/MP4 files is a good choice for the purposes of the project. After the overview given on codecs and containers for multimedia video content, it is therefore safe to state that to produce and deliver a high-quality, well compressed and performing content the most reasonable choice is to rely on the H.264 codec, with lossy compression, and on the MP4 container. In addition, H.264 is the only compression technology that plays on all computers, mobile devices, and OTT players. This makes producing high-quality H.264 files very convenient, as they are suitable for a large plethora of devices and recipients.

Concerning the choices on which repository and standard formats are recommendable to make SIGN-HUB data findable, accessible, interoperable and reusable, together with the more in-depth discussion given in the deliverable D3.2, it is safe and technical sound to state that SIGN-HUB should leverage on standards that are:

- XML-based metadata (e.g., OLAC, CMDI);
- standards for multimedia (MPEG+ suitable container/codec), non-interlaced videos;
- off-the-shelf repositories (e.g., DSpace, Fedora Commons, Islandora);
- strict naming conventions, which can help a lot on the side of the archive platform (e.g., when dealing with thousands of files), while they can also be very convenient when performing batch operations, e.g., when dealing with online/offline files.

To generate, process, and retrieve metadata, it would be needed and suggested to use the built-in registration system of the repository chosen; as an alternative, a manual option is to gather the metadata, have early quality/validity checks (using webform, or at the worst case spreadsheets such as Microsoft Excel), then convert it to standard format and offer it via, for instance, jOAI. As of today, metadata within SIGN-HUB project have already been gathered by means of Excel spreadsheets, given that content providers from Working Package WP2 have already started providing content even though the digital platform to host them is not ready yet, and it was necessary to not lose information about the content provided. Excel has thus been used as the quickest and easiest temporary solution. We are already negotiating with the technical managers of CLARIN the possibility to send them our validated XLS-CSV files so that they can convert them into well-formed and organized hierarchical XML files (batch import). At the end, it will be needed to have metadata represented in some XML-based file, following a schema that is particularly suitable for representing linguistic content, from which information can be even visualized through a dedicated web application (e.g., as it already happens in CLARIN).

With the idea of providing the content providers within the SIGN-HUB project, particularly concerning Working Package WP2, with a generic example and guidance on how to convert a video file into the lossless H.264/MP4 video file format, the following short guide is given. It resorts on VLC. VLC is a free and open source cross-platform multimedia player and framework that plays most modern multimedia files as well as data streams from input devices. Resorting on VLC, it would be sufficient to connect the input device to the PC equipped with the tool (possibly through a USB interface) and open the tool itself; by navigating its menu under the *Media* voice, there is the option to convert or save multimedia contents. By clicking on this voice, a menu like the one shown in Figure 8 appears. On this menu, it is typically necessary just to control that the input device (both for video and audio stream) has been correctly identified. Default parameters are typically good enough and dynamically adapted, but they can be tuned through the *Advanced options* button.

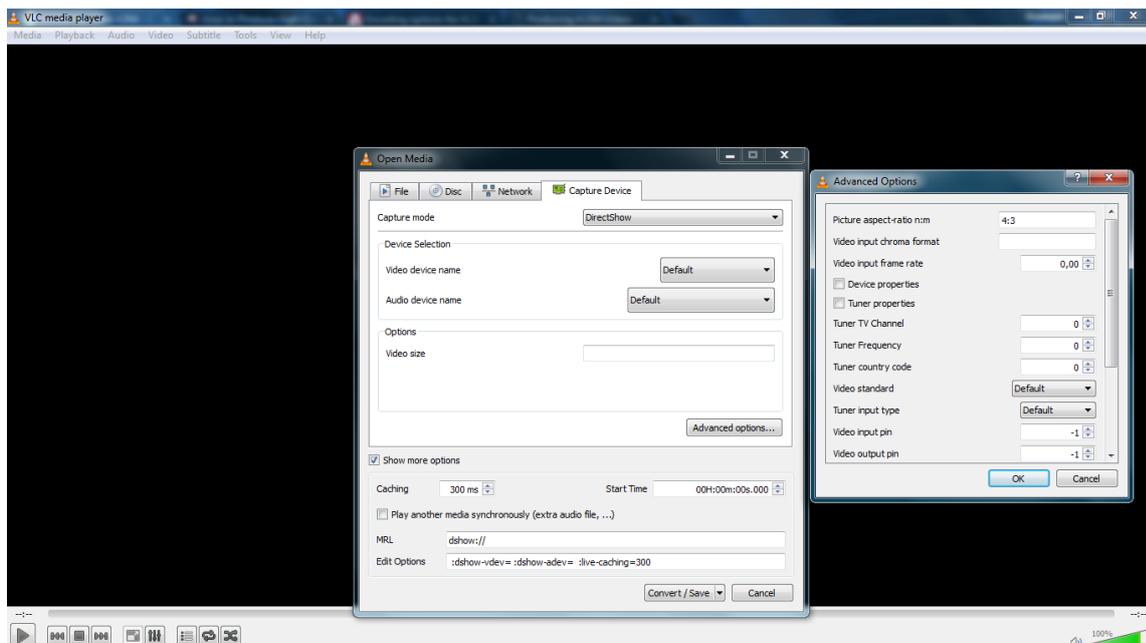


Figure 8 - VLC screenshot to produce video content from external input stream

Then, after having clicked on the *Convert/Save* button, a second menu appears (shown in Figure 4) that allows the user to define the codec and to browse the destination for the outcome of the encoding procedure. Let's choice the H.264 codec within a MP4 file (MP3 will be highlighted as best choice for the hardware audio codec) and proceed with the procedure by means of the *Start* button. A progress bar will follow its development and signal when the outcome is ready. At the end of the procedure, all the content registered from the input device will be encoded on a single high-quality MP4 file. Also in this case, it is typically safe to leave untouched the default parameters, even though they can be accessed by means of the settings button. As shown in Figure 9, it is highlighted that the output file will support most of the existing features (excluding the subdivision in Chapters).

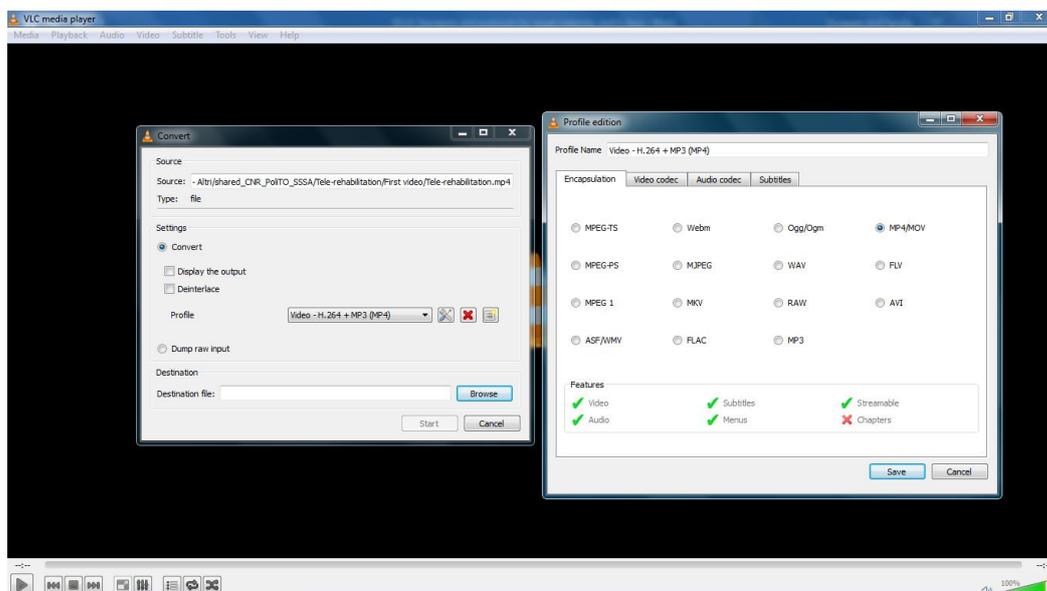


Figure 9 - VLC screenshot to encode video content with the H264 codec and the MP4 container

5. Conclusion

The Deliverable 3.11 gives an overview of the existing standards, procedures and technologies that are typically adopted to represent, and thus store and retrieve, multimedia contents ranging from written text to video.

A digital audio or video file consists of a container holding source data which has been processed through a codec. A codec (coder-decoder, compressor-decompressor, compress-decompress) transforms the analog signal (from a microphone, video camera, etc.) into the ones and zeros of a digital file. A codec also can be used to encode material already existing in digital format into another digital format.

There are two broad categories of codecs:

- **Lossy Codec:** A lossy codec is one that discards certain portions of the signal in order to achieve a smaller file size; for example, mp3 codecs attempt to identify and remove portions of the signal that would not result in a perceived loss of quality of the sound to make the file smaller. These losses are sometimes noticeable and sometimes not. The more aggressive the compression setting chosen, the more data is removed. This can result in digital artifacts, which are audible errors created by the compression.
- **Lossless Codec:** A lossless codec is one that achieves smaller file sizes through means other than removing data. This can include using a variable bit rate which would use fewer bits to encode silences as compared to an active section of music.

Some codecs may be configured to be either lossy or lossless.

According to the project objectives related to the population of the SIGN-HUB software platform with multimedia content, we especially focused on standards and technologies that can be worth for the SIGN-HUB project. Therefore, it is provided a deep analysis of most used standards, technologies and software highlighting, when possible, pros and cons. In addition, this deliverable provides a hands-on tutorial dedicated to the production of high-quality multimedia video content and its upload on online platforms. Moreover, a short description about standards to represent tridimensional figures and virtual characters is also provided, as one of the features of the SIGN-HUB software platform will be a fully compatibility with signing avatars. Finally, the task of creating and sharing high-quality video content is debated in the last part of this document.

We strongly recommend to leverage on existing repositories for ensuring long-term data persistence, as stated in D3.2. In particular, we have already contacted the responsible people from both the EUDAT and the DARIAH network. Then, many discussions have already been taken with the CLARIN ERIC network; a dedicated workshop has been hosted with its technical director, dr. Dieter Van Uytvanck, to discuss the details of the integration of CLARIN resources within the SIGN-HUB project (e.g., pricing model) concerning the usage of CLARIN repositories and the possibilities from CLARIN centers to host SIGN-HUB data. The definitive statements about which platform to use, if any, and which services are to build custom within SIGN-HUB will be made before December the 6th 2017, as described in deliverable D3.2. Therefore, it will be also negotiated with them within this time which are the video codecs and containers, and the standard for metadata and annotations supported and thus to use in the project. Conversely, if none of this platform will be chosen by the project's management, we will build a custom repository, compliant with the analysis and supporting the technologies provided in this document.

All the guidelines, procedures, and standards discussed in this deliverable have been thoroughly discussed with the scientific partners of the project during many occasions and meetings, and have been agreed with them.

Bibliography

- [1] R. Jakobson and J. Mepham, *Six lectures on sound and meaning*. Harvester Press Brighton, 1978.
- [2] N. P. Himmelmann, "Language documentation: What is it and what is it good for?," in *Essentials of Language Documentation*, 2006, pp. 1–30.
- [3] N. I. S. O. National Information Standards Organization (U.S.), *Understanding metadata*. NISO Press, 2004.
- [4] J. Van Vliet, "Text processing and document manipulation: Proceedings of the International Conference on, 14-16 April 1986," in *Text processing and document manipulation*, 2009.
- [5] G. Rockwell, "What is text analysis, really?," *Lit. Linguist. Comput.*, vol. 18, no. 2, pp. 209–219, 2003.
- [6] G. Salton, *Automatic Text Processing The Transformation, Analysis, and Retrieval of Information by Computer*. .
- [7] A. McKee, "Textual Analysis: A Beginner's Guide (McKee)."
- [8] A. Caimi, "Audiovisual Translation and Language Learning: The Promotion of Intralingual Subtitles," *J. Spec. Transl.*, no. 6, pp. 85–98, 2006.
- [9] P. Winke, S. Gass, and T. Sydorenko, "The effects of captioning videos used for foreign language listening activities," *Lang. Learn. Technol.*, vol. 14, no. 1, pp. 65–86, 2010.
- [10] N. Zanon, "Using subtitles to enhance foreign language learning," *Porta Linguarum*, vol. 6, pp. 41–52, 2006.
- [11] Business Dictionary, "Markup Language." [Online]. Available: <http://www.businessdictionary.com/definition/markup-language.html>. [Accessed: 16-Jun-2016].
- [12] J. H. Coombs, A. H. Renear, and S. J. DeRose, "Markup systems and the future of scholarly text processing," *Commun. ACM*, vol. 30, no. 11, pp. 933–947, Nov. 1987.
- [13] T. Bray, "On Semantics and Markup, Taxonomy of Markup," www.tbray.org/ongoing, 2003.
- [14] W. Wohler, *SGML Declarations*. 1998.
- [15] C. F. Goldfarb, *The Roots of SGML %93 A Personal Recollection*. 1996.
- [16] T. Berners-Lee, *Re: SGML/HTML docs, X Browser (archived www-talk mailing list post)*. 1991.
- [17] C. F. Goldfarb, *The SGML Handbook*. 1990.
- [18] I. . Hickson, Ian (Google, "HTML5: A vocabulary and associated APIs for HTML and XHTML," *W3C Recomm.*, pp. 1–34, 2014.
- [19] S. Web, *HTML Mastery : Semantics , Standards*. 2006.
- [20] D. Raggett, A. Le Hors, and I. Jacobs, *HTML 4.01 Specification*, vol. 24, no. December. 1999.
- [21] T. Berners-Lee, "Information Management: A Proposal," *Word J. Int. Linguist. Assoc.*, vol. February 2, no. May, pp. 1–10, 1989.
- [22] B.-L. T and C. D, "Hypertext Markup Language - 2.0," *HJP J.*, p. 4, 1995.
- [23] W3C, "HTML5," 2012. [Online]. Available: <https://www.w3.org/TR/2012/CR-html5-20121217/>. [Accessed: 16-Jun-2016].
- [24] T. Bray, J. Paoli, E. Maler, and S. Microsystems, "Extensible Markup Language (XML) 1.0 (Fifth Edition)," *W3C Recomm.*, vol. 0, pp. 1–37, 2008.
- [25] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau, "Extensible markup language (XML)," *World Wide Web J.*, vol. 2, no. 4, pp. 27–66, 1997.
- [26] W3C, "Document Object Model (DOM)," 2005. [Online]. Available: <http://www.w3.org/DOM/#what>. [Accessed: 19-Jun-2016].

- [27] Webopedia, "SAX." [Online]. Available: <http://www.webopedia.com/TERM/S/SAX.html>. [Accessed: 19-Jun-2016].
- [28] Oracle, "Simple API for XML." [Online]. Available: <http://download.oracle.com/javase/1.4/tutorial/doc/JAXPSAX.html>. [Accessed: 19-Jun-2016].
- [29] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, 2003.
- [30] J. Watkinson, "MPEG video compression," in *The MPEG Handbook (Second Edition)*, 2004, pp. 230–352.
- [31] R. J. Clarke, *Digital compression of still images and video*. Academic Press, Inc., 1995.
- [32] T. Sikora, "The MPEG-4 video standard verification model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 1, pp. 19–31, 1997.
- [33] I. E. Richardson, *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia*. John Wiley & Sons, 2004.
- [34] E. S. Carotti and J. C. De Martin, "Lossless video coding using multi-frame motion compensation," 2005.
- [35] K. Mamou, T. Zaharia, and F. Prêteux, "Famc: The MPEG-4 standard for animated mesh compression," in *Proceedings - International Conference on Image Processing, ICIP, 2008*, pp. 2676–2679.
- [36] M. Preda, I. Arsov, and F. Moran, "COLLADA + MPEG-4 OR X3D + MPEG-4," *IEEE Veh. Technol. Mag.*, vol. 5, no. 1, pp. 39–47, 2010.
- [37] M. Bourges-Sévenier and E. S. Jang, "An introduction to the MPEG-4 animation framework extension," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 7, pp. 928–936, 2004.
- [38] J. D. Funge, *AI for games and animation: a cognitive modeling approach*. AK Peters Massachusetts, 1999.
- [39] L. Daly and D. Brutzman, "X3D: Extensible 3D graphics standard [Standards in a Nutshell]," *IEEE Signal Process. Mag.*, vol. 24, no. 6, pp. 130–135, 2007.
- [40] T. Cisco, "Cisco Visual Networking Index : Global Mobile Data Traffic Forecast Update , 2010 – 2015," *Growth Lakel.*, vol. 2011, no. 4, pp. 2010–2015, 2011.
- [41] H. D. Wactlar and M. G. Christel, "Digital video archives: Managing through metadata," *Build. a Natl. Strateg. Digit. Preserv. Issues Digit. media Arch.*, vol. 84, p. 99, 2002.
- [42] CISCO, "The Zettabyte Era: Trends and Analysis," *Cisco*, no. May 2015, pp. 1–29, 2015.
- [43] Y. Liu, Y. Guo, and C. Liang, "A survey on peer-to-peer video streaming systems," *Peer-to-Peer Netw. Appl.*, vol. 1, no. 1, pp. 18–28, 2008.
- [44] D. Wu, Y. T. Hou, W. Zhu, Y. Q. Zhang, and J. M. Peha, "Streaming video over the internet: Approaches and directions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 282–300, 2001.