Deliverable 5.1 LinkedTV Platform and Architecture

Rolf Fricke, Jan Thomsen

18.4.2012

Work Package 5: LinkedTV platform

LinkedTV

Television Linked To The Web

Integrated Project (IP)

FP7-ICT-2011-7. Information and Communication Technologies

Grant Agreement Number 287911

Dissemination level[1]	*PU*
Contractual date of delivery	*31st March 2012*
Actual date of delivery	*18th April 2012*
Deliverable number	*D5.1*
Deliverable name	*LinkedTV Platform and Architecture*
File	*LinkedTV_D5.1_Platform_and_Architecture.docx*
Nature	*Report*
Status & version	*V1.0*
Number of pages	*85*
WP contributing to the deliverable	*WP5*
Task responsible	*CONDAT*
Authors	*Rolf Fricke, Condat* *Jan Thomsen, Condat*
Other contributors	*Pieter van Leeuwen, Noterik*
Reviewer	*Raphaël Troncy, EURECOM*
EC Project Officer	*Manuel Carvalhosa*
Keywords	*Distributed Software Architecture, Service Oriented Architecture, Video Analysis, Semantic Enrichment, Link Open Data, Contextualization, Personalization, Hypervideo Player*
Abstract (for dissemination)	*The objective of Linked TV is the integration of hyperlinks in videos to open up new possibilities for an interactive, seamless usage of video on the Web. LinkedTV provides a platform for the automatic identification of media fragments, their metadata annotations and connection with the Linked*

[1] • 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)

	Open Data Cloud, which enables to develop applications for the search for objects, persons or events in videos and retrieval of more detailed related information. *The objective of D5.1 is the design of the platform architecture for the server and client side based on the requirements derived from the scenarios defined in WP6 and technical needs from WPs 1-4. The document defines workflows, components, data structures and tools. Flexible interfaces and an efficient communications infrastructure allow for a seamless deployment of the system in heterogeneous, distributed environments.* *The resulting design builds the basis for the distributed development of all components in WP1-4 and their integration into a platform enabling for the efficient development of Hypervideo applications.*

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Objectives of LinkedTV

There are still important gaps and breaks in the current “Networked Media Web”. TV-over-Web does not yet make full use of the interaction possibilities inherent in the Web model, choosing rather to shift the passive watching experience to the screen of a computer (or smartphone). Web-over-TV is either about putting a Web browser onto the TV screen or overlaying parts of the Web over the TV signal; neither explore how one could browse between TV and Web content in a richer and more seamless manner. The vision of the realisation of a truly Networked Media Web is much more challenging: it means to provide audio-video information on the Web usable in a way similar as text based information is used today: interlinked with each other, with other kinds of information, searchable, and accessible everywhere and at every time.

Hypervideo has been pursued for quite a while as an extension of the hypertext approach towards video information. But it needs complex video analysis algorithms and is still an issue of research. Television Linked To The Web (LinkedTV) aims to provide a novel practical approach to Networked Media based on four phases: (1) annotation, (2) interlinking, (3) retrieval, and (4) presentation including personalization, filtering.

LinkedTV will allow us to seamlessly connect multimedia content on the Web by integrating networked media analysis, personalisation and presentation technologies within an integrated and coherent framework. LinkedTV will develop a comprehensive methodology, as the basis for the LinkedTV platform and tools. This platform will enable the information management and usage in the annotation, interlinking, and personalization process. It supports the media production process in a novel and integrated way, and provides support to users by providing linked information, filtering for personal preferences, search capabilities, etc. The LinkedTV platform will support the whole Networked Media process from end to end in an integrated and coherent way. This will be supported by building on the results of FP6 and FP7 projects on context analysis and representation, multimedia analysis, annotation, adaptation and delivery.

1.2 Objectives of the document

The objective of WP5 “Platform development and integration” is to provide a highly integrated, open, and manageable end-to-end platform. The platform should support and integrate all components and services needed for video/audio and Web content analysis, metadata generation and storage, creation of clickable video and user interfaces, and their delivery and presentation to the user.

The objective of this document *D5.1 LinkedTV Platform and Architecture* is to lay the foundation for the LinkedTV platform by defining the general architecture, which covers the definition of the following system configuration, design criteria and tasks.

System configuration

D5.1 defines the overall system architecture consisting of the:

- **Integrating platform for the end-to-end functionality:** supporting the backend as well as frontend functionality developed in WPs 1-4 with the workflows, components, interfaces and data flows needed for the Hypervideo applications at the frontend, which access functions and media resources from the backend.
- **Backend components** for video analysis and enrichment to build up the repository as well as for browsing, search and retrieval for the end user
- **Supporting client components for all variants** needed for the Hypervideo player and specific applications for the end user. The architecture considers to connect different client types and provides at least 2 variants for the project:
  - PC-based Web browsers client using IP networks and
  - TV-Set clients using HbbTV transmission.

General design criteria

The main criteria for designing the architecture have been to:

- consider general **requirements** for the envisioned platform with special regard on the 3 planned use cases and exploitation goals
- ensure **scalability** of the platform deployment for an extended amount of AV materials and number of end users
- regard general quality criteria such as **performance, usability and easy maintenance**.
- Validate that all algorithms and data flows work **correctly** and **smoothly** together
- Ensure **openness** to enable integration of third party components or connect the platform to the environment of content providers
- foresee a **flexible** deployment in heterogeneous environments
- keep **conformance** to existing or emerging **standards** and reliable open source frameworks and products especially from other EC co-funded projects
- support prevailing **coding standards** to use as much AV materials as possible.

Tasks

The definition of the platform includes the following tasks and goals:

- Gather and **consider all requirements** from Scenarios and WPs
- Design and specify the
  - Overall **workflows** to perform the needed analysis, enrichment and retrieval tasks
  - **Components** with the main interfaces, services and data input/output
  - **Communication** infrastructure and data flows to connect all components
  - **Tools** provided for the editor and developer

  - all other functional and non-functional features which have to be provided by the platform.
- Enable the **distributed development** of all components according to a clear integration and testing plan
- allow for a **repeated integration** of components in several cycles considering feedback from developers, editors and end users.
- plan the **evaluation** of the resulting system including a
  - verification, proving that the implementation is according to the functional specification.
  - validation, analysing whether the provided results are accepted by the end users; this includes a usability and performance assessment based on the scenarios of WP6.

1.3 Structure of the document

The document is organized as follows:

Chapter 2 comprises a survey and analysis of the requirements for the platform. This includes

- general requirements such as service orientation, scalability, performance, which will be derived by WP8 (exploitation) and WP5 itself and
- specific requirements from the various usage scenarios as defined by WP6 and the research oriented Work Packages concerning video analysis and segmentation (WP1), video annotation and metadata enrichment (WP2), user interaction models for Web and TV (WP3) and personalization and contextualization (WP4).

Chapter 3 describes the system configuration with the main components and actors, the platform architecture derived from the requirements analysis and the frameworks with their main services and interfaces.

Within Chapter 4 general administrative issues will be covered, such as guidelines and tools for development, issue tracking, component integration etc.

1.4 Status of the document

D5.1 has been worked out on the basis of an initial definition of features and scenarios in parallel to the refinement of specifications of the scenarios in WP6 and the technical WPs 1 to 4. Since many properties, functionalities, etc. required by the components are still subject to research and definition in their WPs, they are defined in D5.1 at the present stage on a very rough level. However, the general properties can already be outlined to a great extent.

With increasing maturity of the deliverables from WP1-4 the platform architecture will be refined and all interfaces and features can be specified in more detail. If the ongoing discussions of scenarios and features would lead to crucial changes of the system configuration, workflow or data structures, an appropriate adaption of D5.1 would be necessary.

D5.1 does not regard any constraints regarding IPR. The options for accessing and using remote videos or other data are considered only on a technical level.

1.5 Related LinkedTV deliverables

The design requires the consideration and input of almost all LinkedTV packages. However, D5.1 strongly relates to the following deliverables in particular:

- *D1.1 State of the art and requirements analysis for Hypervideo*
- *D2.1 Specification of the Media Fragment URI scheme*
- *D3.1 Specification of functionality requirements satisfying user information needs*
- *D4.1 Specification of user profiling and contextualization*
- *D6.1 Scenario descriptions*
- *D8.1 Exploitation plan for the project*.

1.6 History of the document

Table 1: History of the document

Date	Version	Name	Comment
2011/12/15	V0.01	Thomsen, Condat	Initial document structure
2012/1/5	V0.02	Fricke, Condat	Initial structure for section 4 with graphics for system configuration and architecture
2012/1/9	V0.03	Thomsen, Condat	Initial text Section 2 “Requirement Analysis” for 3 Use Cases
2012/1/11	V0.04	Fricke, Condat	Refinement Section 2 Requirement Analysis for Use Case UMons
2012/1/13	V0.05	Fricke, Condat	Specification Section 4 Communication Infrastructure
2012/1/16	V0.06	Fricke, Condat	Specification Section 4 Used Standards
2012/1/18	V0.07	Thomsen, Condat	Specification Section 4 Workflows for Analysis Layer
2012/1/23	V0.081	Fricke, Condat	Specification Section 4: Components of WP1-4, Deployment;
2012/1/28	V0.082	Thomsen, Condat	Specification Section 4 Workflows for Application Layer

Date	Version	Name	Comment
2012/2/2	V0.083	Fricke, Condat	Refinement Section 2 Requirement Analysis General Requirements
2012/2/5	V0.090	Thomsen, Condat	Refinement Section 2 Requirement Analysis for WP1-4
2012/2/19	V0.091	Thomsen, Condat	Refinement Section 1 Introduction / Purpose of the document, Section 4: Data Storage
2012/3/8	V0.092	Weinberg, Condat	Refinement Section 4: Communication Infrastructure, Ontologies
2012/3/12	V0.11	Fricke, Condat	Refinement Section 4: WP3 / WP4 components and interfaces
2012/3/18	V0.12	Thomsen, Condat	Refinement Section 4: WP1/WP2 components and interfaces, NER, output formats
2012/3/26	V0.121	Thomsen, Condat	Refinement Section 2: WP1 / WP2 components and interfaces
2012/3/29	V0.13	Fricke, Condat	Refinement Section 4: Workflows, WP3, WP4
2012/4/16	V0.13	Troncy, EURECOM	Review
2012/4/18	V1.0	Fricke, Thomsen, Condat	Integration of review suggestions and comments

1.7 Abbreviations and Acronyms

Table 2: Abbreviations

Abbreviation	Explanation
API	Application Programming Interface
ASR	Automatic Speech Recognition
GIS	Geographic Information System
JARSTOP	Java RDF Stored Procedure
JavaGAT	Java Grid Application Toolkit
KB	Knowledge Base
LOD	Linked Open Data
LORAPI	Low level RDF APIs

Abbreviation	Explanation
LRMS	Local Resource Management System
LSL	Link Specification Language
MPI	Message Passing Interface
NLP	Natural Language Processing
NER	Named Entity Recognition
NERD	Named Entity Recognition Disambiguation (Tool)
ORDI	Ontology Representation and Data Integration
OWL	Ontology Web Language
OWL	Web Ontology Language
P2P	Peer-to-peer
QoS	Quality of Service
RDF	Resource Description Framework
RDFS	RDF Schema
RDF	Resource Description Framework
RDFa	Resource Description Framework in attributes
REETON	RDF Engine Extension
REST	Representational State Transfer
RPC	Remote Procedure Call
SAWSDL	Semantic Annotations for WSDL
SSH	Secure Shell
SKOS	Simple Knowledge Organisation System
SOA	Service-Oriented Architecture
UI	User Interface
WSDL	Web Services Description Language
WYSIWYG	Acronym for ‘What You See Is What You Get’

2 Requirements Analysis

This section describes the requirements for the LinkedTV platform including

- The general requirements as set by the objectives of the envisioned platform
- The specific requirements resulting from the usage scenarios definitions (WP6)
- The specific requirements as set from the WP1-4 concerning video analysis and fragmentation, annotation and enrichment, user interface models, and personalization and contextualization
- Requirements concerning the later exploitation of the platform.

The formal identification of the requirements is done by numbering according to the different areas which have to be covered by the LinkedTV platform. The following table lists the areas:

Id	Requirement Area [2]
GEN	General properties and guidelines
ANL	Requirements concerning the video analysis (WP1)
FGM	Requirements concerning fragmentation of videos (WP1)
ANO	Requirements concerning the annotation of videos and video fragments (WP1,2)
ENR	Requirements concerning the enrichment of videos and fragments (WP2)
UIM	Requirements concerning general user interface models (WP3)
SCN	Requirements concerning the show case scenarios (WP6)
WEB	Requirements concerning web-based user interaction (WP3,4)
HBB	Requirements concerning HbbTV based user interaction (WP3,4)
PRS	Requirements concerning personalization and contextualization (WP4)

2.1 Priority

The requirements are ranked according a priority schema where “1” is the highest and “3” is the lowest priority.

[2] The list of requirements for scenarios and WPs is preliminary and reflects the current state of scenario descriptions and specifications.

2.2 General Requirements

The general requirements are derived from the project objective to create a platform which provides annotation of AV content, clickable video and non-linear browsing. This vision leads to the following needs for scalability, distribution of data and language support. These general needs have also to be fulfilled for the exploitation requirements, which foresee to address certain markets and a flexible deployment for the prevailing IT environments.

2.2.1 Standards compliance

Scenario ID	Title/Characteristics	Priority
REQ-GEN-001	The platform MUST support and be compliant to existing or emerging standards wherever applicable. The respective standards to be employed, in particular in case of alternative standardization efforts, are discussed and decided within the respective work packages. A list of relevant standards is given in the Appendix.	1

2.2.2 Open and Distributed Platform Design

REQ-GEN-002	LinkedTV should offer an open platform which is agnostic concerning specific server technologies and supports different delivery channels and client types.	1
REQ-GEN-003	The platform MUST be designed in a way that all functionalities can be accessed over an SOA infrastructure, which allows for a physical distributed allocation of components, data, applications and tools organized in a cloud-computing manner. The Service-oriented Architecture SHOULD be compliant with REST principles.	1
REQ-GEN-004	The platform should allow for an easy adaption of the system configuration and flexible communication between all instances. It should be possible to exchange existing platform components for analysis, enrichment and presentation as far as possible. This could be third party or open source, which offers additional value for the platform, e.g. because of improved methods. Possible candidates for a later replacement could be components for video-, voice- or text analysis, semantic enrichment, different Ontologies and other presentation interfaces.	1
REQ-GEN-005	The LinkedTV platform or parts thereof CAN be made publicly available as Open Source frameworks. This will be discussed and decided by the consortium.	2

2.2.3 Scalability, multi-user, multi-client, parallelism, load balancing

REQ-GEN-006	The envisioned LinkedTV platform MUST be designed as a highly scalable and performant system, which generally will allow handling large amounts of videos and related web content. The LinkedTV pilot platform to be built in the project is intended to validate the developed approach in realistic small scenarios without building up exhausting investments in CPU,	1

	RAM, disk and network capacities. However, the employed architecture has to allow to scale up the pilot for extended operational use. For the pilot only a limited amount of media resources and end users will be needed. It is planned to analyse only one video at the same time and have a limited number of end users (e.g. < 10). However, the system architecture has to be designed to be scalable by using a service oriented architecture which could perspectively be deployed on many servers, execute several analysis and distribution channels in parallel and supports load balancing. A multi-client capability, which allows to run several logical separated clients with their resources on one system will not be realized in this project.	

2.2.4 Storage of data

REQ-GEN-007	The envisaged platform should allow to offer the retrieval service for videos distributed over the internet. However, the analysis process will require at least a temporary download of the whole video. For the demonstrator developed in the project, we plan to store the whole videos and meta data on the local platform. This approach ensures, that the server and network environment provides all features required for LinkedTV demonstrator (e.g. streaming server for AV Media Fragments, HbbTV delivery). We assume, that it will not cause problems to provide an appropriate environment needed for future LinkedTV platform. The analysis tools may reference other data in the internet, e.g. of the LOD cloud. The architecture design also considers the option to store metadata and videos in the cloud.	1

2.2.5 Flexible connection to existing standard AV platforms

REQ-GEN-008	It SHOULD be considered, that the resulting platform could be used to enhance existing B2B and B2C platforms providing AV materials. This will be eased by using standard formats for materials and metadata. The Interfaces to connect to the currently prevailing CMS, MASM etc. products should be conceptually considered and be provided if needed.	2

2.2.6 Maintenance

REQ-GEN-009	Since the pilot will include links to background resources in the internet (e.g. for semantic enrichment in the LOD cloud), the platform should include the necessary tools e.g. for checking validity (dereferencability) of URLs.	1

2.2.7 Language support

REQ-GEN-010	The platform should be designed to support any language of text or audio sources, which can be met by integrating open translation services such as Google Translate as well as an appropriate ontology design.	1

REQ-GEN-011	In order to meet the requirements of the usage scenarios and demonstrators, the platform MUST support the basic languages of the media resources, i.e.: • Dutch - Scenario Beeld en Geluid • French - Scenario Uni Mons • German - Scenario rbb Although no English material will be used for the demonstration purposes, English MUST be supported in order to directly allow for the analysis and retrieval of English media resources	1

2.2.8 Distributed development of components

REQ-GEN-012	This requirement related to the development process is one of the most important aspects for the design process. Since up to 10 partners develop in several WPs the platform components in parallel it is necessary, that the interfaces between the WPs and main components are defined very early within D5.1. This enables, together with a set of test data, to separately develop all components and smoothly integrate them later.	1

2.3 Requirements from Showcase Scenarios

The LinkedTV showcase scenarios are designed in such a way that a broad range of possible functionalities will be covered. These range from dedicated TV scenarios exploring possibilities existing with present and future HbbTV technology (by partner rbb, DE), over advanced Web-based “lean-forward” interactive user scenarios (by partner Beeld en Geluid, NL) to highly experimental explorations into what could be made use of in future multimedia and TV experiences (by partner University of Mons, BE). Each partner is developing three different scenarios itself, thus resulting in a total of nine showcase scenarios. For a complete description of the scenarios see *D6.1 Scenario Descriptions*.

The following table gives a short overview.

Scenario ID	Title/Characteristics
SCN-RBB-001	Ralph – Young carpenter in rural area
SCN-RBB-002	Nina - Teacher on maternal leave in the city
SCN-RBB-003	Peter - Retired widower in semi-urban terrace house
SCN-SVN-001	Rita - Explorative search behaviour, medium digital literacy
SCN-SVN-002	Bert - Focused search behaviour, high digital literacy
SCN-SVN-003	Daniel - Explorative search behaviour, high digital literacy

Scenario ID	Title/Characteristics
SCN-UMS-001	VideoEdit
SCN-UMS-002	SoftCinema
SCN-UMS-003	GamesDebate

2.3.1 Scenarios SCN-RBB-001, SCN-RBB-002, SCN-RBB-003

All three rbb scenarios develop from watching a daily news show (“rbb aktuell”), which covers mainly regional news. They differ in the way the different users (“Ralph”, “Nina”, “Peter”) react to and interact with the TV content. The scenarios start in a comfortable living room atmosphere, but address generally a more ubiquitous approach including:

- Enhanced News Show with individual options
- Watching news items in varying depth, according to individual interest
- Using closed AV media pool, but also other data sources such as Web White List

User Groups

The target groups for the rbb scenarios are

1. Types of private end users with different age, education, profession, device, place of living, technical knowledge, income and interests
2. Professional editors using the annotation tool to prepare the materials.

For a detailed description of involved personas and storyboards refer to D6.1 Scenario Descriptions.

The following table summarizes the requirements resulting from the three RBB scenarios. These requirements intentionally only denote the pure required functionality, they do not make any assumptions, constraints, or preconditions on how actually these requirements should be realized. As well, these requirements at present can be given only at a very high, abstract level. The description on how these requirements will be actually realized will be part of WP3.

All rbb use cases address end users with TV-sets

All 3 use cases from the rbb address end users with TV-sets in an HbbTV environment. In the first project phase, when HbbTV is not yet supported, a prototypical web-based implementation with interactions based on keyboard and mouse could be used to evaluate the scenario. In the final stage, the use cases should be realized within a HbbTV environment to demonstrate the full range of planned features.

Requirements for end users

REQ-SCN-101	**Log In / User Identification** User should be able to login and be identified.	1
REQ-SCN-102	**Favourites list** The User should be able to administer a list of favorite subjects. This list MUST be displayed to the user.	1
REQ-SCN-103	**Recommendations list** For the user a list of recommended items SHOULD be presented	2
REQ-SCN-104	**List of visited links** A list of visited links should be presented to the user	1
REQ-SCN-105	**Skip items** The user MUST be able to skip items within a list which he/she does not want to see	1
REQ-SCN-106	**Notification for upcoming items** The user MUST be able to receive notifications	1
REQ-SCN-107	**Request and notify items** The user should be able to request items before they are available, and be notified when they are available	1
REQ-SCN-108	**Selecting and watching related news items** The user must be able to select and watch items	1
REQ-SCN-109	**Getting longer versions of item parts** When selecting and watching a part (fragment) of an item, the user MUST be able to get a longer or complete version of it.	1
REQ-SCN-110	**Profile update for thematically relevant content** The user must be able to update his/her profile for thematically relevant content	1
REQ-SCN-111	**Recommendations of relevant content** The user should get recommendations whenever relevant content is available	1
REQ-SCN-112	**Preferences update** The user should be able to storing content preferences and mark which items have been used already	1
REQ-SCN-113	**Switch to HbbTV text** The user must be able to switch from the HbbTV application to the accompanying HbbTV Text and back	1
REQ-SCN-114	**Visiting not presented links** The user should be able to visit also linked resources which are not presented by the application directly, e.g. through search	1
REQ-SCN-115	**Social network features: Sharing links** The user should be able to see who shared the items displayed and to share links with others on social networks like facebook	1
REQ-SCN-116	**Sharing links not through facebook** The user should be able to sharing links also through other means than facebook	1
REQ-SCN-117	**Connecting with friends** The user should be able to detect other users (“friends”) and enable them to join his/her group and adjust preferences to group	1

REQ-SCN-118	**Second device** The user should be able to see related and extra information on a second device (Smartphone/Tablet)	2
REQ-SCN-119	**Pause items, show extra content on TV** The user should be able to pausing items while watching and show extra content on TV screen	1
REQ-SCN-120	**Pause items, show extra content on TV** The user should be able to pausing items while watching and show extra content on second device	1
REQ-SCN-121	**Control TV application through second device** The user should be able use tablet/smartphone to manipulate or do something with the extra content	2
REQ-SCN-122	**Retrieval** The application should support several types of retrieval: text based search, semantic search, facetted browsing	1

Requirements on an annotation tool for professional editors[3]

REQ-SCN-150	**Show Video Source** An annotation tool should be provided for the editors for manual editing, annotating and linking of materials to be broadcasted. This requires functionality to select and stepwise show the video source with timeline, audio track, video track, text track, metadata, detected video fragments, associated named entities and URLs.	1
REQ-SCN-151	**Show item related information** Offer the editor items to be linked with the following attributes to assess the relevance to the current program: the title or subject, publisher, type of media, legal constraints, creation date, availability, Is it time-critical, requires it pausing or switching away.	1
REQ-SCN-152	**Pre-Filtering** Show the editor the available items filtered by the aforementioned characteristics to support the selection process.	1
REQ-SCN-153	**Editing** Allow to manually add, delete or update annotations and links to • timeline based fragments and • spatial fragments of video track items	1
REQ-SCN-154	**Recommendations** The editor should be supported by recommendations from the tool for content related to a certain video, media fragment or other type of object. The editor could use the recommendation to insert at this point a hyperlink or information offered to the end user when viewing this object.	2

[3] more detailed requirements on the annotation tool are described in the document: “Draft description of broadcaster Hyperlinking Tool”, Version 01, Björn Stockleben, Nico de Abreu, Jenni Müller, March 2012.

2.3.2 Scenarios SCN-SV-001, SCN-SV-002, SCN-SV-003

The scenarios from Beeld en Geluid are developed around the Dutch TV show “Tussen Kunst & Kitsch” as interactive TV scenarios with a cultural heritage background. They are grouped around three different scenarios employing users with different grades of digital literacy and information/interaction needs (“Rita”, “Bert”, “Daniel”).

Scenario SCN-SV-001: Rita, assistant at Art History department (young, medium media literacy)

Rita spends her free time on museum visits and reading about art. One of her favourite programmes is Tussen Kunst & Kitsch, which she likes to watch because she learns more about art history and she thinks it’s fun to guess how much the objects people bring in are worth. She’s also interested in the locations where the programme is recorded, because this usually takes place in a historically interesting location.

Scenario SCN-SV-002: Bert, Tussen Kunst & Kitsch lover (older, high media literacy)

Bert has an antiques shop in Leiden and likes to gather information quickly with a specific topic and goal in mind. Since Tussen Kunst & Kitsch has a well-organised archive of clips that has been split into various themes, offers a lot of background knowledge on the artworks and many other services that he needs for his work, Bert often uses it for research and business alike.

Scenario SCN-SV-003: Daniel: Bargain hunter (young, high media literacy)

Daniel is taking photos from artwork he is interested in and uses Tussen to upload materials and image recognition of the objects he’s photographed to get more background information. He also likes sharing his finds with others in social networks. He keeps a meticulous record of his (near) purchases and has divided the pictures in various categories, periods and locations.

Requirements from scenarios

As with all showcase scenarios the requirements resulting from the S+V scenarios at the present stage can be given at a very high, abstract and preliminary level, as they will be further explored and refined within WP3. At present, these requirements represent more points of interests to be covered than actual hard requirements. Nevertheless they indicate which kind of user interface functionality the LinkedTV platform should be able to support.

For the descriptions of the scenarios refer to *D6.1 Scenario Descriptions*.

Requirements for end users

REQ-SVN-201	**Object Identification** It should be possible to identify an entity (object, person) in a frame or other entities to link to (e.g. topics, logos, brands).	1
REQ-SVN-202	**Personalisation** e.g. the user should be able to save linked information to access at a later moment, or being offered information based on his or her preferences / situation).	1

REQ-SVN-203	**Identification of data sources** It should be possible to identify possible data sources to link to / incorporate (e.g. Wikipedia, Europeana, Flickr, YouTube, fansites, maps, Twitter, thesauri like ULAN#).	1
REQ-SVN-204	**Incorporation of media events** It should be possible to incorporate media events like quizzes (for example: before the price is mentioned, the user can guess the price of an object being discussed. When the user skips the video to watch the answer - before answering -, the user's answer does not count)	1
REQ-SVN-205	**Thematic, non-linear viewing** Viewing and according to a theme-oriented, non-linear way should be supported	1
REQ-SVN-206	**Selection of content items** Content items must be selectable	1
REQ-SVN-207	**Online purchase of objects** It should be made possible to purchase displayed objects directly online	2
REQ-SVN-208	**Real-time chat with others** Real-time chat with other users/viewers should be supported	1
REQ-SVN-209	**Background information** More background information on the objects and their value should be made available	1
REQ-SVN-210	**Information on host** information on the host of the program should be made available	1

2.3.3 Scenarios SCN-UMS-001, SCN-UM-002, SCN-UMS-003

The three scenarios contributed by University of Mons include one scenario for professional users (*VideoEdit*) and two scenarios for general public users (*SoftCinema*, *GamesDebate*). In general, they are designed in a more experimental and explorative way than the other scenarios. The exact features and storyboards of the scenarios are still in discussion.

Scenario SCN-UMS-001 VideoEdit

The VideoEdit scenario employs three different types of users: artists ("Vicki"), media professionals ("Laura") and home working users ("Nicola & Alex"). While different in their actual information and interaction needs, their common overall use case is the production of videos out of fragments, which have to be generated, selected, put in a sequential order and published.

Requirements

REQ-SCN-301	**Audio/Video Navigation and annotation** audio/video navigation and annotation that facilitate the task of selecting and fragmenting movies	1
REQ-SCN-302	**Semi-automatic annotation** Content-based and semantic-based semi-automatic annotation	1
REQ-SCN-303	**Concept map** User-friendly software-based concept map and cue list interactive visualization	1

REQ-SCN-304	**Cross-media organization** Cross-media organization help composing sample-based soundtracks that would fit to the movie collage	1
REQ-SCN-305	**Editing workflow support** Support all the tasks of the workflow by computer-based navigation, selection, annotation, segmentation, recomposition to be performed in realtime, in a performance context, and also offline	1
REQ-SCN-306	**Predefinition and Realtime Selection of Storyboards** Provide storyboards be pre-defined offline during preparation and selected in realtime during a performance	2
REQ-SCN-307	**List of Subject Matters** Generating of a list of subject matters	1
REQ-SCN-308	**Definition of master subjects** Definition of master subjects (such as: mind control, moving objects/furniture etc.)	1
REQ-SCN-309	**Sketch Generation** Generation of "sketches" and order of footage into rough categories	1
REQ-SCN-310	**Film footage edits and mixdown** Timeline of film footage edits and final mixdown	1
REQ-SCN-311	**Drag and Drop** Drags and drops rushes on the timeline of favorite video edit	1
REQ-SCN-312	**Realtime annotation** Quickly annotation in real time while recording, the passages later needed for montage	2

Scenario SCN-UMS-002 SoftCinema

The SoftCinema scenario employs home users watching films or episodes on TV. The idea is that the user's behaviour or emotions to the content while watching is recognized which leads to changes of what will be shown next by recomposing the episodes based on preproduced variants.

Requirements

Preliminary requirements:

REQ-SCN-331	**User behavior detection** User behavior, activity and emotions should be detectable	1
REQ-SCN-332	**Change Rules** Rules for changing video scenes based on user reaction or preferences should be editable	1
REQ-SCN-333	**Automatic Recomposing of video scenes** Video scenes should be automatically be recomposed based on the rules defined	1
REQ-SCN-334	**Multimedia Content Analysis** the user activity, when recognized through computer vision methods, might use some of the algorithms used for the multimedia content analysis (for instance face recognition of users and characters in the videos)	1

Scenario SCN-UMS-003 GamesDebate

The GamesDebate scenario is based on collaborative cooperation of multi non expert users. It extends current usages of televoting and feedback by exploring advanced tactile interfaces such as Microsoft Kinect and direct interactive display of feedback on interactive maps and the like.

The use cases apply to TV games and debates.

Requirements

REQ-SCN-361	**Real-Time Applause Meter** A TV game: real time "applause meter" or more generally "mood meter" to know the viewers fillings valence or arousal depending on the game events	1
REQ-SCN-362	**Collective annotations** A TV game: can be directly used to collective annotations	1
REQ-SCN-363	**Real-Time Map** A debate: a real-time map of the country with the positive/negative reactions to political comments can be shown	1
REQ-SCN-364	**Synchronicity detection** A fitness / gymnastic show can use the behavioral tracking to check the viewer synchrony on the animator's movements	1
REQ-SCN-365	**Mood recognition** Recognition of user’s mood and behavior by behavioral tracking or tactile interfaces	1

2.4 Requirements from WP1 – 4

The following section covers the requirements resulting from the research work packages WP1 to WP4. The keywords MUST, MUST NOT, SHOULD and SHOULD NOT are to be interpreted as defined in RFC 2119.[4]

2.4.1 Requirements concerning video analysis and fragmentation (WP1)

The requirements for WP1 comprise the analysis of different video tracks, type of material, formats, meta data and the analysis process.

REQ-ANL-001	**Original Video Resources** For the initial analysis the platform must provide sufficient access to video material, either directly file-based or via reference to URIs.	1

[4] http://www.apps.ietf.org/rfc/rfc2119.html

REQ-ANL-002	**Availability of related information to video resources** Information related to the video material should be provided, such as general metadata, audio tracks, audio speech recognition transcripts, subtitles, translated subtitles, etc. These can be either provided beforehand, or generated by tools interacting with the platform, like Speech Recognition/Natural Language Processing tools, etc.	1
REQ-ANL-003	**Different variants of video material** The video sources and tracks can be provided in different variants, such as different codecs, formats, resolutions, different language audio tracks etc.	2
REQ-ANL-004	**Standardized addressing scheme** The different tracks etc. should be made available through a standardized addressing scheme	1
REQ-ANL-005	**Parallel access to video material** Access to video material and related tracks or metadata by different tools for analysis purposes should be generally possible. Initially, an optimistic approach concerning access is considered sufficient. In case a particular tool or service will require exclusive access, an appropriate locking mechanism has to be provided.	2
REQ-ANL-101	**Event-based Processing** In order to support sequential analysis of videos and an overall workflow, at least begin and end of each analysis should be notified to the platform as an event. While this is not required in the beginning for the analysis of the videos for the demonstration scenarios, a general workflow and event based processing has to be supported by the final version of the platform.	1
REQ-ANL-102	**Standardized metadata format** A standardized metadata format should be used or developed such that all tools or services (aka agents) which add or update metadata information do so in a compatible way. This metadata information should always contain the information on which agent provided this piece of information. This standardized metadata format must be based on existing or emerging standards like the W3C Media Fragment Specification[5] and the W3C Ontology and API for Media Resources[6]	1

In the following requirements concerning fragmentation are described. As the LinkedTV platform will comply to the W3C Media Fragment Specification relevant requirements concerning generation of fragments are listed here as well; in parenthesis the reference number from the document on “Use cases and requirements for Media Fragments” [7] within is given.

REQ-FRG-001	**Unique Resource** Media fragments are a representation of the parent resource and should not create a new resource, in particular not a new resource of a different Internet media type (or MIME type). (3.3)	1

[5] http://www.w3.org/TR/2011/CR-media-frags-20111201/

[6] http://www.w3.org/2008/WebVideo/Annotations/

[7] http://www.w3.org/2008/WebVideo/Fragments/WD-media-fragments-reqs/

REQ-FRG-002	**Valid Resource** Resources delivered as a response to a media fragment URI request should be valid media resources by themselves and thus be playable by existing media players / image viewers. (3.4)	1
REQ-FRG-003	**Parent Resource** The entire resource should be accessible as the "context" of a fragment via a simple change of the URI. The media fragment URI - as a selective view of the resource - provides a mechanism to focus on a fragment whilst hinting at the wider media context in which the fragment is included. (3.5)	1
REQ-FRG-004	**Single Fragment** A media fragment URI should create only a single "mask" onto a media resource and not a collection of potentially overlapping fragments. (3.6)	1
REQ-FRG-005	**No transcoding** Media fragments should preferably be delivered as byte-range subparts of the media resource such as to make the fragments an actual subresource of the media resource. The advantage of this is that such fragments are cacheable as byte ranges in existing caching Web proxies. This implies that it should be avoided to decode and recompress a media resource to create a fragment. (3.8)	1
REQ-FRG-106	**Temporal fragments** The storage of temporal fragments must be supported. Within the context of WP1 this means that start and end times of fragments within the duration of the parent resource have to be stored in such a way that the can be identified in order to make further analysis and annotation by other agents possible.	1
REQ-FRG-107	**Spatial fragments** The storage of spatial fragments must be supported. Within the context of WP1 this means that rectangular regions of fragments of the parent resource (Regions of Interest, ROI) have to be stored in such a way that the can be identified in order to make further analysis and annotation by other agents possible.	1
REQ-FRG-108	**Track fragments** The storage of track fragments must be supported. Whereas the extractability of tracks by a user agent at delivery time depends on the container format of the media resource, within the context of WP1 and WP2 this means that multiple tracks of media resources should be able to be stored, annotated, retrieved and generated separately.	1
REQ-FRG-109	**Named fragments** The annotation of names to fragments should be supported.	1

2.4.2 Requirements concerning annotation (WP1, WP2)

In the following we list the relevant requirements concerning annotation as has been identified by the W3C Media Annotations Workgroup[8], although some requirements are not agreed upon to be implemented by the Work Group itself among all members.

The LinkedTV platform has to provide an Ontology and API for Media Resources which complies with these requirements:

[8] http://www.w3.org/TR/2010/WD-media-annot-reqs-20100121/

REQ-ANO-001	**Support of Ontology for Media Resource 1.0** For the annotation of media fragments the Ontology for Media Resource 1.0[9] (which is currently a W3C recommendation) should be supported. By means of the mapping table which is described there, other formats can be supported as well, if necessary.	1
REQ-ANO-002	**Methods for getting metadata information stored in different formats** The API MUST provide methods for getting metadata stored in different formats related to media resources. Metadata can be in one of different formats, either as separate document or embedded in media resources.	1
REQ-ANO-003	**Methods for setting metadata information stored in different formats** The API MUST provide methods for setting metadata stored in different formats related to media resources. Metadata can be in one of different formats, either as separate document or embedded in media resources.	1
REQ-ANO-004	**Support of Structured Annotations** The API MUST provide a means to support structured metadata to media resources, like the name of the creator being structured in “first name” and “last name”.	1
REQ-ANO-005	**Support of user-defined Metadata** It MUST be possible to access user-defined metadata to media resources. "user-defined metadata" means metadata that is not defined in a standardized format, but which is being created entirely by the user. While this requirement might not be necessary at first hand within WP1 or WP2, this is definitely required within WPs 3 and 4.	1
REQ-ANO-006	**Providing the ontology as a simple set of properties** The ontology MUST be available as a simple set of properties, to hide complexity for those who do not need it. This requirement is mainly important for the development of user interfaces as part of WP3 and WP4.	1
REQ-ANO-007	**Internal or external format for the ontology** The ontology MUST be provided not only in prose description but also as a machine processable format serialized in OWL/RDFS/SKOS.	1
REQ-ANO-007	**Several abstraction levels in the ontology** The ontology MAY provide several abstraction levels, since several metadata standards like FRBR[10] or CIDOC[11] allow referring to multimedia resources on several abstraction levels	1
REQ-ANO-008	**Access collections of metadata** It MUST be possible to access collections of metadata.	1
REQ-ANO-009	**Provenance information of metadata properties** The ontology MUST support provenance information of metadata properties.	1
REQ-ANO-010	**Description of fragments of media resources** It MUST be possible to relate metadata to fragments of media resources.	1

[9] http://www.w3.org/TR/2012/REC-mediaont-10-20120209/

[10] http://archive.ifla.org/VII/s13/frbr/frbr.htm

[11] http://cidoc.ics.forth.gr/docs/cidoc_crm_version_5.0_Dec08.pdf

REQ-ANO-011	**Description of fragments of media resources** It MUST be possible to relate metadata to fragments of media resources.	1
REQ-ANO-012	**Controlled vocabularies for the values of different properties** It MUST be possible to take information from controlled vocabularies for certain properties into account.	1
REQ-ANO-013	**Different return types for the same property** It MUST be possible to provide different return types for the same property.	1
REQ-ANO-014	**Policy information** The ontology SHOULD provide support for linking policy information related to the media resource.	1
REQ-ANO-015	**Discovery of named and track fragments** The ontology MUST provide properties to query the list tracks that exist in a media resource as well as the list of named fragments.	1
REQ-ANO-016	**Annotation tool to adjust automatic annotations – Show video** An annotation tool should be provided for the editors to adjust annotations of materials previously set by the analysis process. This requires functionality to select and stepwise show the video source with timeline, audio track, video track, text track, metadata, detected video fragments, associated named entities and URLs.	1
REQ-ANO-017	**Annotation tool to adjust automatic annotations - editing** Allow to manually add, delete or update annotations and links to • timeline based fragments and • spatial fragments of video track items	1

2.4.3 Requirements concerning user interfaces (WP3)

In the following table the preliminary requirements with respect to the LinkedTV platform concerning user interfaces are listed.

REQ-UIM-001	**API for Metadata Access** The platform MUST provide an API which enables applications to retrieve Metadata	1
REQ-UIM-002	**API for Video Access** The platform MUST provide an API which enables applications to get media resources and fragments thereof via URIs	1
REQ-UIM-003	**Generic Presentation Framework** The platform MUST provide a generic presentation framework which covers the basic functionalities, such as Search, Browse, Navigate, Get related items, get media resource, , fragment, add user generated metadata etc.	1
REQ-WEB-001	**HTML5 Presentation Framework** The platform MUST provide an HTML5 presentation framework based on the Generic Presentation Framework	1
REQ-HBB-001	**HbbTV Presentation Framework** The platform should provide an HbbTV 1.1 compatible presentation framework based on the Generic Presentation Framework.	2
REQ-WEB-002	**HTML5 Media Player** The platform MUST provide an HTML5 based LinkedTV Media Player	1

REQ-HBB-002	**HbbTV Media Player** The platform should provide an HbbTV 1.1. compatible LinkedTV Media Player.	2
REQ-UIM-004	**Mobile Media Player** The platform MAY provide specific LinkedTV Mobile Media Players, such as Media Players based on Android or iOS.	2

2.4.4 Requirements concerning personalisation and contextualisation (WP4)

REQ-PRS-001	**API for User Profiles** The platform MUST provide an API for the storage, retrieval and update of user profiles.	1
REQ-PRS-002	**Ontology for User Profiles** The platform MUST provide an ontology for user profiles. The user profile ontology MUST be integrated with the other ontologies provided by the platform. Initially, the user profile ontology is read-only with respect to users. It might be necessary to provide a means for users to change the ontology or to add user-specific ontology data.	1
REQ-PRS-003	**Role Profiles** The platform MUST support the administration of general role profiles like “professional”, “user with low digital literacy”, These role profiles MUST be able to be combined and to be evaluated by means of rules.	1
REQ-PES-004	**Individual Profiles** The platform MUST support the administration of individual user profiles, where user preferences can be attached to individual users either explicitly or implicitly, or both.	1
REQ-PRS-005	**Explicit user preferences** The API for user profiles MUST support the storage, update and retrieval of explicit user preferences. Explicit individual user profiles also require that an own user interface has to be provided through which the user can administer her/his user profile.	1
REQ-PRS-006	**Implicit user preferences** The API for user profiles MUST support the storage, update and retrieval of implicit user preferences. This requires that automatically gathered data can be attached to user profiles.	1
REQ-PRS-007	**API for Context Information** The platform MUST support an API for storage, update and retrieval of contextual information.	1
REQ-PRS-008	**Ontology for Context Information** The platform MUST support an ontology for context information.	1
REQ-PRS-009	**API for User Behavior Information** The platform must support an API for storage, update and retrieval of user behavior information which is generated through user behavior tracking.	1

2.5 Conclusions and Summary for Requirements

The platform architecture should cover all requirements with the priority 1 of WP1-4 and Scenario 1 and 2 in order to provide a basic platform which meets the vision of the project.

The platform intends to offer with the Hypervideo Player a general front end with basic functions for a personalized searching and viewing of videos, which will probably cover all needs to show Scenarios 1 and 2. For Scenario 1 from rbb it has to be considered, that it should finally address end users with a TV set and should therefore support the remote control. The remote control could e.g. be provided by a remote control of a TV-card for the PC but at least simulated by the keyboard.

For the Scenario 3 of UMons concerning VideoEdit only basic functions such as retrieval, recommendations and clustering of search results can be offered to support the user to generate the collages. A complete support of the whole process would require the development of a specific application, which offers user interface functions to compose collage elements and include overviews, timeline and final mixdown.

The other 2 use cases of scenario 3: SoftCinema and Games-Debates also require the development of specific applications for Interaction features for user feedback, user input by gesture recognition via camera, mood recognition by camera. The architecture will offer interfaces to implement specific applications for Scenario 3 by using WP3-4 interfaces.

3 System Overview

The overall LinkedTV environment is depicted in Figure 1. The main LinkedTV process takes external media resources (videos), analyses and annotates them within the server platform by generating semantically rich pieces of the videos (media fragments), enriches them semantically further with external information from the Web and Linked Open Data sources (“LOD Cloud”). The annotated tags and links can be adjusted and enhanced by an editor through tools for annotation and hyperlinking.

Based on the enriched and interlinked media fragments various user applications with personalized user interfaces for clickable video allow the user to access the provided videos. Within LinkedTV three different user scenarios demonstrate the possibilities enabled by the LinkedTV approach. This includes a Web client variant using a Browser with the full potential of HTML5 for clickable video for end users with a two-way HTTP communication. The second client variant, using HbbTV 1.1., HTTP and a TV-set with restricted functionality, will be addressed after a feasibility analysis has been undertaken.

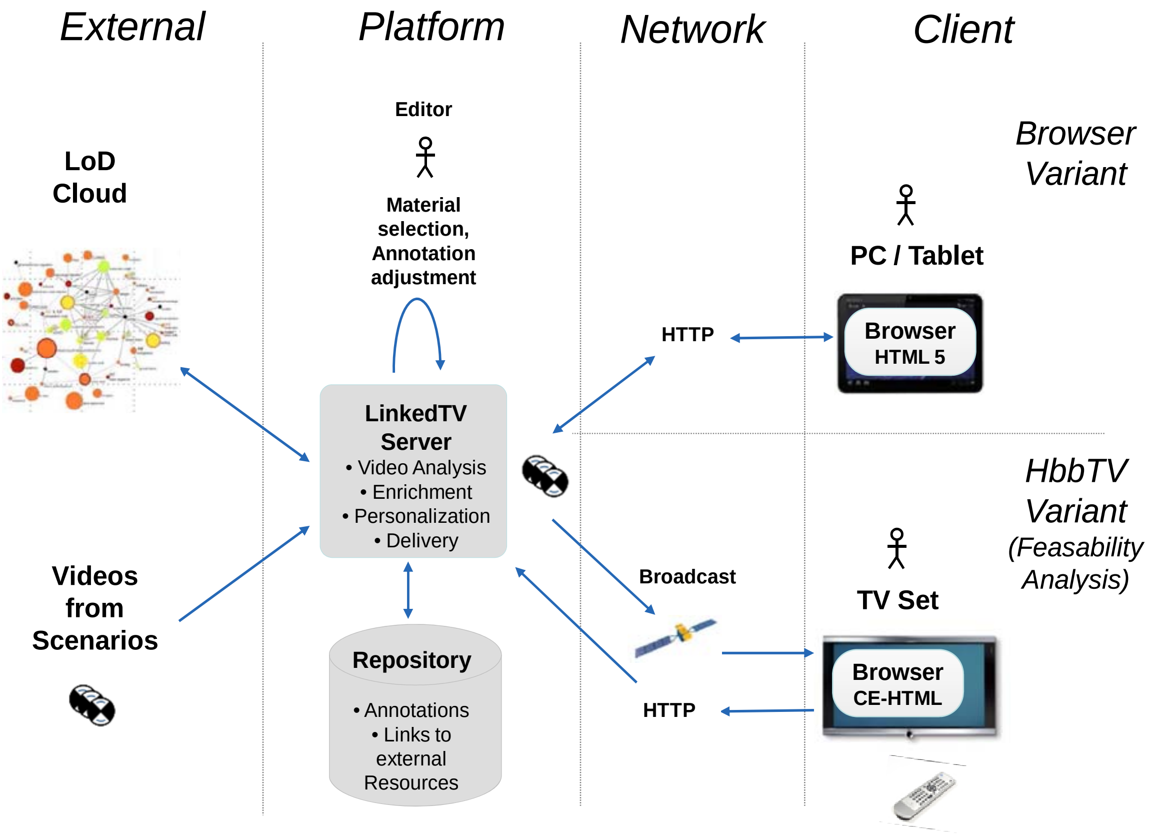

Figure 1: LinkedTV System Overview

4 The LinkedTV Workflow

Within the vision of the overall LinkedTV workflow two main parts can be distinguished: a) the Analysis and Annotation Workflow and b) the Linked Media Presentation Workflow. The purpose of the first workflow is to provide and connect all information which is contained in what is called the Linked Media Layer through subsequent steps of segmentation, metadata aggregation and embedding into the whole Web. The purpose of the second workflow is to present this Linked Media information in interactive and personalised ways to the user.

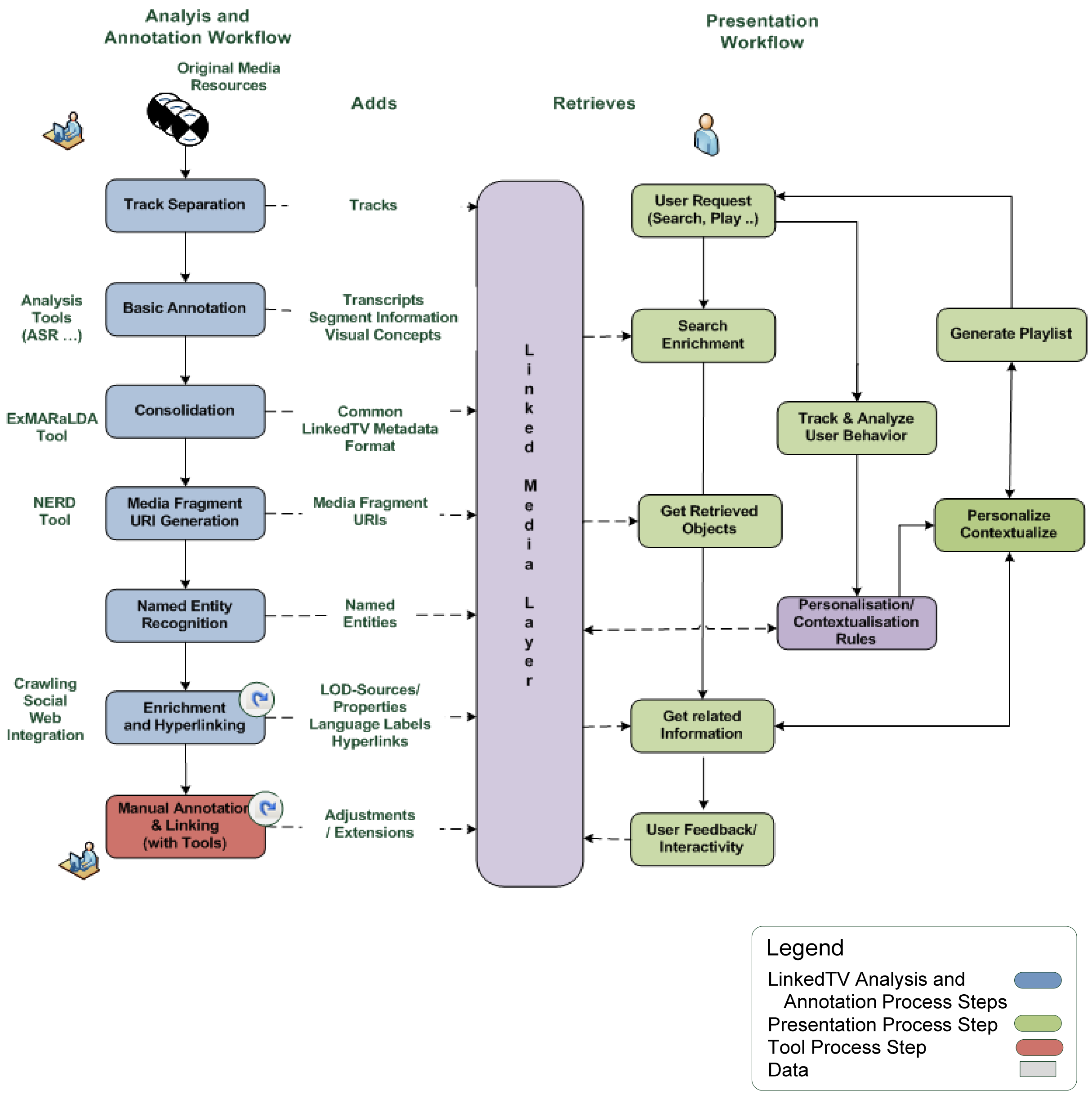
Analyis and Annotation Workflow
Presentation Workflow
Original Media Resources
Adds
Retrieves
Track Separation
Tracks
Analysis Tools (ASR ...)
Basic Annotation
Transcripts Segment Information Visual Concepts
ExMARaLDA Tool
Consolidation
Common LinkedTV Metadata Format
NERD Tool
Media Fragment URI Generation
Media Fragment URIs
Named Entity Recognition
Named Entities
Crawling Social Web Integration
Enrichment and Hyperlinking
LOD-Sources/ Properties Language Labels Hyperlinks
Manual Annotation & Linking (with Tools)
Adjustments / Extensions
Linked Media Layer
User Request (Search, Play ..)
Search Enrichment
Generate Playlist
Track & Analyze User Behavior
Get Retrieved Objects
Personalize Contextualize
Personalisation/ Contextualisation Rules
Get related Information
User Feedback/ Interactivity
Legend
LinkedTV Analysis and Annotation Process Steps
Presentation Process Step
Tool Process Step
Data

Figure 2: General LinkedTV Workflow

The Annotation and Analysis Workflow is mainly a linear sequence of steps including a lot of sub processes. These steps may include manual edition as well (depicted with the symbol ✍), and also tasks which have to be processed not only once, but in a continuous catch-up processing with changing content in the Internet (depicted with ⟳).

The Presentation Workflow, on the other hand, is mainly a circular process, which involves requests issued by a user, generating enriched and related content, adapting it to personal interests and viewing context and then presenting it again to the user, who might trigger a new process. Of course, a lot of different workflows are possible here, including non-personalised and linear broadcast only workflows. These different variants are discussed in detail in *D3.2 Specification of functionality requirements satisfying user information needs*.

Both main workflows are discussed in more detail in the following as far as the interaction with the LinkedTV Platform is concerned.

4.1.1 The Analysis and Annotation Workflow

Attached to the **original video file** there is **general metadata** which refers to the video as a whole, such as media resource data like *format, duration, file name, URI, resolution, language*, etc. as well as content related information like *content summary, director, genre, title*, or e.g. with broadcast related material data like *air date, repetition, channel*, etc.

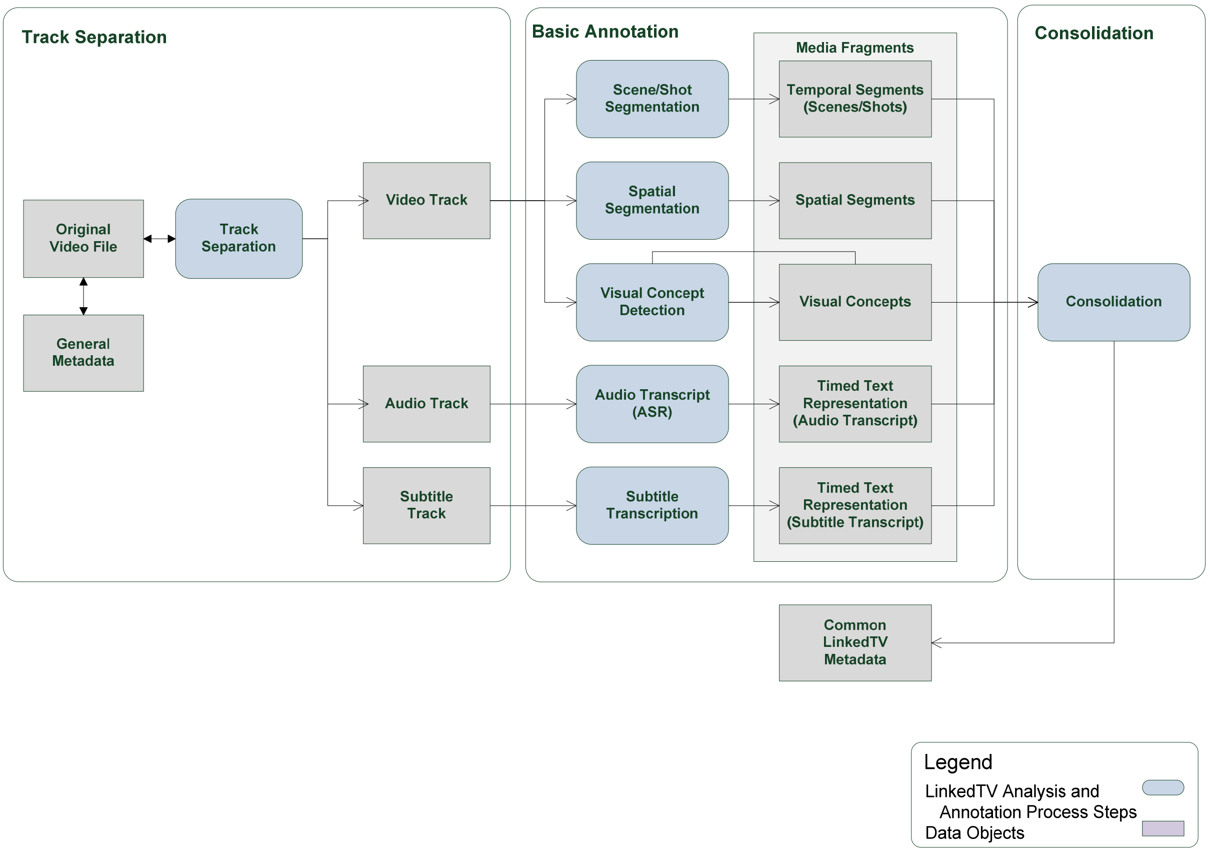

Figure 3: Workflow of Intelligent Hypervideo Analysis and Annotation

Each video normally contains at least two different tracks, the **video track** and the **audio track**. Additional tracks might be available or extractable, like one or several subtitle tracks, audio tracks for other languages or even an additional video track translating the audio voice into sign language for deaf people. For each type of track the LinkedTV platform contains specialized track analyzer modules. Through a **track separation** step the appropriate tracks are made available to the respective analyzers. The different analyzers can operate in parallel on the original track source. Each analyzing process results in a specific segmentation of the original video source. When all analysis processes are completed, the **consolidation** phase has to merge the different results into a single format, the **Common LinkedTV Metadata Format**. This consolidation does not only harmonize the different resulting formats into one common format, but also ensures that the different types of segments are synchronized and linked to each other, and also that the metadata information contained complies to the Media Fragment URI 1.0 standard.[12]

The metadata will be consolidated in an MPEG7 compliant XML file based on the ASR segmentation. This XML file already contains an `<audio_segment>` element, the additionally required types of segments will be integrated. The format of this file is also compliant to EXMARaLDa, the Extensible Markup Language for Discourse Annotation,[13] which allows editing these files with the already existing EXMARaLDa Annotation Tool.

4.1.2 The Media Fragment URI Generation and Enrichment Workflow

After the analysis of the video files is completed the main workflow is executed which prepares the media fragments for later presentation, intelligent retrieval and linking to other resources:

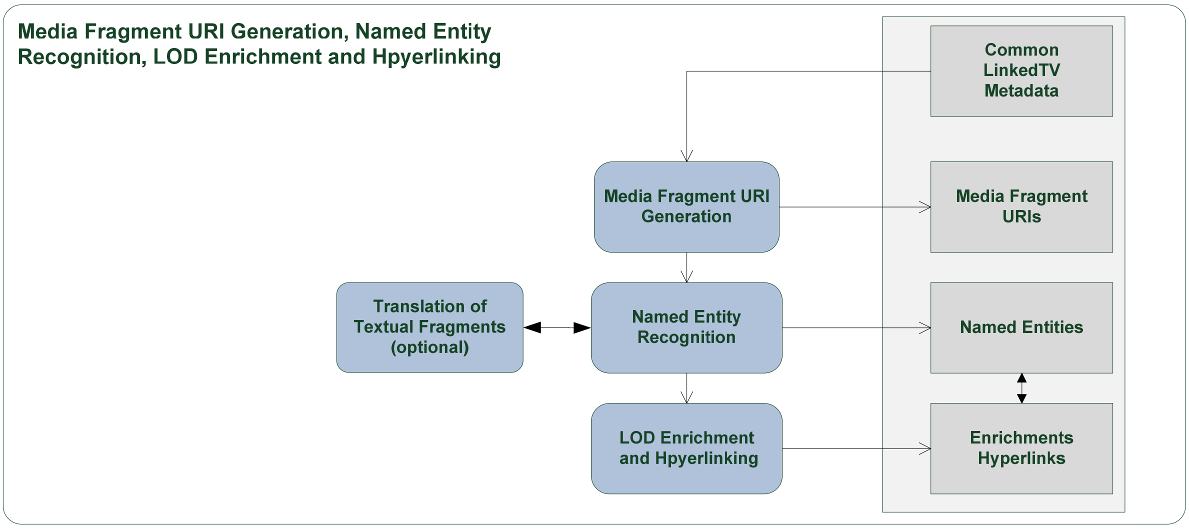
Media Fragment URI Generation, Named Entity Recognition, LOD Enrichment and Hpyerlinking
Common LinkedTV Metadata
Media Fragment URI Generation
Media Fragment URIs
Translation of Textual Fragments (optional)
Named Entity Recognition
Named Entities
LOD Enrichment and Hpyerlinking
Enrichments Hyperlinks

Figure 4: Workflow Media Fragment URI Generation and Enrichment

[12] http://www.w3.org/TR/media-frags/

[13] http://www.exmaralda.org/

4.1.2.1 Media Fragment URI Generation

After the consolidation of the segmentation results for each detected fragment a Media Fragment URI will be generated, which will then be stored in the Metadata Repository. All further processing will be performed on the basis of these Media Fragment URIs.

4.1.2.2 Named Entity Recognition, Enrichment and Hyperlinking

The subsequent processing steps are then attaching semantics to the Media Fragments. This consists basically of the following steps: 1) **Named Entity Recognition**, 2) **Enrichment** and 3) **Hyperlinking**.

Named Entity Recognition

Named Entity Recognition (NER) is first applied to the textual information, i.e. mainly subtitle track, audio transcript and visual concepts. This happens analyzing the textual fragments by means of the Meta-NER service NERD by partner EURECOM.[14]

Enrichment

After the basic named entity recognition, a continuous enrichment process adds further information to the recognized entities by linking them to the various LOD sources detected by the NERD process. This basically generates a big RDF graph which is continuously growing by adding constantly new facts and relationships.

Hyperlinking

The final step is the – also continuous – process of hyperlinking. Hyperlinking adds links to other web resources to the media fragments. These can be normal content sites like news sites, specific social web services like Google Maps/Bing Maps, facebook, twitter, IMDB, Ebay, Amazon, or also links to other Media Fragments themselves. The process of hyperlinking can be both performed automatically through web crawling and connections via APIs, as well as through the manual management of hyperlinks through a controlled editorial or a shared social process.

4.1.3 The Presentation Workflow

In the following section an overview of the general use case workflows covered by the Presentation Layer is given. For an in-depth discussion cf. Deliverables *D3.1 Specification of functionality requirements* and D3*.2 Specification of presentation interfaces*.

The functionalities will be based on general user goals, from which so called interface elements will then be derived. The following table lists general user goals and variants thereof, among which the relevant use cases for LinkedTV will be selected.

[14] http://nerd.eurecom.fr/

Table 3: User Goals

User Goal	Variants
Entertainment	*Lean-back entertainment*: No interaction needed from the user, s/he can perform other activities *Active entertainment*: Users can interact with the TV for entertainment (e.g. Users can participate in a game) *Participation*: Users can influence the content (e.g. they can influence what will happen in a movie/series)
Information	*Getting information*: Users can get additional information on a certain topic (so enable them to ask for and present the information). *Adding information*: Users can add information that they think would be useful at some later point, for themselves or for others *Education*: Users can actively learn more about certain items.
Social	*Sharing information*: Users can share information with others, e.g., by sending a link via email, chatting, social network,.. *Discussing information*: Users can discuss the content in real time with others. *Social context*: Pay attention to the social context, links depend on the group that is currently watching
Manipulation	*Browsing video*: Users can skip items, revisit items, look at preview,... *Editing video*: Users can fragment the video, make summaries, add comments, bookmark *Structuring content*: Users can structure the video (related to editing) or they can manipulate a given structure *Zooming in/out:* Users can select at what level of detail they want to receive information ('magnification tool') Comparing content: Users can compare information of two different sources
Personalization	*Personalize interactive layout*: Users can turn off certain interactive features or arrange them in a certain way. *Personalize link suggestions*: Users can make clear what kind of suggestions/extra information they would like to receive (e.g. "never show me shopping suggestions", "show me items related to the environment") *Different modes*: It is possible to work with different modes (e.g. Different mode when viewing movie, than when viewing the news). This depends on person, situation, location, time,...

From these user goals general use case workflows are derived which define the interaction with the platform. Figure 5 displays the LinkedTV workflows within the Presentation Layer:

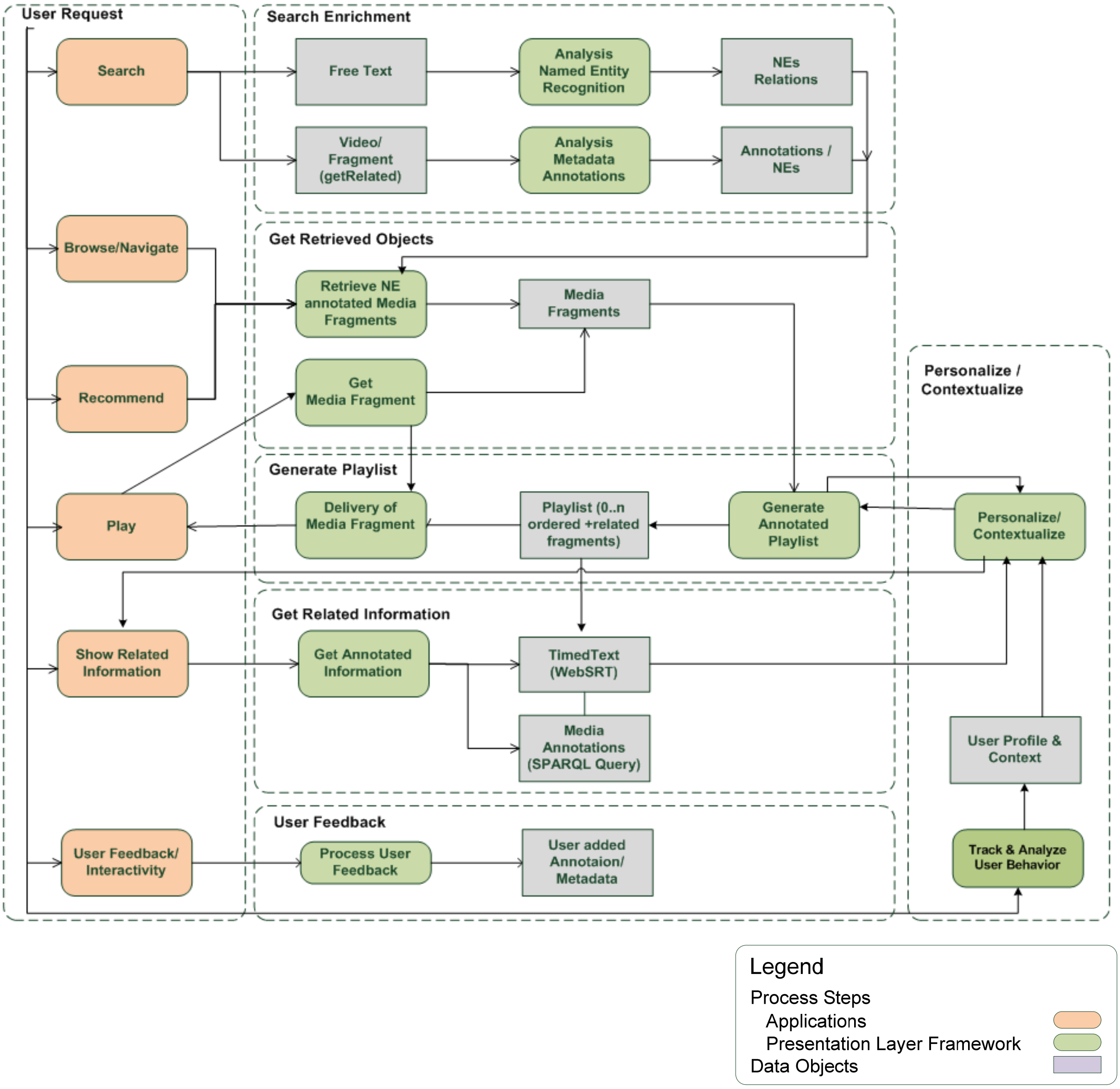
User Request
Search Enrichment
Search
Free Text
Analysis Named Entity Recognition
NEs Relations
Video/ Fragment (getRelated)
Analysis Metadata Annotations
Annotations / NEs
Browse/Navigate
Get Retrieved Objects
Retrieve NE annotated Media Fragments
Media Fragments
Recommend
Get Media Fragment
Personalize / Contextualize
Generate Playlist
Play
Delivery of Media Fragment
Playlist (0..n ordered +related fragments)
Generate Annotated Playlist
Personalize/ Contextualize
Show Related Information
Get Related Information
Get Annotated Information
TimedText (WebSRT)
Media Annotations (SPARQL Query)
User Profile & Context
User Feedback
User Feedback/ Interactivity
Process User Feedback
User added Annotaion/ Metadata
Track & Analyze User Behavior
Legend
Process Steps
Applications
Presentation Layer Framework
Data Objects

Figure 5: Workflow of the Presentation Layer

The workflow of the end user comprises functions for search, browsing, showing related information and feedback or interactivity. This list is most likely not complete with respect to the final use case definitions. It shows, however, the general interaction flows, and can be easily extended. Please note as well, that the following use cases are described on a very abstract level and will be refined in subsequent steps.

Search

For the search use case we assume that both textual retrieval and search based on video content (e.g. for “similar as” queries) can be issued. Both have first to be analysed: textual search queries will be analysed through Named Entity Recognition, search based on video content will be analyzed by taking the annotated metadata information into account.

Optionally, the video content itself could be analyzed by applying the video analysis tools as developed by Work Package 1.

The query analysis results in a collection of named entities and relations between named entities which will then be taken as basis for getting search results in terms of media fragments and annotations.

The retrieval of media fragments can optionally be filtered and adapted through a personalisation and contextualisation process. The result is an internal collection of media fragments which will then be brought into specific order by generating an annotated playlist. This playlist contains the references to the actual media fragments which are delivered to the user by means of a video streaming server and then played by the user.

Browsing/Navigation/Recommendation

Browsing/Navigation basically requires to retrieve media resources and related information based on metadata information attached to the content which is displayed to the user. Browsing and navigation applies to the navigation within an already generated playlist (like “show next item”), but also to implicit queries based on relations and annotations, like “show similar media resources with respect to location, objects, language” or any other property or relation.

In both cases the already given semantic annotations can be exploited and then be used for the retrieval of appropriate media fragments. The subsequent process is roughly the same as for the search use case, including personalisation and contextualisation.

This holds as well with the recommendation use case. Recommendation means basically also the generation of playlists containing recommended items based on the annotations, by optionally applying user preferences and further methods like social recommendations or collaborative filtering

Play

The Play use case covers the actual playing of media resources. This actually covers the basic functionality of the Media Player to be developed within LinkedTV. This particularly implies that the handling of media fragments and annotations can be dealt with, i.e. that fragments of complete resources are handled correctly and annotations (see below, Show Related Information) can be displayed.

The Play use case also includes all normal player features such as pause, stop, fast forward, hide/show annotations, mute on/off, etc.

Show Related Information

The Show Related Information Use Case covers all use cases which deal with the retrieval and display of information annotated to media resources. Possible sub use cases include the display of clickable textual or visual annotations directly within regions of the video, as well as links displayed in a side bar and activated by means of a TV remote control, as well as displaying additional information on a second device.

With respect to the Presentation Layer workflow, at present we basically see two different methods to retrieve the related information: a) by means of retrieval from the metadata repository via SPARQL queries (either direct or exposed through API functions), and b) by means of interpreting information which is directly annotated to the fragment as an own TimedText annotation track, e.g. as WebVTT[15], TTML[16] or similar data.

The related information can also be filtered through personalisation and contextualisation procedures.

User Feedback/Interactivity

The User Feedback/Interactivity use case covers all functionalities where actually information issued by the user permanently flows back to the platform, i.e. will be stored within the metadata repository. This includes user annotations, ratings, tagging, comments, sharing.

Edit/Structuring

The Edit/Structuring use case will cover the functionalities for editing and structuring video material by end users, with uploading, re-arranging parts of videos, interlinking, annotating and more. This does not cover, however, the professional or semi-professional annotation and linking tools required for the Backend platform.

[15] http://dev.w3.org/html5/webvtt/

[16] http://www.w3.org/TR/ttaf1-dfxp/

5 LinkedTV Platform and Architecture

The following Chapter describes the overall LinkedTV Platform and Architecture derived from the requirements listed in Chapter 2 and the LinkedTV Workflow as described in Chapter 3.

This document discusses and defines issues of WP1-4 only as far as they are cross WP- or platform relevant. We do not describe internal issues of the components to be developed within the WPs, such as the details of the editorial tools, the functionality of the components for analysis, annotation and enrichment, or the applications for end users.

5.1 Platform Design

The general requirements for the platform as defined in Chapter 2 allow to derive the basic design principles for the architecture:.

- The need for a **distributed platform with flexible communication means** has lead to employ an Service Oriented Architecture (SOA) with a division of the platform in three frameworks each consisting of several components. The use of REST Services ensures an efficient, flexible and fault tolerant communication between all components.
- The request for an **open, independent platform** will also be ensured by exposing all components through common service interfaces through REST based Web Services
- The requested **exchange of existing platform components by e.g. third party software** and **flexible connection to existing standard AV platforms** are enabled by already considering the relevant common interfaces and standard formats, which are perspectively needed to connect to other systems (e.g. MASM, CMS) and include existing software modules (e.g. for media resource analysis).
- The **distributed development of components** is supported by providing all functionality developed by different partners in separate components offering minimal, clearly defined REST Services; this approach allows for a distributed development of the components at different locations, which communicate to platform components at remote partner locations via REST services. For the joined development process common build guidelines will be defined, a common version control repository relying on Subversion[17] or github[18] will be established, as well as a LinkedTV issue tracking system.
- **Scalability, multi-user, multi-client, parallelism, load balancing:** SOA allows distributing platform components, data and applications on separate machines at different locations. The architecture accounts for the possibility of the analysis of videos in different languages at a time is required, a growing number of end users running, and

[17] http://subversion.apache.org/

[18] https://github.com/

the delivery on several severs in parallel. In addition, with a growing amount of video footage, the analysed videos can be distributed on remote servers or in the cloud.

- The **compliancy to existing or emerging standards:** the platform interfaces, protocols, formats and metadata will uptake relevant standards and build on available Open Source components already covering needed functionality as far as possible (see Appendix “Used Standards”).
- The support of **multilingualism**: the platform has to support different languages both on the metadata level, as well as interface level. This includes the integration of automatic translation and mapping services.

5.2 Architecture Overview

The LinkedTV Platform consists of three main parts:

- the **Analysis and Annotation Layer** containing the components and interfaces for the analysis, annotation and enrichment components
- the **Presentation Layer** containing the components for developing personalised LinkedTV applications for end users
- the **Linked Media Layer** containing all the metadata generated and the services to access them, as well as management tools.

Connected to the Analysis and Annotation Layer there will be additional editorial tools:

- the **Media Selection and Analysis Tool** to select new videos for analysis and include them into the platform.
- the **Annotation Tool** to adjust automatically generated annotations
- the **Hyperlinking Tool** for the manual insertion of links associated to certain objects in the video.

The Presentation Layer provides the basis for the development of specific end user components and applications:

- a **Hypervideo Player** for HTML5 to retrieve and view hyperlinked video and
- an **HbbTV 1.1 compliant player** for TV applications (dependent on feasibility study results)
- **Specific applications** to perform the LinkedTV Scenarios with individual user interfaces and features such as the Use Cases to be developed by Partner University of Mons.

Figure 6 shows the LinkedTV architecture overview:

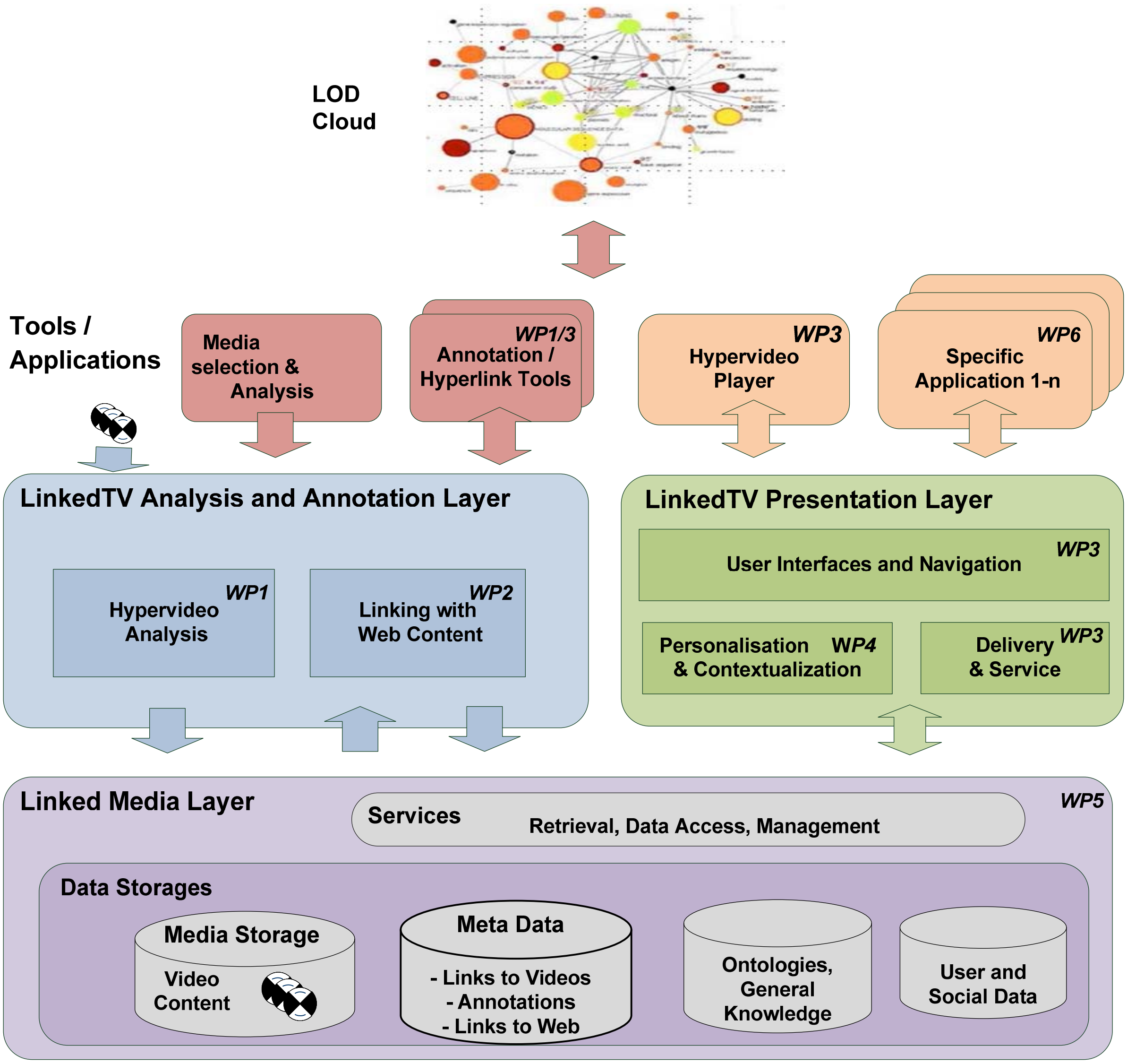

Figure 6: Architecture Overview

5.3 Linked TV Analysis and Annotation Layer

The Figure 7 below shows the Linked TV Analysis and Annotation Layer with the tools Media Selection & Analysis, Annotation and Hyperlinking and the components

- Hypervideo Analysis and
- Linking with Web Content.

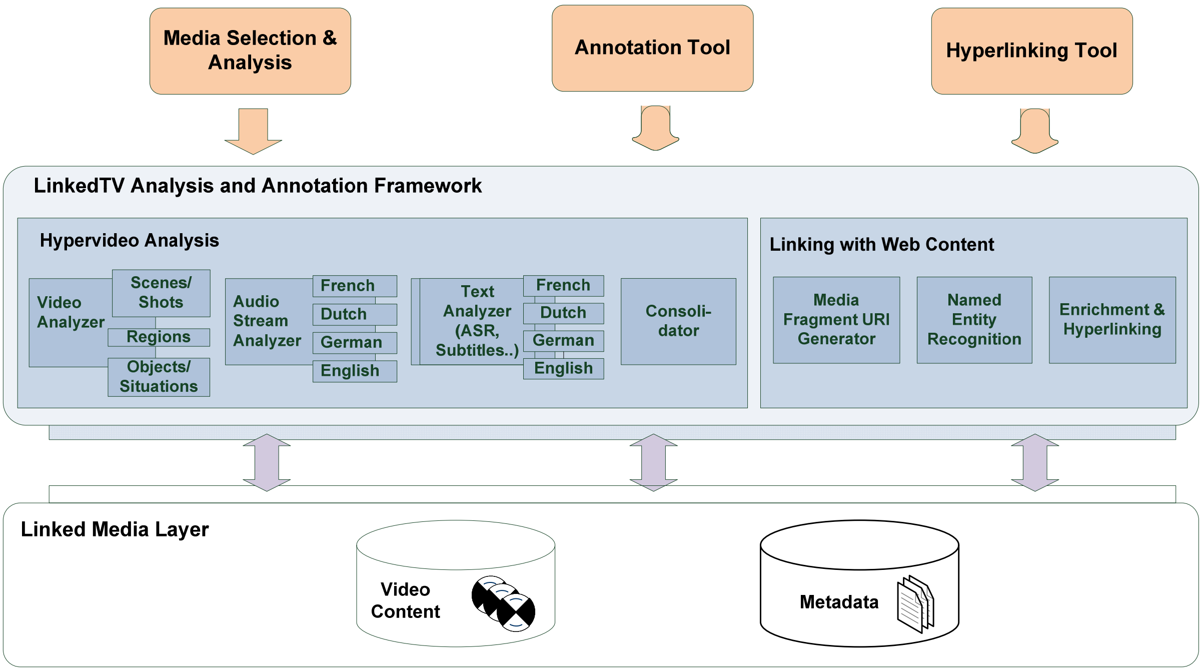

Figure 7: Components and Interfaces

5.3.1 Hypervideo Analysis

The objective of the intelligent Hypervideo Analysis component is the initial automatic and semi-automatic decomposition of external video media resources in order to recognize meaningful parts thereof. The original video files may either be accessible over the Internet via URIs, or belong to a local media archive. The analysis will go through a multistep process, where some steps can also be executed in parallel. Figure 8: Generic Composition of a Media Resource shows an overview of the analysis process.

A media resource is generally composed of different kinds of tracks and associated metadata (see Figure 8).

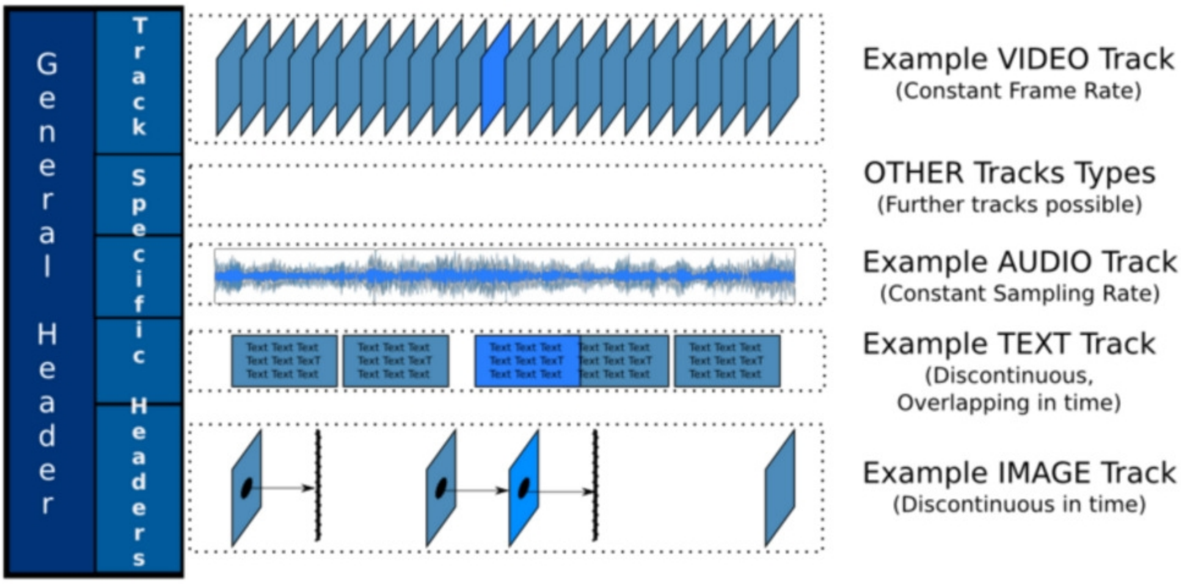

Figure 8: Generic Composition of a Media Resource[19]

Accordingly, the LinkedTV Platform will consist of different specialized analyzers for these different kinds of media resource parts as shown in the following figure with the components of the Hypervideo Analysis:

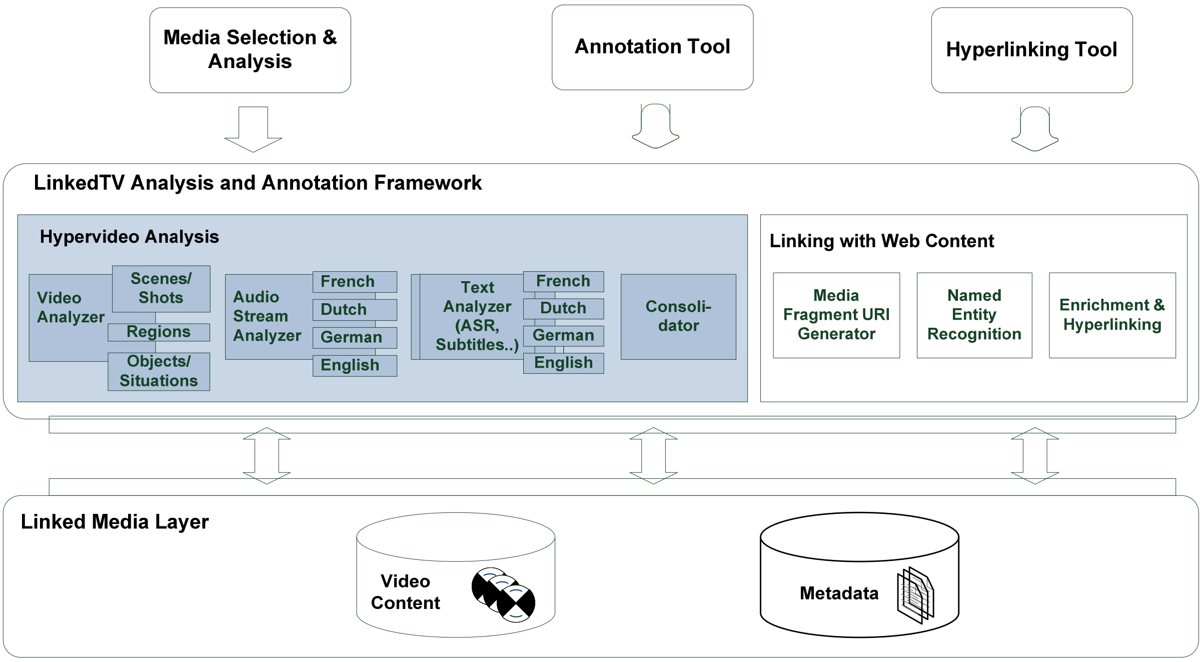

Figure 9: Hypervideo Analysis Components and Interfaces

[19] From *Use Cases and Requirements for Media Fragments* (http://www.w3c.org/TR/media-frags-reqs/)

The following subsection gives an overview of the initial LinkedTV analyzers. For a detailed description refer to *D1.1 .State of the art and requirements analysis for Hypervideo.*

5.3.1.1 Scene/Shot Segmentation

Scene/Shot Segmentation takes a video track as input and creates a segmentation according to the detected scenes and shots.

Feature	Description
Tools/Frameworks / Services	Specific tools developed by Partner CERTH
Input	Video track, mpg or mp4
Input Formats	Containers: mpg/mp4
Output Formats	RealText Streaming Text File (.rt): Begin times of scenes SMIL File (.smi): Container for all files belonging to one video
Standards	SMIL
Output Contents	Media Fragments with shot/scene segmentations presented as subtitles
Sample Output	SMI File <smil> <head> <meta name="title" content="BG_34872"/> <meta name="author" content="VIDIVIDEO"/> <meta name="copyright" content="L2F, INESC-ID / CERTH ©"/> <meta name="keywords" content="Stream"/> <meta name="description" content="Stream"/> <meta name="robots" content="all"/> <layout type="text/smil-basic-layout"> <root-layout width="480" height="320" background-color="black"/> <region id="movie" left="40" width="384" height="288" z-index="1"/> <region id="caption_region" top="289" width="480" height="20" z-index="2"/> </layout> </head> <body> <par> <text src="RBB_Sh.rt" region="caption_region"/> <text src="RBB_Sc.rt" region="caption_region_out"/> <video src="rbbaktuell_20111115_sdg_m_16_9_512x288.mp4?start=0.000000&end=1749.640000" region="movie" begin="0:00.000"/> </par> </body> </smil> RT File <window type="teleprompter" bgcolor="black" wordwrap="true"

Feature	Description
	duration="1745.640000" width="480" height="20"> <font face="arial" size="4" color="white"> <time begin= "0.00"> scene_1</time> <time begin= "4.12"> scene_2</time> <time begin= "26.16"> scene_3</time> <time begin= "194.64"> scene_4</time> <time begin= "206.52"> scene_5</time><clear/> etc.

5.3.1.2 Spatial Segmentation

Spatial Segmentation takes a video track as input and creates segmentation according to detected scenes and shots.

Feature	Description
Tools/Frameworks/ Services	Specific tools developed by Partner CERTH
Input	Video track, mpg or mp4
Input Formats	Containers: mpg/mp4
Output Formats	Currently JPEGs of Key Frames with colored regions; there might be also be metadata
Standards	
Output Contents	Media Fragments with spatial segmentations
Sample Output	

5.3.1.3 Concept Detection

Concept Detection takes a video track as input and creates a collection of concepts detected in the video.

Feature	Description
Tools/Frameworks/ Services	Specific tools developed by Partner CERTH
Input	Video track, mpg or mp4
Input Formats	Containers: mpg/mp4
Output Formats	File

Feature	Description
Standards	TRECVID 2011 Multimedia Concept Detection[20]
Output Contents	Media Fragments with concept annotations
Sample Output	See http://mklab.iti.gr/eventdetection-linkedtv/

5.3.1.4 Spatio-Temporal Segmentation

Spatio-Temporal Segmentation takes the video track as input and delivers a set of segments for objects identified in the video.

Feature	Description
Tools/Frameworks/ Services	Specific tools developed by Partner CERTH
Input	Video track, mpg or mp4
Input Formats	Containers: mpg/mp4
Output Formats	Metadata unclear
Standards	
Output Contents	Pixel based regions which contain detected moving object areas. Originally, they are not rectangular, but of arbitrary shape. In order to comply with the MediaFragment URI standard these have to be mapped to cover rectangular bounding boxes.
Sample Output	See ftp://ftp.condat.de/CERTH%20WP1%20Preliminary%20Results/Spatio-Temporal%20Segmentation/RBB_video_sample.gif

5.3.1.5 Audio Transcription

Audio Transcription takes an audio track as input and generates a timeline based textual representation of spoken text. The audio transcription is also planned to cover restricted **speaker recognition**, i.e. to detect male or female speakers and which speakers are the same within a source file.

Feature	Description
Tools/Frameworks/	Tool developed by partner Fraunhofer

[20] http://trecvid.nist.gov/

Services	
Input	Audio Track of Video
Input Formats	Mpeg4
Output Formats	EXMARaLDA/MPEG 7 XML file
Standards	Extensible Markup Language for Discourse Annotation, EXMARaLDA[21]
Output Contents	
Sample Output	(Excerpt) <ns2:Identifier scope="local">1</ns2:Identifier> </ns2:Speaker> <ns2:Transcription linguisticUnit="word" mediaTimeBase="../../MediaTime[1]/MediaTimePoint" mediaTimeUnit="PT1N1000F"> <ns2:StartTimeDurationMatrix ns1:dim="70 2">0 740 750 400 1160 360 1530 160 1700 340 2050 200 2260 260 2530 420 2960 270 3240 320 3570 370 3950 430 4390 170 4570 150 4730 520 5260 410 5680 200 5890 130 6030 510 6550 520 7080 330 7420 480 7910 210 8130 110 8250 680 8940 480 9430 160 9600 270 9880 310 10200 100 10310 150 10470 90 10570 540 11120 490 11620 290 11920 160 12090 590 12690 270 12970 310 13290 150 13450 400 13860 730 14600 230 14840 350 15200 160 15370 210 15590 360 15960 550 16520 430 16960 300 17270 350 17630 150 17790 590 18390 290 18690 420 19120 300 19430 200 19640 220 19870 450 20330 580 20920 330 21260 490 21760 80 21850 300 22160 470 22640 270 22920 180 23110 270 23390 330 23730 10</ns2:StartTimeDurationMatrix> <ns2:ConfidenceVector>0.9546061 0.5951297 0.9969207 0.9922752 0.97396106 0.9966717 0.9797387 0.7553037 0.9998871 0.99064684 0.78951627 0.813137 0.6202162 0.33452037 0.7453349 0.8856074 0.9848802 0.546146 0.567023 0.3963781 0.800529 0.3122735 0.7808803 0.8423455 0.26145047 0.33347648 0.97299206 0.14274582 0.9825881 0.38414204 0.24682751 0.9789249 0.34279773 0.999982 0.71487373 0.92707825 0.9728416 0.10203785 0.15879703 0.9166587 0.6956825 0.9409761 0.7619217 0.9442757 0.19104947 0.7673864 0.38858473 0.9925167 0.9896343 0.9877608 0.98460466 0.97531575 0.002895629 0.97365236 0.994466 0.9752958 0.018159552 0.9980718 0.95022595 0.23042935 0.9987974 0.1739223 0.9450726 0.86003053 0.81154424 0.9413383 0.9108244 0.8814247 0.05034456 1.0</ns2:ConfidenceVector> <ns2:SpokenUnitVector>sil als auf dem Land immer wieder Berichten über dieses Thema wobei es dann meist darum geht es Ernst wird dort fehlen also in Brandenburg Kaufhaus und Fracht als auch noch die Kliniken sil haben nicht genügend Mittel zieht an einigen Krankenhäusern gibt deshalb wurde schon immense Probleme jetzt ist Hilfe in Sicht Hilfe von weit her aus Litauen Lettland sil auf der ersten Behandlung steht noch das Lachen sil</ns2:SpokenUnitVector> </ns2:Transcription> </ns1:AudioDescriptor> </ns1:AudioSegment> <ns1:AudioSegment>

[21] http://www.exmaralda.org/

```
    <ns1:MediaTime>
        <ns1:MediaTimePoint>T00:00:23:77F100</ns1:MediaTimePoint>
        <ns1:MediaDuration>PT01S66N100F</ns1:MediaDuration>
    </ns1:MediaTime>
</ns1:AudioSegment>
<ns1:AudioSegment>
    <ns1:MediaTime>
         <ns1:MediaTimePoint>T00:00:25:43F100</ns1:MediaTimePoint>
         <ns1:MediaDuration>PT03S59N100F</ns1:MediaDuration>
    </ns1:MediaTime>
<ns1:AudioDescriptor xsi:type="ns2:SpokenContentType">
<ns2:Speaker gender="female">
```

5.3.1.6 Metadata Analyzer

The Metadata Analyzer takes accompanying general metadata as input, e.g. EPG data and analyzes it semantically through Named Entity Recognition.

Feature	Description
Tools/Frameworks/ Services	Tools developed by partners Fraunhofer, EURECOM and Condat
Input	Dependent on source material
Input Formats	TVAnytime, proprietary formats, and plain text
Output Formats	To be specified
Standards	TVAnytime, MPEG7
Output Contents	To be specified
Sample Output	To be specified

5.3.1.7 Consolidation

The metadata which is generated by the different analyzers needs to be integrated into a common format. It has to be ensured that all the different pieces can actually be merged together and that also metadata information which is aggregated through further processing or through manual addition can be integrated into a single timeline. Therefore, a consolidation phase merges and integrates these different results into a single result format and synchronises the different associated timelines as far as possible. The consolidation also covers the detection of inconsistencies and their elimination. The result of the consolidation process is an enriched MPEG7 compliant XML file containing all the different media fragments and their annotated information. This XML file serves as the exchange file between the Hypervideo Analysis components and the components for Enrichment and Linking to Web Content.

Consolidated Analysis Exchange File

There are several proposals available for annotation exchange formats [22,23], which include different concepts such as: i) explicit external / implicit timeline, ii) single / multi labeling for each annotation, iii) multi tiers for each type of annotation / single list and iv) different speakers in different tiers / the same tier.

The main measures to enable a fully and easy extraction of the relevant information for the Linking with Web Content and Enrichment tools are to employ an annotation graph formalism with

- a single explicit timeline; for all tools not generating output according a precise timeline, the annotations have to be adjusted to the best fitting time stamps.
- precise enough time stamps for all annotations as needed for the further processing (identify semantic relations, appropriate presentation for the end user, etc.).
- separate tiers for each track and
- separate tiers for different speakers.

We assume it as less important to have single labeling or place all items in the exact temporal order.

As most of the Hypervideo Analysis tools focus on specific video tracks without looking for relations to other tracks, they mainly insert all detected items in the exchange file ordered by the timeline.

The complete source videos are addressed by URIs, which allows an editor to identify and delete a video. This also enables the notification of the subsequent components through an event mechanism about the deletion of a video.

This exchange format allows the Linking with Web Content component i) to parse the annotations and generate for all relevant scenes, shots, texts, etc. Media Fragments according the MediaFragment URI 1.0 standard and ii) detect and connect annotations according their semantic relations (e.g. which graphical objects are in a shot) in order to support the further processing.

Example for exchange file using the extended EXMARaLDA Format

The data exchange uses an extended EXMARaLDA file format based on XML (cf. 5.3.1.5), as illustrated in the following example (in a free notation). The compatibility with EXMARaLDA also opens the potential to use other existing tools such as ELAN,[24] Theme[25] or Transformer[26].

[22] "An exchange format for multimodal annotations", http://embots.dfki.de/doc/Schmidtetal08.pdf

[23] further references: http://www.exmaralda.org/publikationen.html

[24] http://www.lat-mpi.eu/tools/tools/elan

[25] http://www.noldus.com/site/doc200403003

[26] http://www.oliverehmer.de/transformer/

```
Metadata associated to the whole video (e.g. according to
EBUCore/TVAnytime)

Timeline of Timestamps/Anchor Ids (in milliseconds):
   T0 ="0"
   T1 ="300"
   T2 ="1200"
   T3 ="2311"
   …
   Tn = …
Tier1: Scenes
   T0 – Ti: "Scene 1"
   Shots: T0, Tm, …, Tn
   Ti+1  - To: "Scene 2"
    …
Tier2: Speaker1, verbal
   T0 – T3: "text 1 ....."
   T3 - T4: ...
   Tn – Tn+1: "text i ....."
   …
Tier3: speaker2, verbal
   T4 – T5: "text 2 ...."
   ….
Tier4: subtitles
   T2 – T3: "text 3 ...."
   ….
Tier5: objects of video fragments
   T6 – T9: "visual_object1", rectangle p1,p2
   …..
Tier6: text in the video
   T10: "text 4", rectangle p2,p3.
```

Common LinkedTV Metadata Format

The parameters transferred by the XML file also serve as the basis for the Common LinkedTV Metadata Format, which will be specified in detail in the next project phase. This format will be based on existing and emerging standards such as *Media Fragments URI 1.0*, *Ontology for Media Resource 1.0*,[27] *WebVTT*[28] for timed text information, perhaps a relevant subset of *MXF*[29] (MPEG Material Exchange Format), and include mappings between different schemata. It should also use the general LinkedTV Ontology with different schemata for the respective dimensions; cf. Section 5.5.1.3 Ontologies.

5.3.2 Linking to Web Content

Figure 10 shows the part of the architecture containing the components for Media Fragment URI generation, NER, Enrichment and Hyperlinking:

[27] http://www.w3.org/TR/mediaont-10/

[28] http://dev.w3.org/html5/webvtt/

[29] http://www.pro-mpeg.org/

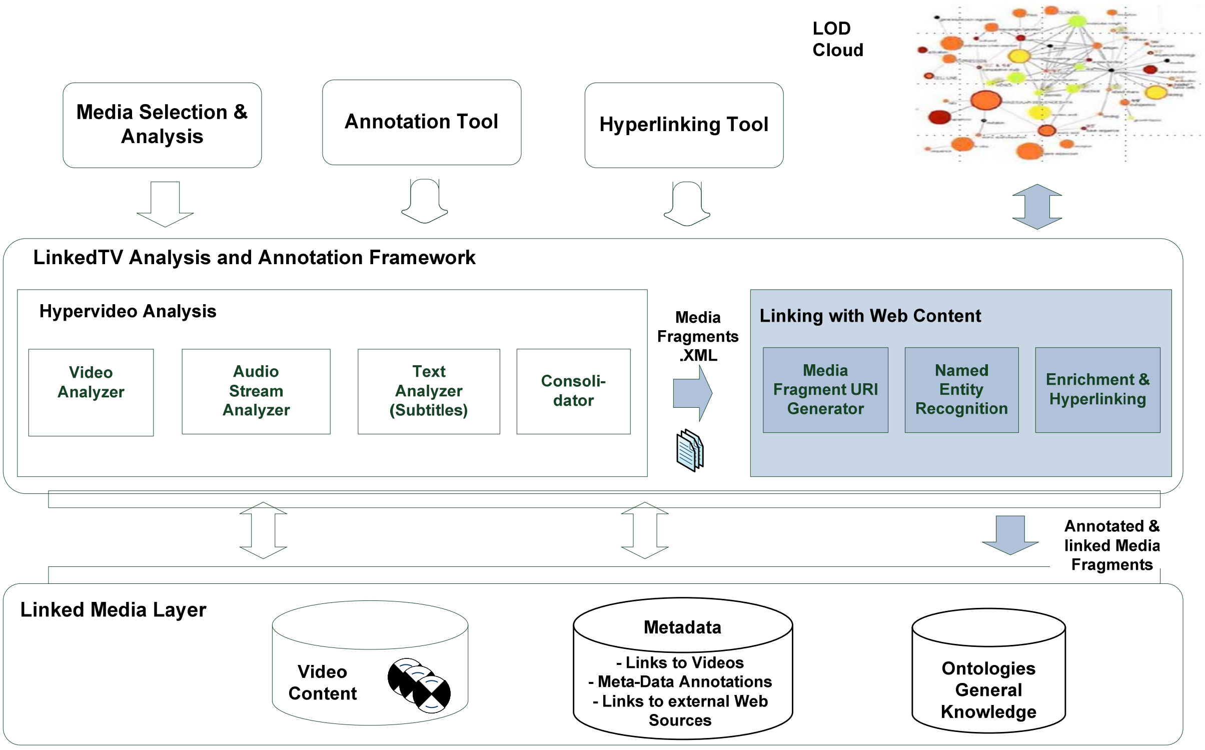

Figure 10: Components and Interfaces for Linking with Web Content

5.3.2.1 Media Fragment URIs Generator

The Media Fragment URIs Generator reads and parses the exchange file received from the Hypervideo Analysis. From each segment with annotations of the XML file a Media Fragment URI is generated and then stored in the Metadata Repository. This will later enable Media Fragment URI players and servers to directly retrieve and play the fragments. The component for the generation of Media Fragment URIs has to ensure that each fragment complies with the Media Fragment Standard 1.0.

The Media fragments support addressing the media along four dimensions:

Dimension	Description
Temporal	denotes a specific time range in the original media, such as "starting at second 10, continuing until second 20"
Spatial	denotes a specific range of pixels in the original media, such as "a rectangle with size (100,100) with its top-left at coordinate (10,10)"
Track	denotes one or more tracks in the original media, such as "the English audio and the video track"
ID	denotes a named temporal fragment within the original media, such as "chapter 2", and can be seen as a convenient way of specifying a temporal fragment

Examples of Media Fragment URIs are shown below:

```
http://www.example.com/example.ogv#a=b&c=d
http://www.example.com/example.ogv#t=10,20
http://www.example.com/example.ogv#track=audio&t=10,20
http://www.example.com/example.ogv#id=Cap%C3%ADtulo%202
```

The Media Fragment scheme is described in detail in Deliverable *D2.1 Specification of the Media Fragment URI scheme*.

5.3.2.2 Named Entity Recognition

Named Entity Recognition (NER) is first applied to the textual information, i.e. mainly subtitle track and audio transcript. This happens analyzing the textual fragments by means of the Meta-NER service NERD [30] by partner EURECOM [31]. It is a Web Service plugged on the top of various Named Entity (NE) extractors. It allows a user to analyze any textual resource, published on the Web and accessible through a URI, and to extract named entities through different NER Services such as Alchemy [32], DBpedia Spotlight [33], Extractiv [34], OpenCalais[35], Zemanta [36] Web APIs. Other NER services can be easily integrated as long as they provide an API, which is in particular important for the support of languages other than English.

Each extracted named entity is classified according to the NERD ontology[37]. The classification of the ontology provides an appropriate expressivity and granularity for the later personalization and contextualization.

Feature	Description
Tools/Frameworks/ Services	NERD Service
Input	Any textual file, URI
Input Formats	
Output Formats	JSON
Standards	

[30] nerd.eurecom.fr/ontology

[31] http://nerd.eurecom.fr/

[32] alchemyapi.com

[33] wiki.dbpedia.org/spotlight

[34] wiki.extractiv.com

[35] opencalais.com

[36] developer.zemanta.com

[37] http://nerd.eurecom.fr/ontology

Feature	Description
Output Contents	Detected Entities

Sample Output	`===========================================================`
	`SPOTLIGHT`
	`[{"idExtraction":2653,"NE":"Ärzte","URI":"http://dbpedia.org/resource/Die_%C3%84rzte","confidenceScore":0.427461,"NERDType":"http://nerd.eurecom.fr/ontology#"}]`
	`===========================================================`
	`EVRI`
	`[]`
	`===========================================================`
	`EXTRACTIV`
	`[{"idExtraction":2654,"NE":"Brandenburg","type":"LOCATION","URI":"http://dbpedia.org/resource/Brandenburg","confidenceScore":0.0,"NERDType":"http://nerd.eurecom.fr/ontology#Place"},`
	`{"idExtraction":2655,"NE":"Lettland","type":"COMMERCIAL_ORG","URI":"http://dbpedia.org/resource/Latvia","confidenceScore":0.0,"NERDType":"http://nerd.eurecom.fr/ontology#Company"}]`
	`===========================================================`
	`OPENCALAIS`
	`[]`

5.3.2.3 Enrichment

After the basic analysis and annotation of the media resources has been performed the next step is to enhance these annotations in order to get truly rich semantic descriptions of the items and fragments and to interlink them on a deep level with content and data on the web. That is, through the enrichment process further annotations are attached to media resources.

For every detected entity the NERD tool returns a NERD ontology URI, which is further linked to the respective entities in general Linked Open Data encyclopaedic repositories such as dbpedia[38] or Freebase[39], or specialized LOD sources such as musicbrainz[40] or the Internet Movie Database.[41] With these references further RDF structured information can be retrieved and linked to each entity, thus giving rise to a semantic embedding of the items contained in media resources within the whole Web.

The enrichment process is based on the named entities initially recognized, not on the media fragments themselves. This also means that the main starting point for enrichment is not so the completion of a media resource analysis, but rather the addition of new named entities.

Input/Output

The following table lists input and output of the Enrichment component:

[38] dbpedia.org

[39] freebase.com

[40] musicbrainz.org

[41] imdb.org

Input	Description
Events	Events generated by the platform or external feeds which can trigger enrichment are: - Media resource analysis completed - New Media Fragment added - New Annotation/Named Entity added - New type of connector added (see below) - Regular: update on a regular basis, because Web content is always changing
Data	Data which can serve as input for the enrichment process - Named Entities (independent from media resources) - Media Resource Metadata - Other?

Output	Description
Events	Events which can be triggered after enrichment is finished: - *At present none, perspectively remote applications, which refer to the LinkedTV repository.*
Data	Data which is added through the enrichment process - References/mappings to LOD URIs - References to Web resources (URIs, microformats) - References to Web Services (e.g. REST URIs) - Other …

The Enrichment component will include connectors for different types of sources. The project will use existing connectors, extend them and implement completely new ones. At least the following list of connectors is planned for the project.

Connector Types	Description
LOD Connectors	Adds domain specific LOD references dependent on the type of a named entity
General Web Content Connectors	Adds unstructured web resources through web mining
Web API Connectors	These are connectors which enrich through using APIs available for specific web content services, e.g. facebook, twitter, Google Maps, Ebay, etc.

5.3.2.4 Hyperlinking

Although enrichment is also adding hyperlinks to the Linked Media Layer, Hyperlinking within the LinkedTV process means specifically adding hyperlinks to Media Resources or Media Fragments. This can be:

- Linking Temporal Media Fragments to each other, e.g. for generating playlists or multimedia presentations
- Linking annotations to media fragments, i.e. Wikipedia entry or a Google Maps Link to a certain region
- Linking Media Fragments to other sources on the web, e.g. to an Ebay query for the same objects as recognized in the media fragment.

5.3.3 Tools

The LinkedTV Platform provides tools for administration and editing workflows, annotation and Hyperlinking.

5.3.3.1 Media Selection and Analysis Tool

The Media Selection and Analysis Tool allows to select new videos for analysis to be included into the platform and launch the analysis process. The tool supports the editor to monitor the analysis progress by showing the status of the workflow (e.g. completion of analyzed tracks, NER or linked with LOD ...) and results (e.g. number of generated fragments of each type, annotations and links, severe or minor failures).

5.3.3.2 Manual Annotation Tool

The publishing process for LinkedTV applications requires that media fragment annotations can also be added and corrected through manual edition. Therefore a dedicated Annotation Tool is provided for editors to correct annotations which have been set previously during the automatic analysis. Initially, the Annotation Tool shows the metadata annotations of the video associated to segments and frames of the video. The metadata can be changed, deleted and new ones inserted.

A first version of the Annotation Tool will be described in *D1.3 LinkedTV Annotation Tool.*

5.3.3.3 Manual Hyperlinking Tool

As media resource hyperlinking is very dependent on context and intended audience, fully automatic identification of relevant concepts is only limited. Therefore a Manual Hyperlinking Tool will be provided for working with the LinkedTV platform. This is a specialized video editing tool, which provides functionalities such as playing, stepping through it frame by frame, insert links and position them in the frame. The tool also enables the definition of elements to be displayed as overlay over the video on the end user device. It is planned to define a generic set of elements such as subtitle, hyperlink, additional information to an object, etc.

Since this tool supports to develop smooth non-linear multi-media presentations, the editor will probably have functions which allow foreseeing how the resulting linked videos will be presented to the end user.

As it is closely related to the Annotation Tool a first version of the Hyperlinking Tool will also be described in *D1.3 LinkedTV Annotation Tool.*

5.3.4 Support of different languages

Whereas scene/shot segmentation and visual object recognition does not depend on the language, audio and subtitle tracks are language dependent. For the LinkedTV scenarios original material will be available in Dutch, French and German. The common metadata language within the platform, however, will be English. In order to support multi-language annotation of media fragments, a two-way translation support should occur:

- Transcripts should be automatically translated into English, so that for each media resource always an English base transcript is available. This translation does not have to be perfect, as it is normally sufficient to extract named entities from it. The integration of an automatic translation service like Google Translate should be sufficient for this purpose.
- The schemata used for annotations and named entity extraction should always contain the basic English term, but also alternate terms in different languages, e.g. by employing the SKOS (Simple Knowledge Organization System) [42] meta schema, which allows to label each item with designators in different languages.

5.3.5 Event-oriented Platform Interaction

All tools and components that are part of the Analysis and Annotation Layer interact with the core platform and with each other on the basis of an event-oriented approach. They have to be registered as Publish/Subscribe components at the Linked Media Service Bus (see Section 5.5.2.1).

The data which is produced by the analyzers is not stored within the Metadata Repository directly, but rather as files which can be processed by subsequent analyzers. Each relevant processing step issues an event which is published through notifications to the platform. Components which need to be notified can subscribe to the events.

Each event will include at least the following parameters:

Event Parameter	Description
TimeStamp	Date/Time of the event
EventName	The Name or ID of the event, such as VISUAL_CONCEPT_DETECTION_FINISHED
MediaResourceURI	The URI to the original media resource

[42] http://www.w3.org/2004/02/skos/

Event Parameter	Description
ResultURI	The URI to the result file; these can be pointers to ftp-sources or as well REST URIs which return the result content directly
ResultFormat	The format of the result, such as XML, RDF, JSON, CSV, TXT etc.

5.4 LinkedTV Presentation Layer

The LinkedTV Presentation Layer covers the frontend part of the LinkedTV platform. It provides the basic services to access the metadata generated by the analysis and enrichment components as well as the video content for delivering annotated media fragments. Furthermore it includes the basic frameworks for the development of the applications as envisioned by the user scenarios, including the LinkedTV media player, the LinkedTV HbbTV application and personalization features.

5.4.1 Overview

The purpose of the LinkedTV Presentation Layer is to provide generic components and APIs for the development of LinkedTV applications as derived from the requirements analysis. Although the requirements are not finalized by now the general architecture can be already designed on a conceptual level.

The Presentation Layer itself relies on the Linked Media Layer and consists of three main parts:

1. User Interfaces and Navigation: contains the interface elements and functionality
2. Personalisation and Contextualisation: contains underlying APIs for advanced personalisation and contextualisation methods
3. Media Resource Delivery: contains the APIs for the actual delivery of annotated Media Fragments.

Figure 11 gives an overview of the Presentation Layer architecture. The different components will be described in more detail in the following section.

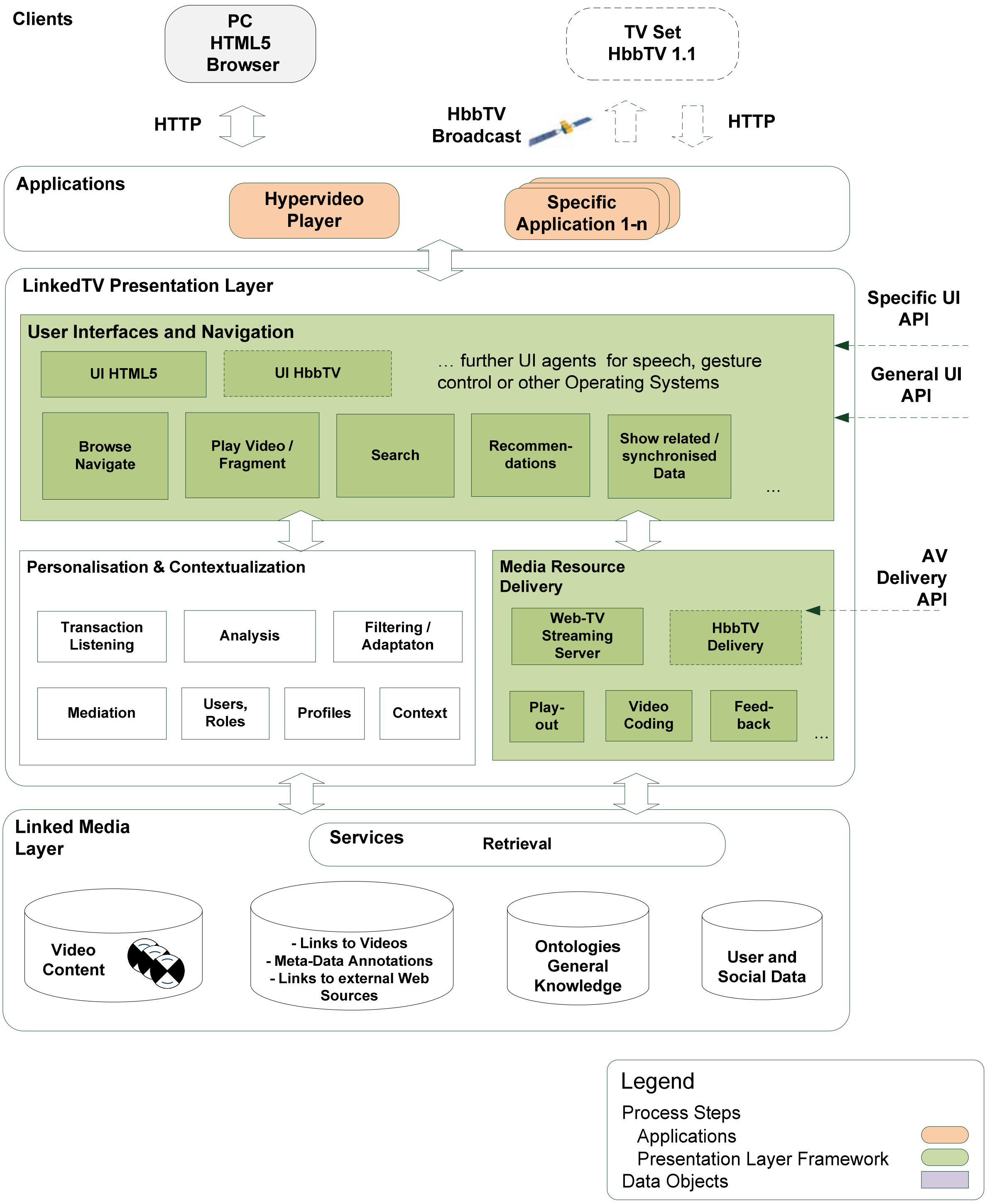

Figure 11: Components of the Presentation Layer

5.4.2 Clients

The LinkedTV Presentation Layer will basically support two client variants: a) clients based on HTML5 compliant user agents, whether these will be browsers or Web Apps based on HTML5 and b) clients based on the current HbbTV 1.1 specification. As we expect that the HTML5 standard will be adopted by HbbTV within the next versions as well, the latter is primarily foreseen to be able to demonstrate LinkedTV applications on present day TV sets as well. Of course, they will able to include only a limited subset of functionality and a limited controllability of the interaction compared to HTML5 based applications.

As well we will see further development of frameworks which go beyond the HTML5 standard, like *Webkit*[43], and also further development and emergence of new media related JavaScript frameworks, like the HTML5 Media Framework *Popcorn.js*[44]*, MediaElement.js*[45] *or Kaltura Video Library Solution.js* [46]*.* However, with respect to the LinkedTV Presentation Layer these should be able to use just the same API services and do not need to be reflected by the architecture itself.

Besides these presentation frameworks there are also different kinds of clients for interaction, behavior tracking and input, like mouse, keyboard, touch interface, remote control, speech UI or gesture devices like MS Kinect. However, they are conceptually considered and could be subject of perspective LinkedTV development phases.

5.4.3 Applications

5.4.3.1 Hypervideo Player

The Hypervideo Player is the standard end user interface of LinkedTV for the use cases, which is designed to take full advantage of the semantically annotated and interlinked media fragments. The Hypervideo Player reads the video materials provided by the LinkedTV platform together with the identified media fragments, annotations and elements to be displayed over the screen layer. It employs video hotspot technology which gives end-users and content creators the option to find objects using layered technology enabling an enriched user experience when viewing videos. The Player uses the services of the application layer to show and play the videos and place links and display elements on the canvas. A dynamic click on hyperlinks or hotspots will enable to navigate to related content in the video itself or of other available media in the Web.

The LinkedTV project also plans to specify and develop a version of the Hypervideo Player which is restricted to the limits of HbbTV 1.1 (see also 5.4.4.3).

Both versions will be described in detail in *D3.4 LinkedTV interface and presentation engine V1* (September 2012).

5.4.3.2 Specific Applications

Certain applications, in particular those developed by Université de Mons, require additional client components. E.g. the Use Case Scenario “VideoCollage” needs specific user interface elements for presentation, editing and composition of video snippets to create a collage.

[43] http://www.webkit.org/

[44] http://popcornjs.org/

[45] http://mediaelementjs.com/

[46] http://www.kaltura.org/project/HTML5_Video_Media_JavaScript_Library

5.4.4 The LinkedTV Presentation Layer

The HTML5/HbbTV Hypervideo players and the specific applications are all developed on the basis of the Presentation Layer API, which offers the basic functions to navigate, set profiles or play a Media Fragment. Within the following subsections the different components of the LinkedTV Presentation Layer are described.

5.4.4.1 User Interfaces and Navigation

The User Interface and Navigation component has two main layers:

a) The Generic User Interface Layer

b) The User Agent Specific Interface Layer

The Generic User Interface Layer hides all access calls to the Linked Media Layer from user agent specific realizations.

According to the workflow definition (cf Figure 5) the Generic User Interface Layer consists of the following subcomponents (the concrete service signatures with parameters and return values will be described in the project wiki when the specification of the use case scenarios is finalized). Note that there is no 1:1 mapping between the workflow definition and the subcomponents described here, as e.g. the “Play” use case is part of the Media Delivery Framework.

Subcomponent	Description
Search	Services for searching which take some input as search expressions (which can be text like “Silver Box”, an image displaying a silver box, or an URI which points to a media fragment which features a silver box, or structured metadata information) and queries the repository for all items which match the search expression, in this case all e.g. media fragments which are annotated as containing silver boxes. Due to the fact that the items are semantically annotated, semantically related items can be retrieved, like “golden chest”, “all boxes”, or “all items annotated as being from the same time period or location of origin”.
Recommendation	The recommendation is very similar to the search function, but more consequently considers the user profile and preferences using the filter mechanisms of the Personalization and Contextualization framework.
Navigate	Services for semantically qualified browsing and navigation through collections of media resources, like `getNextFragment, getTopFragment, getRelatedFragments, getFragmentWithProperty`[47], etc.

[47] Function names are to be seen only logically here. The actual methods will be defined in a REST style denotation.

Subcomponent	Description
Play	Services for playing annotated MediaFragments, getting time fragments, region fragments, Tracks, etc.
RelatedInformation	Services for retrieving and displaying information related to a complete MediaResource or MediaFragment, like `getAnnotations, getIdentifiedObjects, getLinks,` getTimedText
UserProfile	Services for setting individual profiles or role profiles, to switch profiling on or off, setting user preferences etc. These methods will then passed to the Personalisation and Contextualisation package (see below)
Interaction	This package contains the services required by an application when it needs to send back information which has to be stored or processed by components of the platform, like `addAnnotation`, or `uploadMediaResource`. This package will also cover services needed for editing and structuring fragments.

Additional packages will be provided when required by the use case scenarios.

User Agent Specific Interface Layer

User Agent Specific Interfaces are built on top of the Generic Interface Layer. They produce output which is tailored to specific types of user agents or client technology. Within LinkedTV there will be two main types supported: a) the HTML5 based Video Player and as well the specific applications based on HTML5, and b) the HbbTV 1.1 based client.

Most notably the difference will be in the usage of the `<video>`, `<canvas>` and `<track>` elements. Whereas the HTML5 based User Agent will implement Play methods by making use of these elements, the HbbTV based User Agent can only render HTML pages based on CE-HTML and using the `<object>` element. Also, only a more limited number of codecs (MPEG2, H.264/AVC) is supported.

Additional types of user agents can be added later, e.g. User Agents based on native Android or iOS environments.

HTML5 User Agent Layer

The HTML5 User Agent Layer is built on top of the Generic User Agent Interface Layer. It provides the basic services for developing the LinkedTV HTML5 Hypervideo Player, which will comprise of the following main parts:

The WebTV User Agent Layer is built on top of the Generic User Agent Interface Layer. Clients will be based on an embed tag to provide playback on multiple devices/systems. The embed tag will autoswitch between flash, html5 and iOS playout depending on the browsers user agent.

The embed tag will also allow for more highlevel concepts as playlists, portal functions and exhibitions to be implemented. If possible the player will use HTML5 and during the project we will implement more features once HTML5 is able to support them. Flash will be the fallback for features that are not yet possible or very hard to implement in HTML5 like webcam, upload, and IPR protection.

HbbTV User Agent Layer

The HbbTV User Agent Layer will be specified within the next phase of the LinkedTV project.

5.4.4.2 Media Resource Delivery

The purpose of the Media Resource Delivery Framework is to provide the components needed for retrieval, selection and delivery of media resources, in particular media fragments. As such, it has to implement the Media Fragment URI specification,[48] which will enable the development of a LinkedTV Media Fragment Server.[49] It also includes the generation of ordered collections of media fragments (playlists).

The Media Resource Delivery Framework will comprise the following main subcomponents:

Subcomponent	Description
PlayList	Services for the generation of playlists containing annotated media fragments
MediaFragment	Services for the resolution of URI fragments or URI queries returning the appropriate byte ranges from the complete media resource.
Stream	Services for streaming of media fragments and complete resources.[50]

WebTV Streaming Server

Within the LinkedTV platform a dedicated Streaming Server will be developed based on the above described subcomponents. The LinkedTV project needs the streaming support for Media Fragments, which will be provided optionally by the WebTV Streaming Server or Hypervideo player [51]**.** The WebTV Streaming Server offers for the developer the AV Delivery API with the following functions to start and stop Http-streaming and receive feedback from Users:

[48] http://www.w3.org/2008/WebVideo/Fragments/WD-media-fragments-spec/

[49] This does not exclude the possibility that media fragment resolution can also be realized by the LinkedTV clients to be developed.

[50] Relevant standards like the recently published MPEG DASH standard will be considered. However, adaptive streaming is not a goal of the project itself, regarding MPEG DASH standard see also: mpeg.chiariglione.org/meetings/geneva11-1/geneva_press.htm

[51] See Media Fragment URI 1.0, Sections “Clients Displaying Media Fragments” and “Media Fragment Servers”

- Open Video for streaming
- Set parameters (such as formats, bandwidth, colour …) to control streaming
- Start / Stop stream
- Close Video for streaming.

Delivery will be split into 2 types of services. The first is meta information needed for playout for example format data, playlists, exhibitions, time/spatial information. This will be a xml stream that is requested by the playout client. The second type of services are content services for example video, audio, images, documents and screenshots. These will be controlled by the player based on the meta information stream.

5.4.4.3 Analysis of options for HbbTV client variant

For the first phase of the project the plan foresees a HTML5 variant for the browser, because this is the most appropriate front end for interactive TV by offering a two-way communication and clickable video.

For the second phase of the project a HbbTV variant is planned. This variant requires a more detailed feasibility analysis, because the envisioned interactive features of LinkedTV need to be provided for HbbTV under consideration of the following aspects:

1. Clickable Video is not fully possible, because an overlay of Hyperlinks on top of the HbbTV video layer can only be realized in a restricted form. The main reason is a missing synchronization support and a lower performance than PCs caused by the limited CPU power of TV-Sets; these constraints only allow to use Hyperlink overlays for applications with slowly moving objects.
2. HbbTV 1.1 only supports CE-HTML, which offers less features than HTML5.
3. non-linear Video is not fully possible, because the Broadcast always goes on and the user can hardly return to a determined point of the Broadcast.
4. Since standard TV-Sets offer no pointing devices, an alternative UI to select objects has to be provided [52].

There are several options to be analyzed:

- Offering a UI concept adapted for HbbTV 1.1:
    - As non-linear video is not fully possible with HbbTV, the HTTP data could only concentrate on background or detailed information shown in a separate window; after finishing the excursion, the user returns to the current scene of broadcast
    - The TV-Set or STB CPU does not allow to place Hyperlinks above fast moving objects in the video; they could be placed in fixed placed windows at the UI.

[52] Several manufacturers will offer Smart TVs with various types of remote controls (mouse, Wii-like pointing, cameras supporting gesture controls, etc.) the next years. However, we do not expect, that TV Sets with pointing devices and standardized interfaces will be widespread available in the next 3 years. Therefore we plan to use alternative means in LinkedTV, but already consider a later switch to pointing devices.

  - Applications without fast changing video content can be offered with Hyperlinks on top of the video layer, which allow for a sufficient precise synchronization of video and GUI elements.
  - Due to the missing pointing device, the selection of elements or clicking on hyperlinks is offered e.g. by using arrows or numbers on the remote control.
  - A second screen device directly connected to the TV like in several existing Smart TV applications might help to solve these issues; other approaches could be to use additional devices for voice or gesture control.
- Concentrate on the upcoming HbbTV 2.0 version for HTML5. The planned upgrade of HbbTV for HTML5 would allow LinkedTV to provide a single solution for both variants. The usage of HbbTV 2.0 resolves the main restrictions 1)”Clickable video” and 2) “CE-HTML”, only 3)”no non linear video” and 4)”Pointing device” still have to be considered.

5.4.4.4 Personalization and Contextualization Framework

The Personalization and Contextualization Framework covers the services to support adaption of content to user preferences and viewing contexts. This includes gathering of user preferences and context information, user profile administration, learning algorithms, adaptation of content to be displayed, and more. For an in-depth discussion of the personalization and contextualization features planned within LinkedTV see *D4.1: Specification of user profiling and contextualization.*

Figure 12 shows the Personalization and Contextualization Framework within the context of the LinkedTV platform.

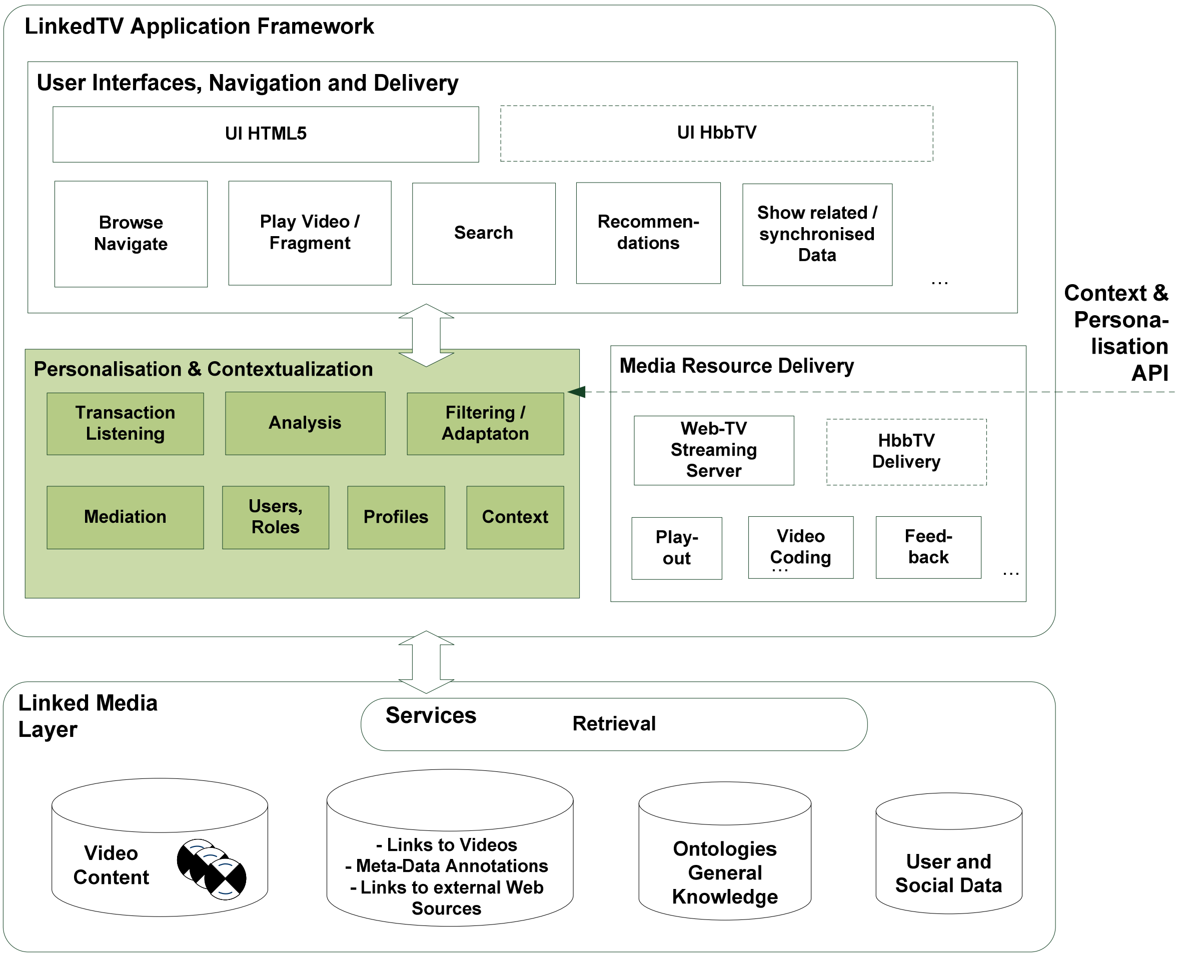

Figure 12: Components and Interfaces

The Personalization and Contextualization Framework will consist of three main parts: the first part **Transaction Listening** covers the gathering of implicit and explicit user information by means of user preferences, viewing history data, tracking user behavior, or recording contextual information. The second part **Analysis** covers the different methods and rule sets for analyzing the data and define the algorithms for contextualization and personalization. The third part **Filtering and Adapting** consists of actually applying these algorithms to media resources and annotations in order to generate the personalized and contextualized content.

The framework will include the following components and subcomponents:

Component / Subcomponent	Description
Transaction Listening	Due to privacy reasons as much information as possible must remain on the client and should be processed there. The Transaction Listener will be embedded in the LinkedTV platform, listening to user transactions. However, selected information about user transaction which is anonymized may be sent via a Mediation cloud to the platform.
Mediation cloud	In the mediation cloud content mappings and further classification will be elicited in order to retrieve the semantic representation of the user transaction. The individual user profile will be stored and be made available for inferencing always on the client. Transaction assessment and concept weighting will also be performed always on the client.

Analysis

Profiling	Services for building implicit user profiles. This subcomponent covers the different profiling methods which will be used and experimented with in LinkedTV.
History	Services for getting, setting, storing and retrieval of the user’s history data.
OperationMode	Services to support interface activities, like activating and deactivating implicit profiling, or switching between different profiles.

Filtering / Adaptation

Access	Services for managing user access and policies; this also includes authorization and authentication. Multiple authentication options might be offered, such as OpenID, facebook, twitter, or a separate LinkedTV access policy.
Personalize	Services for generating personalized content through filtering, i.e. services which take actual search results, annotations, etc, and match them to a given user profile and return filtered results.
Contextualize	Services which adapt content to a given context, like location, device, time etc. Contextualisation services per se do not take necessarily personal profiles into account, but will normally combined to generate the Contextualised User Model (CUM).
Users / Roles	Services for managing different kind of users and roles like Administrator, Editor, Parent, Child, etc. User Roles can as will be used to define access rights in general, as well as to define rules for the adaptation of content.
Profiles	Services for getting, setting, storing and retrieval of explicit user preferences as well as dynamically mined user data gathered e.g. through behavior tracking, learning, semantic classification or other techniques. These will be encoded by using the general LinkedTV

	ontology, which includes information on content related data like genres, actors, locations etc. and also context related data like device, time-of-day, location, etc. Explicit user preferences might not only include positive preferences (interests), but also negative ones (disinterests). User Profiles are always stored on the client.
Context	Services for getting and setting context information. Context parameters possibly include device information, location, time, input from sensors, camera, microphone, tracking devices, and more.

Server and client side functions

Dependent on different user scenarios and user agents personalisation and contextualisation can be processed on the client or on the server. In order to minimize or even avoid server-based storage of personal information altogether due to privacy concerns, within rich-client scenarios all personalisation will be done completely on the client. On the other, with limited client capabilities, as for instance in HbbTV applications, some anonymized personalisation processing may be done on the server.

Therefore the LinkedTV personalisation and contextualisation services will support two integration modes: 1) server-side as part of the platform services to be exposed via REST web services, and 2) client-side as Java or JavaScript frameworks which can directly be integrated in the client application.

Privacy constraints

Since privacy is a crucial point about the system acceptance by the users, the following measures will be undertaken to avoid violating people intimacy:

- Absolutely no rough confidential data will be stored anywhere (client or server)
- The features extracted from the rough data will be stored in a temporary location on the server. After some minutes, those features will be definitely deleted.
- The personalized content decision will be done on the client to avoid sending private information to the server.
- If data are sent to the server, they will be first aggregated into a contextual instance (fingerprint) of the profile and anonymized.

Interfaces

The framework will have the following main interfaces to other components of the platform:

Subcomponent	**Description**
Core Framework	
Metadata	The framework needs to retrieve the annotations of Media Fragments in order to match these with user profile data. This will be achieved through SPARQL queries against the

Subcomponent	**Description**
Repository	Metadata RDF repository.
LinkedTV Ontology	The frameworks need access to the general LinkedTV ontology in order to model user preferences, e.g. genres, languages, locations, events, TV series, directors, object categories, etc. which are also used to denote annotations.
Presentation Layer	
User Interface	The User Interface Layer will communicate with the Personalization/Contextualization Interface in order to submit user and context information and to get the resulting content adapted to the user's preferences and viewing context.

5.5 Linked Media Layer

The Linked Media Layer provides all services for accessing the persistency components such as storage for video content, annotations, ontologies and user data as well as functions for retrieval, data access and management. It also provides the central integration communication infrastructure. Thus, the Linked Media Layer consists itself of three main sub-layers (Figure 13):

Repository Layer: includes all persistency storage facilities for videos, metadata, ontologies, personalization and context data, as well as for application specific data.

Integration Layer: includes the central integration and communication component, the Linked Media Service Bus.

Service Layer: provides API services for retrieval, accessing and manipulating data stored in the repositories.

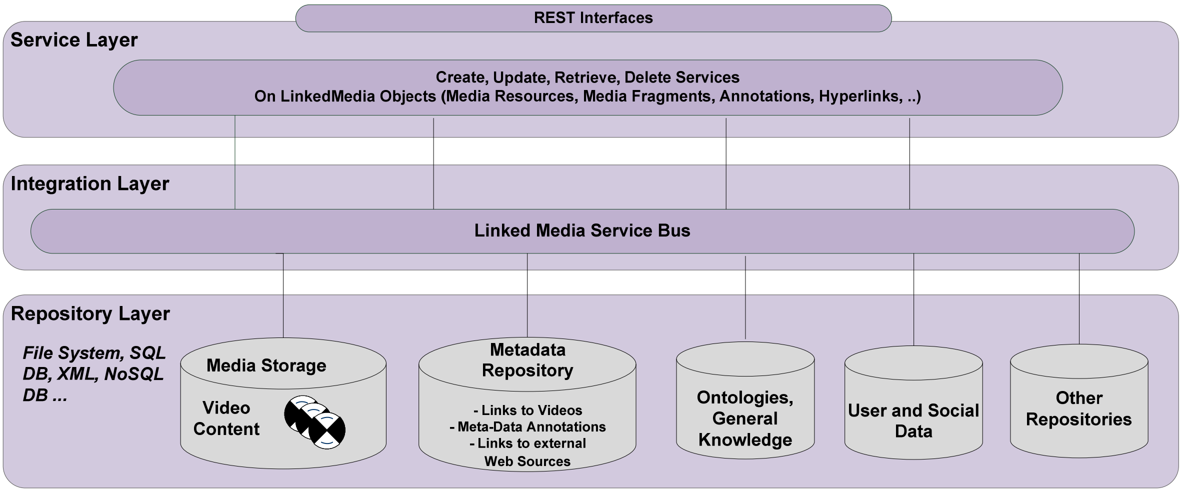
REST Interfaces
Service Layer
Create, Update, Retrieve, Delete Services
On LinkedMedia Objects (Media Resources, Media Fragments, Annotations, Hyperlinks, ..)
Integration Layer
Linked Media Service Bus
Repository Layer
File System, SQL DB, XML, NoSQL DB ...
Media Storage
Video Content
Metadata Repository
- Links to Videos
- Meta-Data Annotations
- Links to external Web Sources
Ontologies, General Knowledge
User and Social Data
Other Repositories

Figure 13: Linked Media Layer

5.5.1 Repository Layer

The Repository Layer is usually located on the server side of the platform. However, some smaller data sets, such as for the user profile or history could also be stored on the client side.

For the Metadata Repository, the Ontologies and User and Social Data a RDF repository is provided, most likely OpenLink Virtuoso [53], which also offers other types of storages (SQL, XML, Documents, Free Text, Linked Data).

5.5.1.1 Media Storage

In general, LinkedTV will not require that media resources which are being analyzed and annotated are stored within the platform itself, but can rather be accessed either through URIs over the Internet or by connecting to media archives (e.g. to that of a broadcaster). However, the platform will support the storage of media resources itself which is needed in particular for the analysis part.

5.5.1.2 Metadata Repository

The Metadata Repository is the central part of the LinkedTV platform, as it contains the metadata aggregated through the different analysis, fragment detection and annotation processes. It will contain the following data (non-comprehensive):

Type of Metadata	**Description**
Base Resource URI	URI reference to the original and complete media resource
General Metadata	Metadata which applies to the base resource as a whole, like *title, duration, owner, author, format, genre*, etc. The General Metadata will be based on EBUCore.
Track URIs	URI references to different tracks of the same base resource, e.g. different audio tracks for different languages. These can also be textual transcripts of the audio track generated by LinkedTV analysis tools
Media Fragments	Metadata which refers to fragments of a given base media resource. These can be temporal fragments (e.g. scenes and shots), spatial fragments, track fragments, or named fragments.
Media Annotations	Metadata which contains annotations associated to complete media resources or media fragments, either generated automatically or edited manually. The annotations will be based on the general LinkedTV ontology, but might probably also contain free text or tags.

[53] http://virtuoso.openlinksw.com/

Type of Metadata	Description
Media Hyperlinks	Metadata which contains links from media resources (complete or fragments) to other media resources, to general resources on the Web, or to LOD sources in particular like dbpedia,[54] Freebase,[55] etc. These hyperlinks can be automatically aggregated as well as manually added.

5.5.1.3 Ontologies

The Ontology Repository contains the General LinkedTV Ontology as well as domain or application specific ontologies. The ontologies might be physically stored in the repository itself, or just be references to ontology resources on the web.

The following table lists some preliminary ontologies or metadata schemes which are currently considered to be used within LinkedTV. The examination of relevant ontologies will be addressed in *D2.2 Specification of lightweight metadata models for multimedia annotation.*

Type of Metadata	Description
NERD Ontology	Scope: General Domain Knowledge The NERD Ontology, developed by Partner EURECOM, is a meta ontology mapping entity types from different Named Entity Recognition tools, currently the following: Alchemy, DBpedia Spotlight, Extractiv, OpenCalais and Zemanta. The NERD Ontology can be easily extended by integrating additional NER Tools when needed.
schema.org	Scope: General Domain Knowledge schema.org is a shared vocabulary supported by the major search engine providers Google, Microsoft, and Yahoo! Intended to be used for markup of web pages in order to support better knowledge based indexing.
LSCOM	Scope: Multimedia Object Annotation LSCOM[56] (Large Scale Ontology for Multimedia) is a taxonomy developed specifically for annotating video content. It contains about 2000 concepts related to events, objects, locations, people, and programs. The LSCOM taxonomy is only available in English and can be downloaded as an XML file.
EBUCore	Scope: Description of Media Resources Initially started as extension of Dublin Core EBUCore[57] is a metadata scheme for the description of media resources

[54] http://dbpedia.org/

[55] http://www.freebase.com

[56] lscom.org

[57] http://tech.ebu.ch/lang/en/MetadataEbuCore

Type of Metadata	Description
EBU P/Meta	Scope: Description of AV content EBU P/Meta[58] is a library of XML-based descriptive elements and data types designed to reflect the intrinsic structure of AV content (essence) from programme groups down to shots or audio channels. It also offers the capacity to provide more detailed technical characteristics, including rights information. Several P/Meta constructs have been reused in EBUCore.
W3C Ontology for Media Resources	Scope: Description of Media Resources The W3C Ontology for Media Resources[59] is a meta schema for mapping different descriptions schemas to each other. This includes EBUCore, TV-Anytime, Quicktime, MPEG7, and many more.
CIDOC CRM	Scope: Cultural Heritage The CIDOC Conceptual Reference Model (CRM)[60] is a formal ontology intended to facilitate the integration, mediation and interchange of heterogeneous cultural heritage information.

5.5.1.4 User and Social Data

User and Social Data will be stored mainly in two forms: first as anonymized data gathered through behaviour tracking or automatic logging. These can be large amounts of statistical data for which a normal SQL Database is best suited. The other one is structured data of implicit or explicit user preferences. This kind of data will be stored using an RDF metadata scheme which uses the LinkedTV ontology for description. If structured data does not depend on ontologies, it might be stored in a normal SQL database as well.

5.5.1.5 Repository for Training Data for ASR and Object Recognition

Training data for Automatic Speech Recognition and Visual Object Recognition is a special case as it will not be part of the final LinkedTV platform but only provided for editors to include new AV materials for other languages into the repository.

5.5.1.6 Other Types of Storages

The following further repositories might also be part of the LinkedTV platform if required:

File Repository: a lot of data will be stored using flat files. This can apply to media resources as well as XML files such as EXMaralDA-Files. In later versions of the platform dedicated XML Repositories will be integrated if required.

[58] http://tech.ebu.ch/docs/tech/tech3295v2_2.pdf

[59] http://www.w3.org/TR/2012/REC-mediaont-10-20120209/

[60] http://cidoc.mediahost.org/standard_crm(en)(E1).xml

Also, training data for Automatic Speech Recognition and Visual Object Recognition (although not part of the final LinkedTV platform) will be stored as files or within the RDF Repository; also the Ookaboo RDF dump [61] for image search and classification which is a good resource to train new visual concept classifiers can be stored in the RDF repository.

SQL Database: A SQL Database will also be part of the platform, which can be used by LinkedTV applications to store application specific data.

NoSQL Database: As some application specific data might be more document oriented, a NoSQL Database such as MongoDB[62] or CouchDB[63] is considered to be part of the platform as well.

Content Management System: At present, there is no integration of a Content Management System required. If needed by the application scenarios a CMS will also be included.

5.5.1.7 Transactions

If data sets are stored in several storages on one server, e.g. the objects in the SQL-Database and relations in the triple store, then the Linked Media Layer provides a mechanism to ensure the transactional integrity.

5.5.2 Integration Layer

The Integration Layer provides the components and services for the integration of the different LinkedTV components and the communication between them. The main purpose is to ensure an open, extendable, distributed scalable integration backbone for LinkedTV in order to support different business models. Since the analysis process, data provision and applications could be located at different locations, it would be possible to flexibly integrate different content or service providers.

5.5.2.1 Linked Media Service Bus

The original media resources are accessed by the analysis process via URIs and stored as .mpg2 files on the server. The analysis results are stored on the platform and the enrichment process is initiated by an event. The Media Fragments and metadata gained through the enrichment process are stored in the Triple Store and SQL-DB, which can be accessed via a SPARQL endpoint but also by specific functions to directly access certain objects. The Applications and Tools access the platform via REST services.

An event handler is used for the major steps to indicate the progress of the workflow from analysis, enrichment, filtering to delivery to the related platform components. The following

[61] http://rdf.ookaboo.com/

[62] http://www.mongodb.org/

[63] http://couchdb.apache.org/

table lists the most important properties and services of the Linked Media Service Bus (LMSB):

Property / Service	Description
Event Driven Architecture	In order to enable subsequent or parallel processing of media resources the Linked Media Service Bus allows the definition of relevant events, such as SCENE_DETECTION_FINISHED. Relevant events are those events which should trigger another processing step by other components. Which events are relevant will be defined within the Deliverables specifying the components.
Service Interfaces	All components need to interact with the Linked Media Service Bus through common interfaces exposed as RESTful Web Services. These interfaces define a Linked Media standard protocol, which include response formats (e.g. JSON), standard parameter names and value types, naming conventions, namespaces, etc.
Message Communication	The LMSB allows the asynchronous communication between components through messaging, which normally is connected to Linked Media events. I.e. when a component has finished a task it can publish a message notifying the platform about that event, and all components which need to know about that event can subscribe to these notifications.
Openness	Due to the common service interface definition, new components can be easily integrated into the LinkedTV workflow. E.g. if a new component would be available for specific tasks, like a new audio stream analyzer for Greek, or a subtitle generator for Polish, or a video analysis tool specialised on the recognition of car brands and models, this could be easily integrated into the platform when it implements the common LinkedTV Service Interface.
Infrastructure Agnostic	For all components which are integrated via the LMSB the actual infrastructural and physical properties are irrelevant, i.e. it is not important to know on which server a particular component is actually residing, where a particular database is located, or which underlying operating system is installed. The actual LinkedTV platform can be highly distributed among several servers at all partners' locations, thus also forming a LinkedTV cloud. To provide a higher scalability, each server providing REST services could be replaced by a cluster of servers.
Feed Service	The Linked Media Service Bus supports a distributed publish - subscribe communication via the Internet for near-instant feeds of LOD change updates. This enables LinkedTV components to receive instant update notifications without permanently polling several LOD sources. LinkedTV will act as a subscriber to a feed server, which will notify a certain LinkedTV component about changes of LOD content referred by meta data in the repository. Perspectively, when LinkedTV itself offers content for other remote services, the Linked Media Service Bus will act as a feed server to notify repository updates to remote

Property / Service	Description
	subscribers. LinkedTV will select an appropriate feed service such as PubSubHubbub[64] and support open protocols such as XMPP[65], RSS, ATOM to communicate with the service.
Further Services	The LMSB will also provide further services needed by the presentation layer such as full text search or indexing for resources.

5.5.3 Service Layer

The Service part includes the general LinkedTV services which are needed by the different components within the other layers (Analysis and Annotation, User Interface and Applications, Personalisation and Contextualisation). Their main purpose is to provide CRUD[66] operations on the common Linked Media objects and their properties.

The following table lists the main Linked Media objects and example services:

Linked Media Object	Services
Media Resource	CRUD methods for getting collections of Media Resources and operations on properties of Base Media Resources, like title, format, duration, genre, etc. The properties will be a subset of those defined by the W3C Ontology for Media Resources[67] . In that respect this will include the implementation of a subset of the API as defined in the API for Media Resources 1.0 [68]. The Media Resource object contains the URI to the physical sources (Media Object).
Media Object	CRUD methods for actual physical sources, which can be files or streams. These can be also different versions of the same video with different resolutions or codecs
Media Fragment	Methods for retrieval, getting and manipulation collections of Media Fragments and accessing properties of Media Fragments as defined by the W3C Media Fragments Group[69] within the different dimensions of time, region, track and name.
Track	Specific CRUD methods for tracks.

[64] http://code.google.com/p/pubsubhubbub/

[65] http://xmpp.org/

[66] CRUD: Create, Read, Update, Delete. Although the CRUD methods will be realized through RESTful Services using the HTTP GET, POST, PUT, DELETE methods, please note that CRUD methods are not equivalent to REST methods.

[67] http://www.w3.org/TR/mediaont-10/

[68] http://www.w3.org/TR/2011/WD-mediaont-api-1.0-20111122/

[69] http://www.w3.org/2008/WebVideo/Fragments/WD-media-fragments-spec/

Linked Media Object	**Services**
TimedText	Specific CRUD methods for Timed Text tracks, e.g. for creating WebVTT tracks
Annotation	CRUD methods for annotations of media fragments
Hyperlink	Methods for retrieval, getting, manipulation and linking Media Resources with other Media Resources and content on the Web or LOD entities
Ontology Object	Methods for setting, getting and manipulation of information within the ontology, like getting all subconcepts of a given concept, or getting the instances, or getting properties/relations.

The Linked Media Layer which connects media fragments to named entities and other types of annotations will probably employ three main complementary indices within its RDF repository:

1. an index of complete media resources connecting to its media fragments
2. an index of media fragments connecting to its annotations, mainly named entities
3. a reverse index of named entities connecting to the media resources in which they occur.

The metadata will be stored in an RDF repository with a SPARQL endpoint or in a SQL database, if the data structure is known in advance with a SQL endpoint. So all metadata is retrievable through SPARQL resp. SQL queries. However, predefined RESTful API functions will also be included for the most common retrieval or update functions, like:

- get all fragments of a given base resource
- get all fragments annotated with a certain annotation (text or entity)
- add an annotation to a fragment
- get all related information (hyperlinks) of a certain Linked Media Object
- get next/previous fragment.

The following table lists a preliminary definition of REST URI patterns which will be provided by the LinkedTV Service Layer. The exact LinkedTV REST API will be defined within the next phase of the project. Request parameters and results are denoted using the JSON[70] format. Alternative response types such as XML, HTML, RDF or SRX[71] will be considered.

[70] JSON: JavaScript Object Notation, http://www.json.org/

[71] SRX: SPARQL Query Results, http://www.w3.org/TR/rdf-sparql-XMLres/

REST URI Patterns	Description
`/resource`	GET, PUT, DELETE requests on complete media resources, e.g. `GET /resource/{id}` returns the URI of a mediaresource `PUT /resource/{id}/event=”Title”` sets or updates the title of the media resource `GET /resource/{id}/fragments/` returns the IDs of all fragments of this mediaresource Through additional parameters the requested fragments can be further qualified
`/fragment`	GET, PUT, DELETE requests on complete media fragments, e.g. `GET /fragment/{id}` returns the URI of a media fragment `PUT /fragment/{id}/annotation={URI}` sets or updates an annotation for the media fragment with identifier “id” `GET /fragment/{id}/annotation/` returns all annotations of a fragment; these can be further qualified e.g. as in `GET /fragment/{id}/annotation?concept=person&country=france`

`/annotate`	PUT requests for media fragment annotations e.g. like
	`PUT /annotate?id={id}&annotation={annotation}`
	This is an alternative pattern to
	`PUT /fragment/{id}/annotation={annotation}`

`/search`	GET requests to the metadata repository; different types of queries can be qualified, e.g.
	`/search?type=sparql&expression={SPARQLExpression}`
	`/search?type=text&expression={BOOLEANTextExpression}`
	`/search?type=sql&expression={SQLExpression}`

`/event`	GET, PUT or POST requests for events, e.g. `POST /event/item={id}&event=AUDIO_ANALYSIS_STARTED` sets an event for the item with identifier “id” `GET /event/{id}` returns all events attached to the item with identifier “id”

5.5.4 Communication Infrastructure

The design employs a distributed architecture, which provides all cross-WP used APIs as REST interfaces, covering at least these services of:

- the **Linked Media Service Bus** to
  - access the repository
  - define and use ontologies, profiles and context
- the **Analysis and Annotation Layer** as well as the **Presentation Layer** interfaces to
  - the tools for the editor
  - the Hypervideo Applications and delivery of videos to the end user.

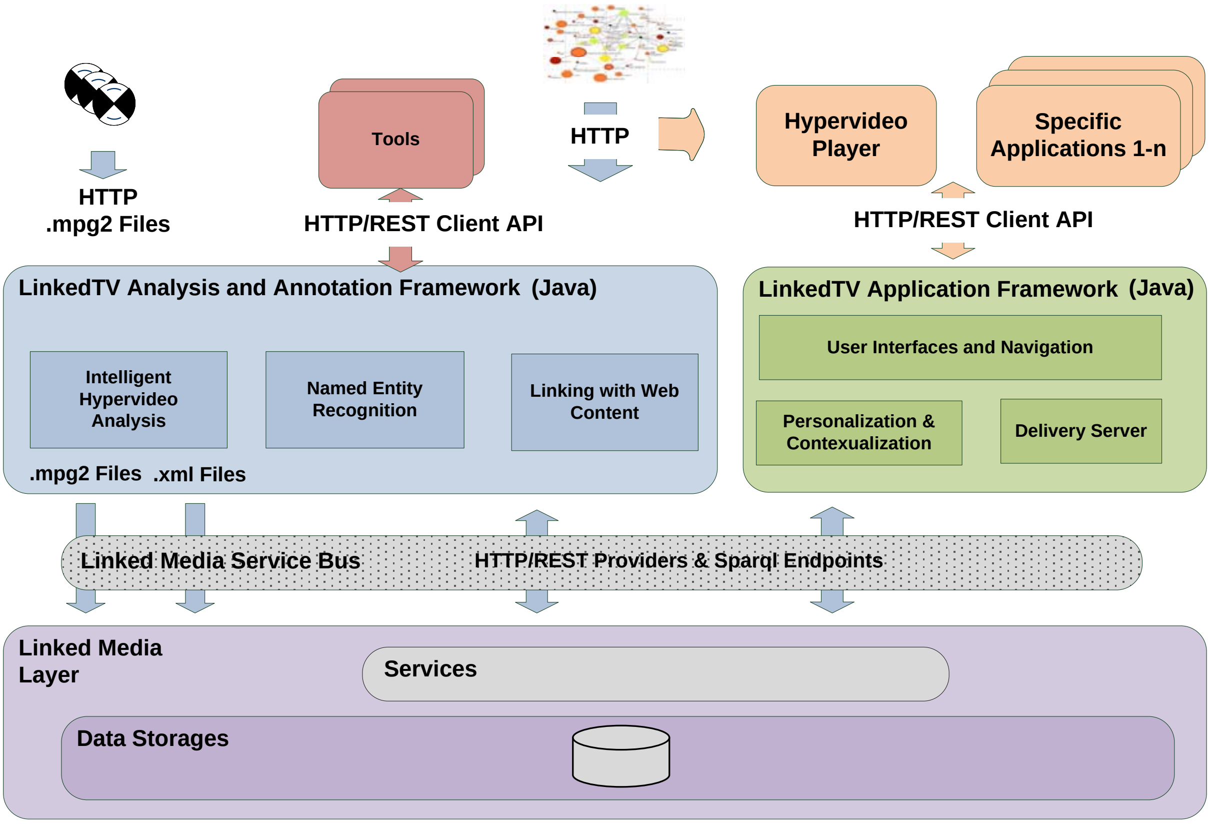
HTTP
.mpg2 Files
Tools
HTTP/REST Client API
HTTP
Hypervideo Player
Specific Applications 1-n
HTTP/REST Client API
LinkedTV Analysis and Annotation Framework (Java)
Intelligent Hypervideo Analysis
Named Entity Recognition
Linking with Web Content
.mpg2 Files .xml Files
LinkedTV Application Framework (Java)
User Interfaces and Navigation
Personalization & Contexualization
Delivery Server
Linked Media Service Bus
HTTP/REST Providers & Sparql Endpoints
Linked Media Layer
Services
Data Storages

Figure 14: Communication Infrastructure.

6 Development

6.1 Distributed architecture for the development process

The distributed architecture also allows for a flexible development of component by different partners developing their components at their site, which are always integrated in the whole platform by communicating with the remote components of the partners through file transfer and Rest services:

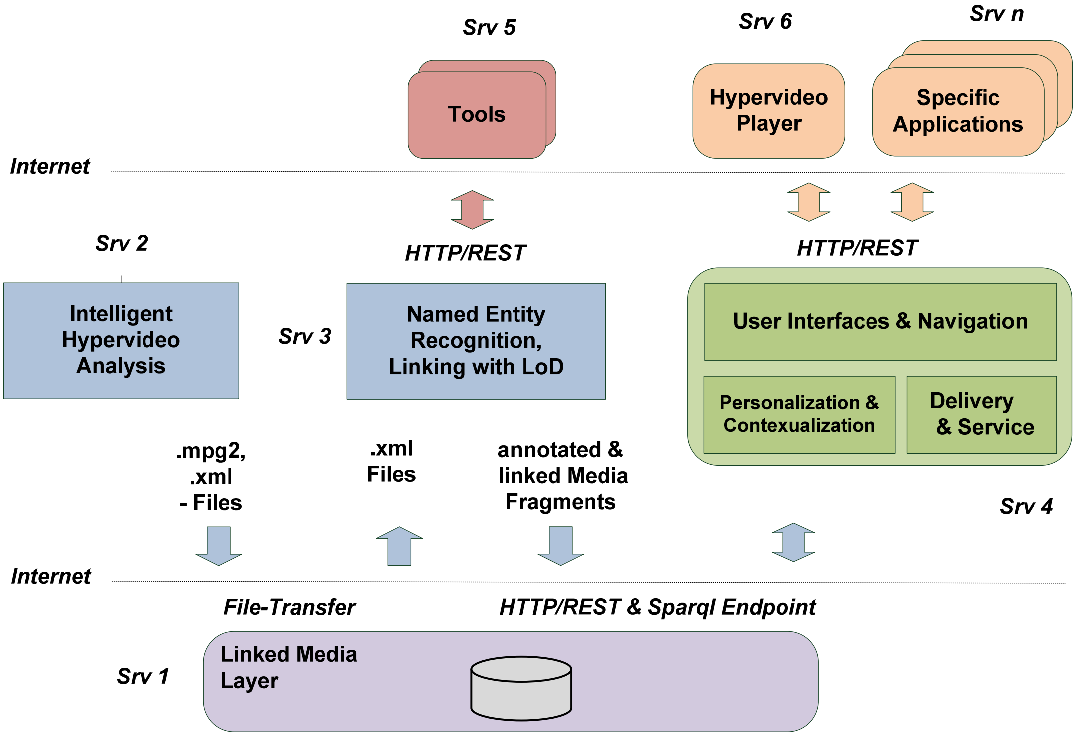
Srv 5
Tools
Srv 6
Hypervideo Player
Srv n
Specific Applications
Internet
HTTP/REST
HTTP/REST
Srv 2
Intelligent Hypervideo Analysis
Srv 3
Named Entity Recognition, Linking with LoD
User Interfaces & Navigation
Personalization & Contexualization
Delivery & Service
.mpg2, .xml - Files
.xml Files
annotated & linked Media Fragments
Srv 4
Internet
File-Transfer
HTTP/REST & Sparql Endpoint
Srv 1
Linked Media Layer

Figure 15: Distributed architecture for development process

The distributed development of WPs is enabled by the following communications:

- WP1 delivers .mpg2 and .xml-Files to the Repository
- WP2 reads the .xml -Files and writes the results via REST services in the Repository
- WP3 and 4 access the AV-materials and user data via REST services
- The Applications access the UI generation and materials via REST services of the related platform components.

6.2 Implementation Phases

According to the DoW the project plans the following milestones:

Month	Milestones
M06	Implementation Plan
M18	**Pilot 1:** Proof of concept Demonstrator for LinkedTV platform with HTML5 clients connected via HTTP
M24	Evaluation and Demonstration for Pilot 1
M36	**Pilot 2:** Refined demonstrator for LinkedTV platform with - HTML5 clients connected via HTTP and - additional HbbTV clients connected via HbbTV and HTTP
M42	Final demonstrator

WP5 has designed an architecture, which allows to develop and extend the platform stepwise according to this project plan. It also allows to flexibly integrate different client variants with their distribution channels in subsequent stages.

6.3 Used Tools, Frameworks

The evaluation and selection of tools and frameworks has begun and will be finished with the specification of the WPs. The following candidates are already under assessment:

Runtime

- Framework: Spring Beans, Eclipse
- Triple Store: OpenLink Virtuoso, Sesame
- REST Framework: Jersey
- DB: MySQL, PostgreSQL
- No SQL: MongoDB, CouchDB
- XML-DB: Xindice
- Application Server: JBoss with the ESB (JBoss Enterprise Service Bus) or Apache ActiveMQ

Development

- Development tools: Eclipse
- Version Control System: SVN
- Configuration tools: Maven

6.4 Standards

The LinkedTV Project intends to adapt the following standards. For some of them there is already a concrete employment and specification in the related sections; some others are not yet needed, but could become relevant for later project phases.

Domain	Standards
Platform	HTTP REST JSon with Java/JavaScript Ajax Binding URI, URN HbbTV
TV Program Data	DVB metadata model EBU metadata model ARD BMF BBC Programmes ontology TV anytime EU Screen IPTC W3C ontology for Media Resources Metadata Working Group's Guidelines COMM
Meta Data	RDF OWL Media Fragments URI 1.0 WebSRT DC (Dublin Core) CIDOC/CRM EDM (Europeana Data Model) FOAF (Friend of a Friend) SIOC Mpeg7, Mpeg47 Schema.org
UI	HTML, HTML5, CE-HTML, XHTML Ajax, YAML SMIL
Content	XML, XSLT, XACML UTF8
Sparql Endpoint	Joseki, Open Search
AV-Format	Mpeg4, Mpeg2

6.5 Deployment

The architecture allows the deployment of the platform server side components on different OS. Due to the experience of the partners and cost efficiency it is intended to deploy the server side components on Linux.

However, the distributed architecture also allows to integrate servers with other OPs, if components (e.g. from third parties) have to be integrated which only run on a certain OP (e.g. Windows).

7 References

1. Minelli, S. (2007). Gathering requirements for multilingual search of audiovisual material in cultural heritage. Proc. of Workshop on User Centricity – state of the art (16th IST Mobile and Wireless Communications Summit), Budapest, Hungary.
2. "Television and the Future Internet: the NoTube project", Lora Aroyo, Lyndon Nixon, Stefan Dietze and NoTube consortium at the Future Internet Symposium (FIS 2009), Sept 2009
3. Köhler, Joachim (Hrsg.); Larson, Martha (Hrsg.); Jong, Franciska de Jong (Hrsg.); Kraaij, Wessel (Hrsg.); Ordelman, Roeland (Hrsg.); Association for Computing Machinery / Special Interest Group on Information Retrieval: Proceedings of the ACM SIGIR Workshop, 24 July 2008, Singapore
4. Oomen, J. and V. Tzouvaras (2007), Providing Access to European Television Heritage. In Ariadne. Issue 53 http://www.ariadne.ac.uk/issue53/ooman-tzouvaras/
5. Bates, M. et al. (1993). An Analysis of Search Terminology Used by Humanities Scholars: The Getty Online Searching Project Report Number 1. The Library Quarterly 63(1): 1-39.
6. Dalton, M. S. and Charnigo, L. (2004). Historians and their information sources. College & research libraries 65:400-425.
7. Wright, Richard (2008) Preservation of Digital Audiovisual Content: briefing paper. Accesed online at: www.digitalpreservationeurope.eu/publications/briefs/audiovisual_v3.pdf
8. K-Space NoE, http://www.k-space.eu/
9. Boemie, http://www.boemie.org/
10. N. Sebe, M. S. Lew, X. Zhou, T. S. Huang, E. M. Bakker, "The state of the art in image and video retrieval", in Proceedings of the 2nd International Conference on Image and Video Retrieval, Urbana, July 2003
11. C.G.M. Snoek, M. Worring, “Multimodal Video Indexing, a Review of the State-of-the-art,” Multimedia Tools and Applications, Springer Netherlands, pp. 5-35, January 2005.
12. Shih-Fu Chang, Manmatha, R., Tat-Seng Chua "Combining text and audio-visual features in video indexing" in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05), March 2005, Philadelphia, USA
13. Jane Hunter. Adding Multimedia to the Semantic Web - Building an MPEG-7 Ontology. In 1st International Semantic Web Working Symposium (ISWC), pages 261–281, 2001.
14. Chrisa Tsinaraki, Panagiotis Polydoros, and Stavros Christodoulakis. Interoperability support for Ontology-based Video Retrieval Applications. In 3rd International Conference on Image and Video Retrieval (CIVR), pages 582–591, 2004.
15. Richard Arndt, Raphaël Troncy, Steffen Staab, Lynda Hardman, and Miroslav Vacura. COMM: Designing a Well-Founded Multimedia Ontology for the Web. In 6th International Semantic Web Conference (ISWC), 2007.
16. Raphaël Troncy, Óscar Celma, Suzanne Little, Roberto García and Chrisa Tsinaraki. MPEG-7 based Multimedia Ontologies: Interoperability Support or Interoperability Issue?. In Workshop on Multimedia Annotation and Retrieval enabled by Shared Ontologies (MAReSO'07), Genova, Italy, December 5, 2007.
17. N Maillot, M. Thonnat and A. Boucher. Towards ontology-based cognitive vision. Machine Vision and Applications MVA, 16(1),pp33-40, Springer-Verlag Heidelberg, Dec 2004
18. http://www.metadataworkinggroup.com/
19. http://www.w3.org/2008/WebVideo/Annotations/
20. Z. Li, H. Davis, and W. Hall. Hypermedia Links and Information Retrieval. 14th Information Retrieval Colloqium, Lancaster University, http://eprints.ecs.soton.ac.uk/772/, 1992
21. L. Hardman, J. van Ossenbruggen, K. S. Mullender, L. Rutledge, and D. C. A. Bulterman. Do you have the time? Composition and linking in time-based hypermedia. In

HYPERTEXT’99: 10th ACM Conference on Hypertext and Hypermedia, pages 189–196, New York, NY, USA, 1999.

22. Synchronized Multimedia Integration Language (SMIL 3.0). W3C Recommendation, 01 December 2008, http://www.w3.org/TR/SMIL/
23. S. Pfeiffer and C. Parker. Specifying time intervals in URI queries and fragments of time-based Web resources, IETF Draft, http://www.annodex.net/TR/URI_fragments.html
24. http://youtube.com/watch?v=UxnopxbOdic
25. Sieg A.,Mobasher B., Burke R. (2007). Ontological User Profiles as the Context Model in Web Search. In Proceedings of the 2007 DePaul CTI Research Symposium, Chicago, IL, May 2007.
26. Sieg A.,Mobasher B., Burke R. (2007) Learning Ontology-Based User Profiles: A Semantic Approach to Personalized Web Search. In IEEE Intelligent Informatics Bulletin, vol. 8, NO. 1, November 2007
27. G. Th. Papadopoulos, V. Mezaris, I. Kompatsiaris, M. G. Strintzis, "Combining Multimodal and Temporal Contextual Information for Semantic Video Analysis", Proc. IEEE International Conference on Image Processing (ICIP 2009), Cairo, Egypt, November 2009, pp. 4325-4328.
28. V. Mezaris, S. Gidaros, G. Th. Papadopoulos, W. Kasper, J. Steffen, R. Ordelman, M. Huijbregts, F. de Jong, I. Kompatsiaris, M. G. Strintzis, "A system for the semantic multi-modal analysis of news audio-visual content", EURASIP Journal on Advances in Signal Processing, vol. 2010, 2010.
29. C. Bizer, T. Heath, T. Berners-Lee, "Linked Data: Principles and State of the Art", 17th International World Wide Web Conference, W3C Track @ WWW2008, Beijing, April 2008
30. Rüdiger Klein, Erich Rome, Césaire Beyel, Ralf Linnemann, Wolf Reinhardt, Andrij Usov: Information Modelling and Simulation in Large Interdependent Critical Infrastructures in IRRIIS. CRITIS 2008:36-47
31. Wagner, M., Kellerer, W. 2004. Web services selection for distributed composition of multimedia content, Proceedings of the 12th annual ACM international conference on Multimedia, Oct 2004, New York, NY, USA.
32. D. Helic, N. Scherbako - Introduction to Software Architecture – 2008
33. Rolf Fricke, Dominik Zimmermann, Karsten Trint, "The EPG Recommender – Semantic Technologies in the cross-media Production Process", 24. Annual Conference of the TV-/ Media Technical Association (FKTG), 2010 Hamburg
34. Wikipedia article on REST, http://en.wikipedia.org/wiki/Representational_State_Transfer
35. Scherp, R. Jain, M. Kankanhalli, V. Mezaris, "Modeling, Detecting, and Processing Events in Multimedia", Proc. ACM Multimedia 2010, Firenze, Italy, October 2010, pp. 1739-1740.
36. D. Tsatsou, F. Menemenis, I. Kompatsiaris and P. C. Davis. "A Semantic Framework for Personalized Ad Recommendation based on Advanced Textual Analysis", 3rd ACM Conference on Recommender Systems (RecSys'09), pp.217-220, NY, USA, Oct 2009
37. S. Stefanov, Svetlin, V. Huang. "A Semantic Web Based System for Context Metadata Management", in Metadata and Semantic Research, Communications in Computer and Information Science, Volume 46. ISBN 978-3-642-04589-9. Springer Berlin Heidelberg, pp. 118-129, Jan 2009
38. Playing MUSIC — building context-aware and self-adaptive mobile applications, Jacqueline Floch, Rolf Fricke et al, onlinelibrary.wiley.com/doi/10.1002/spe.2116/abstract John Wiley & Sons, Inc., 2012
39. http://dbpedia.org
40. OWL-S Coalition: OWL-S 1.1 release. (2004). http://www.daml.org/services/owls/1.1/