Deliverable 5.7 Validation of the LinkedTV Architecture

Heike Horstmann (Fraunhofer IAIS),
Jan Thomsen (Condat)

31/03/2015

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 2015*
Actual date of delivery	*31st March 2015*
Deliverable number	*D5.7*
Deliverable name	*Validation of the LinkedTV Architecture*
File	*LinkedTV_D5.7.docx*
Nature	*Report*
Status & version	*Final, V1.0*
Number of pages	*33*
WP contributing to the deliverable	*WP 5*
Task responsible	*Fraunhofer IAIS*
Other contributors	*Condat*
Author(s)	*Heike Horstmann* *Fraunhofer IAIS* *Jan Thomsen* *Condat*
Reviewer	*Evlampios Apostolidis, CERTH*
EC Project Officer	*Thomas Küpper*
Keywords	*Software Architecture, Validation*
Abstract (for dissemination)	*The LinkedTV architecture lays the foundation for the LinkedTV system. It consists of the integrating platform for the end-to-end functionality, the backend components and the supporting client components. Since the architecture of a*

[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))

	software system has a fundamental impact on quality attributes, it is important to evaluate its design. The document at hand reports on the validation of the LinkedTV architecture.

Table of contents

1 Introduction

Software architecture, as defined by [IEEE1471], is the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution. In practice this means, that the architecture describes the system structure and serves as a common communication means. The architecture is a design artefact that addresses the functional and quality goals of the system in terms of interoperability, availability, security, and so on. Choosing and designing an architecture that satisfies the functional as well as the non-functional (or quality attribute) requirements are vital to the success of a system.

The LinkedTV architecture lays the foundation for combining distinct functional components into an integrated system for audio-visual content enrichment. It consists of, as defined in [D5.1], the LinkedTV core platform for the end-to-end functionality, the backend components (e.g. services) and the supporting client components (e.g. player).

The document at hand reports on the validation[2] of the LinkedTV architecture. Since the architecture of a software system has a fundamental impact on quality attributes, it is important to evaluate its design. The objectives of a software architecture validation or review are to assess an architecture's ability to deliver a system capable of fulfilling the (quality) requirements and to identify potential risks [Cle02].

The outputs depend on the stage at which the evaluation is performed. The more architectural decisions have been made, the more precise the evaluation can be. On the other hand, the more decisions have been made, the more difficult it is to change any one of them. [Cle02] sees three different stages of architecture design when an architecture evaluation should be done. The classical application of architecture evaluation occurs when the architecture has been specified but before implementation has begun. There are two useful variations from the classical: early and late. The first variation (early) does not wait until the architecture is fully specified, whereas the second variation (late) takes place when not only the architecture is fixed but the implementation is completed as well. Although the outcome of a late evaluation cannot be used to make conceptional changes, it helps to understand if the evaluated architecture meets its requirements.

The validation of the LinkedTV architecture is performed at the latest stage of architecture design and implementation. The implementation of the LinkedTV platform and components is finished besides of minor improvements on component level. This implies that any findings from this validation will not result in recommendations for improvements, since they cannot be implemented within the project duration. Nevertheless, the findings are documented and

[2] Please note: In this document the terms validation, evaluation, review and assessment are used interchangeably.

can be used either as lessons learned or as starting point for refactoring the LinkedTV system.

In software engineering literature, several different architecture evaluation techniques exist. They can be categorised broadly as either questioning techniques or measuring techniques. Questioning techniques generate qualitative questions to be asked on an architecture and they can be applied for any given quality. Measuring techniques suggest quantitative measurements to be made on an architecture. They are used to answer specific questions on specific software qualities and rely on metrics, simulations and prototypes. Qualitative approaches can again be divided into experience- and scenario-based reviews.

The LinkedTV architecture validation documented in this report has followed a qualitative approach. More information on the objectives, scope and the approach of the LinkedTV architecture validation are given in the following Chapter 2. Chapter 3 presents the evaluation criteria of the validation and Chapter 4 defines a few scenarios that were derived from several project-internal discussions on possible business models of the LinkedTV system. Chapter 5 provides a summary of the architecture documentation compiled from deliverables created by WP5 during the duration of the project. This Chapter also gives some insights on the influence of the architectural decisions that have been made on the quality attributes of the resulting LinkedTV system. The findings of the validation and lessons learned are documented in Chapter 6. Finally, the report closes with a short summary.

1.1 History

Table 1: History of the document

Date	Version	Name	Comment
2015/02/24	V0.1	Horstmann, Fraunhofer	Initial version with structure
2015/03/12	V0.2	Horstmann, Fraunhofer	List of architecture specific requirements in first version added
2015/03/16	V0.3	Horstmann, Fraunhofer	First input on architecture concerns and concepts
2015/03/17	V0.35	Horstmann, Fraunhofer	Restructuring of chapter 3, First findings added, introduction included
2015/03/19	V0.4	Horstmann, Fraunhofer	Finalising chapter 1, input for chapter 2 and 3
2015/03/20	V0.5	Thomsen, Condat	Added input to chapter 5 and 6
2015/03/21	V0.55	Horstmann, Fraunhofer	Finalising chapter 3 and 5, further input to chapter 2
2015/03/22	V0.6	Horstmann, Fraunhofer	Finalising remaining chapters
2015/03/23	V0.7	Horstmann, Fraunhofer; Thomsen, Condat	Finalising for QA
2015/03/24	V0.8	Horstmann, Fraunhofer	Addressing QA comments, finalising for approval by Lyndon
2015/03/31	V0.9	Horstmann, Fraunhofer Thomsen, Condat	Addressing comments from Lyndon Nixon, additional section included
2015/03/31	V0.95	Horstmann, Fraunhofer	Finalisation
2015/03/31	V1.0	Merzbach, Fraunhofer	Last layout change

1.2 List of abbreviations

Abbreviation	Explanation
API	Application Programming Interface
CMS	Content Management System
FOAF	Friend of a Friend
JSON	JavaScript Object Notation
LMMS	Linked Media Management Systems
LSCOM	Large-Scale Concept Ontology for Multimedia
MAM	Media Asset Management System
OPMV	Open Provenance Model Vocabulary
RDF	Resource Description Framework
REST	Representational State Transfer
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SRT	SubRip subtitle format
STL	Spruce Technologies subtitle format
TTL	Terse RDF Triple Language

2 Objectives, scope and methodology

2.1 Objectives

The validation of the LinkedTV architecture is triggered by the idea to have a final evaluation of all LinkedTV outcomes. With the architecture laying the foundation of the LinkedTV system it is important to assure that it is capable to support the current and (to some extent) also future business objectives of the LinkedTV system. The validation is performed at the end of the project being the right time to reflect the design process and document any lessons that were learned from it. The objectives of this validation can be summarised by the following two questions:

- Is the architecture a suitable basis for fulfilling the current (and future) requirements?
- Are there design decisions that should be revised to improve the architecture?

2.2 Scope

The scope represents the subject of the evaluation. It describes the boundaries of a specific instance of a software architecture evaluation. The LinkedTV architecture validation involved the architecture of the complete LinkedTV system but not individual specific components like services or data repositories. It determined all non-functional (quality attribute) requirements influenced by the architecture, being a subset of the originally (in [D5.1]) defined functional and non-functional requirements.

2.3 Methodology

Before the validation was conducted, the most prominent of the current approaches for qualitative software architecture reviews were studied[3] and evaluated in relation to their applicability for the LinkedTV architecture validation. Although the (research) literature offers an extensive variety of approaches, only a few of them (if any at all) are really used in industry practice [Bab09]. We found that most of the evaluated approaches were either very formal and abstract or complicated and resource consuming. None of the studied approaches really fitted to the given objectives right “out-of-the-box”. Applying a method such as [SAAM96] or [ATAM98] without prior experiences in this field, leaves many unanswered questions about abstract concepts like attribute-based architectural styles or utility trees. They often demand the involvement of several stakeholders and propose an evaluation workflow requiring at least a 3-day meeting with all of them. During the final period of the project with time consuming user trials needing extensive preparation and final implementation tasks, the evaluation of the architecture could not be executed with such a great effort. Even lightweight approaches, such as the Tiny Architectural Review Approach

[3] We considered [SARA02], [SAAM96], [ATAM98], [TARA11] and others.

[TARA11], did not suit our needs. Although it is defined to provide a simple approach to perform a basic architectural review, it applies a metrics-based code analysis, which makes no sense for an architecture that was implemented by a variety of services, developed by different partners. Some of the services and especially the LinkedTV core platform forming the basis of the LinkedTV system are not open-source components and therefore are not available for code analysis by other partners.

We therefore relied on expert judgement and no formal approach. From the existing architecture documentation, we compiled the evaluation criteria and the employed architectural concepts. Supported by existing general evaluations of service-oriented architectures we analysed the architecture based on our experiences in software architecture design and implementation.

3 Evaluation criteria

Criteria for the validation of the LinkedTV architecture are all architecture specific requirements, a subset of the functional and non-functional requirements defined in [D5.1] at the beginning of the project. The requirements were compiled from:

- General requirements defined by the business goals of the envisioned system
- Specific requirements resulting from the use case scenarios
- Specific requirements coming from the technical work packages (WP1-4)

In total, 121 requirements, most of them functional ones, were defined in [D5.1]. 42 requirements were identified to be specific to the architecture, with 34 being functional ones. During the validation process, we differentiated between the functional and the non-functional requirements. Functional requirements with relevance to the architecture were checked according to their fulfilment. Although this group is the majority of requirements, they were not in the focus of the validation. The implementation of all functional requirements was checked. All functional requirements that have been stated as must-haves are fulfilled in the final LinkedTV system. Detailed inspections were not possible, since the specified requirements did not provide the needed information for a deeper inspection.

Therefore, the validation concentrated on the non-functional (quality attribute) requirements, that are presented in the following table.

Origin	Title / Description
REQ-GEN-006	**Scalability** 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-scale scenarios without building up exhausting investments in CPU, RAM, disk and network capacities. However, the employed architecture has to allow **scaling** 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 support load balancing.
REQ-GEN-006	**Performance** 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.

Origin	Title / Description
REQ-GEN-002 REQ-GEN-008	**Interoperability** LinkedTV SHOULD offer an open platform, which is agnostic concerning specific server technologies and supports different delivery channels and client types. 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, MAM etc. products should be conceptually considered and be provided if needed.
REQ-GEN-004	**Extensibility** 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.
REQ-GEN-001	**Conformance to relevant standards** 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 Section 6.4 of [D5.1].
REQ-GEN-003	**Physical distributed allocation of components** The platform MUST be designed in a way that all functionalities can be accessed over a service-oriented architecture (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.
REQ-GEN-012	**Distributed development** 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 the 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.

4 Architecture scenarios

The descriptions of the quality attribute requirements as extracted from [D5.1] and presented in Chapter 3 are still vague. Therefore, they had to be further specified and prioritised prior to the architecture validation. This has been achieved based on scenarios that were derived from several project-internal discussions on possible business models of the LinkedTV system. [ATAM98] as the most prominent approach for scenario-based architecture validation, distinguishes between three types of scenarios: use case scenarios, growth scenarios and exploratory scenarios. These different types of scenarios are used to test a system from different angles.

As introduced in [D8.7], LinkedTV offers a solution for content owners enabling them to create a new generation of (Web)TV applications interweaving TV and Web content for the benefit of their audience. However, this business case is only the most likely one. Newspaper publishers or news agencies wanting to create compelling new service offers for end consumers could also benefit from using LinkedTV. Furthermore, E-Learning and video lectures could profit from LinkedTV. Even the advertising market could be interested in using LinkedTV technologies. Thereby it becomes clear that the main business goal for the LinkedTV system and the architecture is to serve as many business cases as possible. Extensibility, scalability and interoperability are therefore the most important quality attributes. To further specify them, we defined prior to the architecture validation three exemplary scenarios. Seeing the longitudinal user trial of RBB as one realistic use case scenario, we added a fourth scenario to specify the current performance requirements.

4.1 Scenario 1: Add new services to fulfil the needs of a customer

A customer will want to connect its legacy systems to the LinkedTV system, either to directly export content to or import metadata from it. Besides, a customer might have the need for special analysis services. For example, a Swedish broadcaster would need to have an automatic speech recognition service capable of Swedish.

This scenario further specifies the interoperability and the extensibility requirement.

4.2 Scenario 2: Create a LinkedTV application for another TV format

Using the LinkedTV system to create annotations and enrichments for another TV format, such as an edutainment show, will result in many adaptations and modifications of the system. Foremost, a new companion application will have to be designed and implemented based on the Multiscreen Toolkit (see [D5.5] for more information). Enrichment services will have to be adapted to the specific requirements of the client application. The same applies to the Editor Tool (see [D1.5] for more information]).

This scenario further specifies the extensibility requirement. Modifications of services and tools is not an architectural requirement.

4.3 Scenario 3: Use LinkedTV to automatic annotate and enrich a large video archive

Processing the content of a large video archive requires the ability to handle large amounts of video data. The architecture has to be able to deal with several processing tasks in parallel, and data repositories have to scale to the higher data amount. Depending on the customer, the created annotations and enrichments would have to be exported to the legacy system of the customer, resulting in the need for an export service and the ability to delete exported data from the local repositories.

This scenario further specifies the scalability and, to some extent, the extensibility requirement.

4.4 Scenario 4: Produce a companion application for a daily news show

This scenario is based on the longitudinal user trial, which took place at partner RBB during March 2015. On five consecutive days, the RBB team used the LinkedTV system in order to produce video annotations and enrichments and present them via their companion application on second screen devices. An editor used the Editor Tool to curate the automatically created information. Test users watched the daily news show via the companion application. For this user trial, the daily news show had to be processed by the system in a timely manner.

This scenario further specifies the performance requirement.

5 Employed architectural concepts

To validate a software architecture against given evaluation criteria, it is crucial to understand the employed architectural concepts and underlying decisions. While the LinkedTV architecture is documented in several deliverables created by WP5, the underlying decisions are hard to comprehend because they were made some time ago and not implicitly documented. Therefore, the validation focused on the architecture documentation. The main concepts are summarised in Section 5.2. They were extracted from:

- [D5.1], Sections 3, 4 and 5
- [D5.4], Sections 2 and 3.1
- [D5.6]
- Update on LinkedTV Platform Architecture (see following Section)

During the preparation of the validation, it became obvious that even in the final six month of the project, additional updates to the architecture based on experiences from LinkedTV application development and user trials have been made. They are documented in the following Section.

5.1 Update on LinkedTV Architecture

In particular, we now distinguish between three workflows:

1. The production workflow
2. The publishing workflow
3. The personalization and consumption workflow

Figure 1: Updated LinkedTV workflow

5.1.1 The production workflow

The objective of the production workflow is to make the videos “LinkedTV-ready” and consists of the steps:

1) **ingestion** of the video itself and related metadata such as TVAnytime metadata or subtitle files;
2) **analysis** of the video and audio tracks with all the various techniques as performed by WP1;
3) **serialization** of the results and metadata files into the common LinkedTV data model, which is an RDF-based description format making use of all kinds of existing ontologies such as the W3C Media Ontology and which provides annotated media fragments;

4) **annotation**;
5) **named entity recognition,** which will already provide data about basic entities in the videos.

The outcome of the production workflow is RDF data, which is stored in the Virtuoso Triple Store within the LinkedTV core platform. For a 30 min *rbb AKTUELL* news show, approximately 50.000 triples are generated, with about 800 media fragments and about 3.000 annotations. This data is made accessible through the Data/Video Layer API (see below).

5.1.2 The publishing workflow

The objective of the publishing workflow is to take the "raw" LinkedTV production data, evaluate it, correct it, filter out unwanted data, and most notably, enrich it by adding all kinds of related material to the various chapters or entities by making use of the rich set of the LinkedTV enrichment services as described in D2.6. Within the first version of the workflow, enrichment was seen as part of the production workflow, but in the last phase of the project it turned out that this is an own workflow. Concerning the data produced, only part of the data is stored back in the platform repository, i.e. mainly that part which concerns data about the structure of the video, such as chapter titles, and start and end points of chapters. The publishing workflow is managed through the LinkedTV Editor Tool and ends with actually publishing the video.

5.1.3 The personalisation and consumption workflow

While both the production and the publishing workflow contribute to the LinkedTV platform, the personalisation and consumption (or viewing) workflow uses the LinkedTV platform. The personalisation and consumption workflow is the process of:

- playing the video itself to a viewer,
- displaying the related content either on the same screen or a second screen depending on the respective scenario,
- adapting the related information to the viewer's profile,
- reacting to viewer events like pause, fast forward or switch channel,
- building the user profile out of his or her implicit or explicit preferences.

5.1.4 LinkedTV architecture layers

The final LinkedTV architecture that supports this workflow consists of three main sub-layers: 1) the Data/Video Layer, 2) the Integration Layer, and 3) the Service Layer.

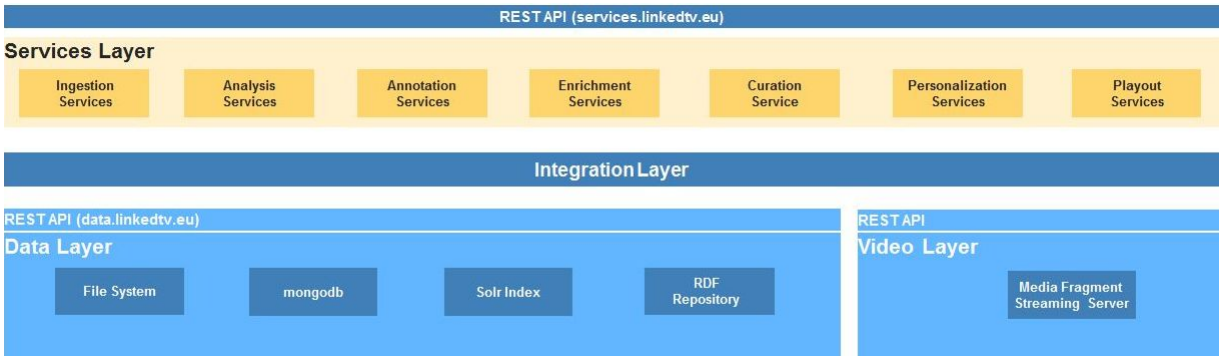

Figure 2: Layers of the LinkedTV Platform

The Data/Video Layer includes all persistent data, including metadata as well as the video resources themselves. Conceptually, the Data/Video Layer is a layer of Web resources and not necessarily limited to the LinkedTV platform itself. The Data/Video Layer in LinkedTV provides a unified REST API under data.linkedtv.eu to all generated media fragments and their annotations.

The Integration Layer, however, is a unique part of the LinkedTV platform, and provides the workflow orchestration which connects the different services and ensures the persistent and consistent storage of all data generated and aggregated throughout the LinkedTV workflow.

The Services Layer comprises all the different specific LinkedTV services for ingestion, analysis, serialization, annotation, enrichment, curation and personalisation that create the richness of the LinkedTV system. These services are distributed over the Web among all partners and can be accessed and combined through genuine REST API interfaces. The LinkedTV platform also provides a unified access API under services.linkedtv.eu.

5.2 LinkedTV architectural concepts

The basic design principles for the architecture were already introduced in Section 5.1 of [D5.1]. There, the decision to employ a service-oriented architecture (SOA) and the use of REST Services were introduced. SOA and RESTful Web Services are the main architectural concepts of the LinkedTV architecture. It is realised as a distributed system, with different services hosted at the local facilities of different partners.

In Section 5.2 of [D5.1], three different layers of the architecture and most importantly a distinct data layer were presented. Although the roles of the other layers have changed during the implementation of the system, the data layer and its idea to provide access to the persistency components, such as storage for video content, annotations, ontologies and user data stayed the same. In the final architecture (see Figure 2), this layer is renamed to Data/Video Layer to stress the fact that not only data but also videos (media fragments) are stored in this layer. The Data/Video Layer in LinkedTV provides a unified REST API under data.linkedtv.eu to all generated media fragments and their annotations. On top of the Data/Video Layer, the Integration Layer takes care of the orchestration of the LinkedTV workflow. The Services Layer contains all the distributed services provided and hosted by the different partners. It defines a dedicated interface to these services by an additional REST

API and several general service classes that can be used to easily extend the catalogue of available service components. The final architecture includes the following service components:

Importers (Ingestion Services)

The purpose of the importers is to import media resources and related files into the LinkedTV platform. This might be either physically by downloading a copy of the media resource and related files or just virtually by storing links. Internally all resources are stored as URIs. Importers do also create a unique identifier, a UUID, for each media resource. There are different predefined types of Importers, such as *FileSystemImporters*, *FTPImporters* or *HTTPImporters*. Additional importers can be implemented as respective subclasses when needed. For importing a concrete TV programme, a related subclass has to be implemented. For instance, for importing *rbb Aktuell* news show episodes, there is a component *RbbAktuellImporter* derived from *FileSystemImporter*, which defines all concrete parameters, such as directories, which kind of metadata files to import, where to get basic information like title, duration, etc.

Analysers

Analysers trigger the analysis of the different tracks of a media resource, mainly the video track, the audio track and the subtitle track. Consequently, the LinkedTV platform comes with the predefined Analysers *VideoAnalyzer* and *AudioAnalyzer*. Subtitle tracks usually are sent directly to the processing service, the SubtitleAnnotator. Within LinkedTV, analysis is performed by the tv2exb service provided by partner CERTH, hence subclasses *TV2EXBVideoAnalyzer* and *TV2EXBAudioAnalyzer* have been defined which incorporate all basic tv2exb properties such as the base URL for the REST service and available service parameters. Again, for analysing concrete episodes of a TV programme a final component has to be implemented, in the case of *rbb Aktuell*, these are the *TV2EXBRbbAktuellVideoAnalyzer* and the *TV2EXBRbbAktuellAudioAnalyzer*, which configure parameters like language and the concrete analysis algorithms to be processed. The result of the services are XML files in the Exmaralda format.

Annotators

Annotators serialise the analysis results or associated metadata files. For each piece of information they generate a media fragment URI according to the W3C Media Fragment URI standard and an annotation according to the W3C Open Annotation standard. LinkedTV comes with three predefined annotators: a TVAnytimeAnnotator, a SRTAnnotator for subtitle files in SRT format and an ExmaraldaAnnotator for serializing and annotating analysis result files. The annotation service for LinkedTV is performed by the tv2rdf service provided by partner EURECOM, hence three subclasses *TV2RDFTVAnytimeAnnotator*, *TV2RDFSRTAnnotator* and *TV2RDFExmaraldaAnnotator* have been implemented. Again, for processing files of concrete TV programme episodes the respective components had to be again derived, such as the *TV2RDFRbbAktuellTVAnytimeAnnotator* or the

TV2RDFTkkExmaraldaAnnotator. Among other things, they configure in which language entities should be detected. The result of a TV2RDFAnnotator is a RDF TTL file that can be stored in a Triple Store.

Enrichers

Enrichers take the media fragment annotations, in particular entities, and collect related media on the Web in various different ways. This can be either normal web crawlers, which look for related content from a white list, or enrichers, which connect to specific Linked Data endpoints, e.g. Europeana Space. As enrichment of media resources are usually not generated as part of the automatic workflow, but either as part of the curation process or even on-demand at playout time, enrichers are not triggered by the automatic processing workflow but from the client. The LinkedTV Platform comes with a general Enricher component from which the *TVEnrichEnricher* is derived, which implements a call to the tv2enrich service by partner EURECOM. As above, specific enrichers have to be implemented for specific programmes, in LinkedTV e.g. the *TVEnrichRbbEnricher*, which collects related material from a specific white list defined by partner RBB.

Indexers

For supporting search, the platform also contains indexers as a component. In the same way as above an abstract component, *Indexer* has been implemented. In LinkedTV indexing and search is based on Solr and there is an according service *TV2LuceneIndexer* and the respective indexers *TV2LuceneRbbIndexer*, *TV2LuceneSVIndexer* and *TV2LuceneUMONSIndexer*. Of course, other indexers, such as an *ElasticSearchIndexer* based on the ElasticSearch engine could also be implemented.

Controllers

The *LinkedTVController* component is the basic component for managing the workflow chain. The LinkedTVController defines the orchestration of the different workflow steps. For processing specific programmes, the specific components have to be derived first and then the according controller has to be implemented. Taking the example of *rbb Aktuell*, the *LinkedTVRbbAktuellController* manages the specific components *RbbAktuellImporter*, *TV2EXBRbbAktuellVideoAnalyzer*, *TV2EXBRbbAktuellAudioAnalyzer* etc. New programmes can be easily processed by implementing the respective components and a controller for that programme.

All services are integrated via an event and messaging based solution (via an Enterprise Service Bus (ESB)), instead of using a direct point-to-point integration of all services. The services interact with the ESB, allowing it to manage the routing and transformation of messages between the services. All services are static in contrast to dynamic web services that would need a registry of services.

With [D5.4] and further on in [D5.6] an administration dashboard with monitoring component was introduced. The dashboard allows for manual uploading of media resources and their related files, as well as for a subsequent triggering of the different steps of the LinkedTV workflow. All information for a particular media resource and its status within the workflow are presented. An additional monitoring component allows for monitoring the services and processes of the system. This component checks the availability of processes and services and may restart local processes if needed.

6 Findings and Lessons Learned

6.1 General remarks

During the preparation of the validation, it became obvious that the originally defined requirements were not further managed throughout the project. As a result, some of the requirements reported in [D5.1] are outdated by now or still only roughly defined. This led on the one hand to a non-transparent state of requirements, while on the other hand it was hard to assess if, and to what extent unspecific requirements were fulfilled. Especially the quality attribute requirements were described only vaguely. At the beginning of the project this was caused by unclear business goals, nevertheless this should have been revised during the course of the project. However, requirement changes and updates have taken place based on the outcome of the user trials and scenario evaluations. Their influence on the requirements resulted in issues that were tracked within our Issue Tracking system, but they were not formally documented as changed requirements.

6.2 Findings of the architecture validation

In this section findings about the influence of the architecture on the prioritised and further specified quality attribute requirements (see Chapter 4) are given.

6.2.1 Extensibility

Extensibility is the ease with which the capabilities of services can be extended or new services can be added without affecting other services or parts of the system. According to the defined scenarios, extensibility means for the LinkedTV architecture to allow the integration of new services without affecting other services.

Extending the LinkedTV architecture by adding new services or adding new capabilities into existing services is well supported due to its nature of being a service-oriented architecture. With the introduction of the service layer and dedicated service components in [D5.6] the ease of adding customer system services has further grown.

6.2.2 Scalability

Scalability is the ability of a system to handle a growing amount of work in a capable manner, or its ability to be enlarged to accommodate that growth. According to the defined scenarios, scalability means for the LinkedTV architecture to allow the modification of the system's size or volume in order to meet the needs of a customers' use case.

The current version of the LinkedTV architecture is only to some extent scalable. Most of the services can be scaled individually in terms of parallel analysis and processing of content items, but to handle large amounts of content or to cope with a high number of client requests, the system has to be improved. This is already known and was documented in [D5.6], Section 3.1.

6.2.3 Interoperability

Interoperability is the ability of making systems work together. With respect to software, the term interoperability is used to describe the capability of different programs to exchange data. In the context of the scenarios defined in Chapter 4 this means mainly the ability of the LinkedTV system to exchange data with legacy customer systems such as CMS or MAM.

Service-oriented architectures provide a good interoperability because of the standardized technology, which is available for different programming languages and can be deployed on different platforms. Due to the high degree of extensibility, a dedicated importer can be introduced as additional service to the system, allowing the latter to easily inter-operate with legacy customer systems.

6.2.4 Performance

Performance can have different meanings in different contexts. In general, it is related to response time (how long it takes to process a request), throughput (how many requests can be processed) or timeliness (to process a request in an acceptable amount of time). According to the current scenarios, performance is limited to the facet of timeliness, with response time and throughput being currently not relevant.

Although performance can often be an issue in service-oriented architectures due to network delays etc., the SOA of LinkedTV was designed to minimize the overhead coming with a SOA. RESTful Web services for example are more performant in contrast to SOAP-based Web services. Using static services instead of dynamic ones also improves the performance. For the current use case scenario, the performance of the LinkedTV system is not an issue. Depending on the selection of analysis services, videos can be annotated and enriched almost in real-time, as a log file from the longitudinal RBB user trial can prove (see Appendix, p. 30). Nevertheless, for other customer use case scenarios performance might become an issue and should be evaluated again.

6.3 Compliance to relevant standards

Compliance to relevant standards was a major requirement of the LinkedTV architecture. This applies to data formats as well as communication protocols, and APIs, respectively. Section 6.4 of [D5.1] lists the standards that had been foreseen to be relevant at the beginning of the project, grouped in different areas. In the following, we will briefly describe which standards have been proven relevant for LinkedTV. Furthermore, to some standards the LinkedTV project contributed itself; for an in-depth description of LinkedTV compliance and contribution to standards see [D7.7].

Putting aside the very basic standards described in Section 6.4 of [D5.1], such as HTML, HTTP, we will discuss the main standards in the following.

6.3.1 Platform APIs, communication protocols, data exchange formats

REST – all services that are connected via the LinkedTV platform and those of the core platform itself follow the general REST paradigm. Through this, a very lightweight and highly configurable orchestration of services is achieved. Furthermore, the whole approach is open to the integration of other services or the replacement of existing ones – as long as the data format is compatible.

JSON/XML/TTL/SRT Data exchange format – data exchange formats comply to standards JSON, XML and TTL (Turtle); mostly, services offer different formats via content negotiation. Where directly RDF data is generated, in particular via using the EURECOM tv2rdf and tvenricher services, the data exchange format is TTL. In case of the multimedia analysis service (tv2exb – denoted as WP1 REST Service in [D1.4]) the output is an XML-based format named Exmaralda, which, however, is no real standard. To address this, an effort for storing the multimedia analysis results in an MPEG7-compliant format has been just before the end of the project duration. The audio analysis component of the service, however, produces data that is compliant to the SRT format (SubRip Text file format).

6.3.2 TV related standards

TVAnytime – LinkedTV is able to process programme information in the TVAnytime format. In particular, *rbb Aktuell* episodes come with associated metadata describing general information such as genre, title, airtime, resolution, duration and so on.

W3C Media Ontology / EBU Core – internally, LinkedTV uses the W3C Media Ontology standard, which is fully compatible to the EBU Core Metadata standard. Metadata from external formats, such as TVAnytime, is mapped to the internal Media Ontology format, which is also used in the later serialization process performed by the tv2rdf service.

STL/SRT/WebVTT – LinkedTV supports subtitle standard formats STL and SRT; *rbb Aktuell* episodes come with associated subtitles in STL format which are converted into SRT format; the newer standard WebVTT is not supported officially, however, as WebVTT is just an extension of SRT, its support could be easily added if needed.

BBC Programme Ontology – tv2rdf serialization uses the BBC programme ontology[4] for describing TV programme information in more detail (compared to EBU Core / W3C Media Ontology).

HbbTV – The Noterik Springfield Toolkit supports the generation of content according to the HbbTV 1.5 standard.

6.3.3 Standards related to LinkedTV Core technology

The following standards are those, which are related to the core of the LinkedTV approach:

[4] http://www.bbc.co.uk/ontologies/po

W3C Media Fragment URI – media fragments are generated according to the W3C Media Fragment standard.

W3C Open Annotation – annotations to media fragments are generated according to the Open Annotation standard.

6.3.4 Linked Data Vocabularies

Wherever possible, LinkedTV uses existing Linked Data vocabularies. The following lists these vocabularies:

Dublin Core	http://purl.org/dc/elements/1.1/
DBPedia Ontology	http://dbpedia.org/ontology/
Ninsuna	http://multimedialab.elis.ugent.be/organon/ontologies/ninsuna#
Provenance	http://www.w3.org/ns/prov#
OPMV	http://purl.org/net/opmv/ns
FOAF	http://xmlns.com/foaf/0.1/
Timeline	http://purl.org/NET/c4dm/timeline.owl/
LSCOM	http://vocab.linkeddata.es/lscom/
NERD	http://nerd.eurecom.fr/ontology#
Event	http://purl.org/NET/c4dm/event.owl/

6.4 Lessons Learned

While the previous sections presented the findings from the validation of the LinkedTV architecture against the specified non-functional requirements, the following section will explore what kind of lessons we have learned and in how far assumptions and technological decisions are still valid or should be changed in future evolutions of the LinkedTV system.

6.4.1 Cascading integration approach

The cascading integration approach was described in [D5.3]. It states that for the different steps like analysis, serialization, annotation and enrichment there are single integration services that in themselves control which single steps or services are executed within these general areas. This has been certainly a good way to decrease and control the overall complexity, and from an organizational point of view, this led to clear responsibilities within the different LinkedTV work packages. Configuration variants, such as which exact methods or subservices should be executed, are controlled via parameters in the respective REST services. On the downside, however, this approach leads to a decrease in flexibility. For future releases of the LinkedTV platform, we would now rather adopt an approach, which favours the integration of atomic services directly with the LinkedTV platform and a more flexible way to orchestrate these services. The cascading integration approach also entails that the second tier integration services themselves communicate with remote services, while

the general workflow is triggered and controlled by the core LinkedTV platform. By this, the management of the core platform workflow has no control over the availability or response time of the remote service. In practice, this led to the problem where the core workflow could not continue due to the failure of a remote service, without knowing the reason for that interruption. It would be much more resilient, if the core platform could then decide to continue with the processing and just keep triggering the failing service in parallel until it succeeds. Of course, sometimes this would not be possible because subsequent steps depend on the successful termination of a previous step, but often these are also just aggregating services which would only mean that in the end there is just less data available but which could also be added later to make it complete.

6.4.2 Update strategy and interdependence of data

Although it was never stated in any official document, the LinkedTV platform was very much based on the assumption that it is possible to always *add* data from incoming new videos without having to reprocess or remove existing data. Of course, it was always clear that this assumption would not hold through the first stages of the development and integration of services until stable versions of these services would have been achieved. However, it turned out that in order to support finding enrichments and related material from existing processed content, it is in fact necessary to reprocess data whenever new metadata dimensions are added to the content. This is because all existing content is interdependent due to cross-video search functionality. In addition, in order to keep the data consistent it is necessary to reprocess almost the whole chain, because only then the data being indexed by the Solr search service is in sync with the data that has been serialised by tv2rdf. While during the lifetime of the LinkedTV project complete content reprocessing has been done a couple of times, this functionality cannot be supported as a regular maintenance operation. For future productive releases, however, a solution is mandatory, similar to search engines that permanently re-index their data.

6.4.3 Indexing and enrichment workflow steps

When the original LinkedTV workflow is triggered automatically by the LinkedTV platform, it follows the steps: ingestion – analysis – serialization/annotation – enrichment. In the end, this chain has slightly changed. While the first three steps remain the same, after the serialization/annotation step an additional step *Indexing* has been added. The indexing step enables easy search features for finding annotations and media fragments (in particular chapters and shots) for LinkedTV applications. Since the indexing process uses data that has been generated by the tv2rdf annotation service (in particular identifiers, such as chapter IDs), it has to be performed *after* the annotation workflow step. Therefore, indexing is an additional workflow step, which has not been foreseen in the original concept but proved to be important and has been added later.

On the other hand, the automatic enrichment workflow step now does not seem to be mandatory anymore at all, since the curation and manual enrichment step (see next

subsection) has gained much more importance in the last phase of the project. The automatic enrichment by the LinkedTV platform would now only be important for unsupervised, un-curated scenarios, which, however, in the broadcast industry do not really exist, since broadcasters want to control what is being added and published as related content. Therefore, the automatic enrichment step is now treated as optional and could be automatically performed in scenarios where this is needed.

6.4.4 Curation and manual enrichment and persistence

As mentioned above, the curation of content has become a much more important feature of LinkedTV than thought of in the beginning of the project, having also a big impact on the original platform architecture and workflow. In fact, in the two LinkedTV scenarios (namely LinkedCulture and LinkedNews), the Editor Tool and the curation process have become the core steps for publishing a video and its related content. The automatic analysis and annotation steps here are important to lay the foundation, make initial chapter segmentation recommendations, and show which entities have been detected. However, the editor in the curation process finally decides what will be displayed. This includes correction of the automatic chapter segmentation if needed, checking whether important entities have been detected and successfully linked to Linked Data sources. The most important curation step is then the manual enrichment by finding related content via different connected enrichment services (see [D6.5] for details). For the general LinkedTV architecture, this means that many connected LinkedTV services have moved from the core platform to the Editor Tool, namely all the enrichment services that are now evoked manually during the curation process. The Editor Tool in itself has also grown from a mere core platform client to a full web-based application with its own storage facility, or, to put it the other way round, the Editor Tool has become an own important, full business component of the overall LinkedTV architecture.

7 Summary

In this report, the methodology and the findings of the evaluation of the LinkedTV architecture have been presented. Overall, the LinkedTV architecture is a suitable basis for fulfilling the current requirements. If it also fulfils the future requirements, depends on those requirements and the related use cases. The architecture can easily be extended and is interoperable with legacy customer systems, which is a good starting point for customer specific adaptations. Nevertheless, some risks exist, that should be evaluated again when other customer use cases are defined. For example, performance might be an issue. Other quality attributes have not been evaluated at all, since they were neither part of the initial set of requirements nor further specified by current scenarios.

By reporting on the findings of the evaluation and by providing lessons learned, we hope to contribute both, to the improvement of the LinkedTV architecture and to the improvement of other similar architectures. We see the LinkedTV architecture and the implemented LinkedTV platform, i.e. the core platform as well as the different integrated services, as one of the first prototypes of an emerging kind of software systems, which we call *Linked Media Management Systems (LMMS)*. Compared to Content Management Systems, LMMS do not deal with the publishing of content in the first place but rather with annotating and linking content, in particular multimedia content, and is thus a perfect complement to Content Management Systems, and Content Delivery Networks as well. Linked Media Management as defined by LinkedTV includes the following main features:

- **Ingesting** media resources and related metadata files, such as subtitles; this can be a physical copy of the media resource as well as just a link to a media resource on the Web
- **Analysing** the video and audio tracks of a media resource
- **Generating fragments** of a media resource, such as chapters, scenes, shots, or keyword indexes
- Exposing media resource fragments with playable **media fragment HTTP URIs**
- **Annotating** these media fragment URIs with information on which kind of fragment this is (i.e. chapter, shot, keyword, detected face, etc.), with information on when the annotation was created, by which service, etc.
- **Detecting named entities** such as persons, events, locations, objects in these annotations, thus **semantically indexing** the media fragments
- **Linking** these detected entities with Linked Data resources on the Web, such as DBpedia
- **Storing** all this metadata information in a metadata repository; LinkedTV has chosen a RDF format and a RDF Triple Store for this, because this quite naturally fits with the Linked Data sources. Hence, alternatives such as other Graph DBs or NoSQL Databases could also be used.
- Offering **enrichment services** for finding related content based on the annotations

- **Exposing** access to this data through a consistent interface, preferably a REST interface for building clients and applications, such as applications for HbbTV, Second Screen devices or further services like personalisation services
- **Managing the workflow** of all these different steps and phases
- Offering a tool for **editing and curating** the annotations and enrichments
- Offering a framework for **integrating** these different systems and services, as well as further ones, e.g. integration with Content Management Systems, or Rights Management Systems

The LinkedTV platform implements the above-mentioned features in a prototypical way. It provides a basis for future customer based adaptations and for further developments.

8 Bibliography

[ATAM98] R. Kazman, M. Klein, M. Barbacci, H. Lipson, T. Longstaff, S.J. Carriere; The Architecture Tradeoff Analysis Method; Proc. Fourth International Conference on Engineering of Complex Computer Systems (ICECCS), 1998

[Bab09] M.A. Babar, I. Gorton; Software Architecture Review: The State of Practice; IEEE Computer, vol. 42, no. 7, pp. 26-32, 2009

[Cle02] P. Clements, R. Kazman, and M. Klein, Evaluating Software Architecture, Addison-Wesley, 2002.

[D1.4] E. Apostolidis, et all; Visual, text and audio information analysis for hypervideo, final release; LinkedTV deliverable; 2014

[D1.5] J. Blom; LinkedTV annotation tool, final release; LinkedTV deliverable; 2014

[D5.1] R. Fricke, J. Thomsen; LinkedTV Platform and Architecture; LinkedTV deliverable; 2012

[D5.3] J. Thomsen, A. Sarioglu, S. Weinberg, R. Fricke; First LinkedTV End-to-end Platform, LinkedTV deliverable; 2013

[D5.4] J. Thomsen, A. Sarioglu, S. Weinberg, R. Fricke; Final LinkedTV Integration Platform; LinkedTV deliverable; 2013

[D5.5] D. Ockeloen, P van Leeuwen; LinkedTV front-end: video player and MediaCanvas API version 2; LinkedTV deliverable; 2014

[D5.6] J. Thomsen; Final LinkedTV End-to-End Platform; LinkedTV deliverable; 2014

[D6.5] L. Nixon; Final evaluation, LinkedTV deliverable; 2015

[D7.7] L. Nixon; Dissemination and standardisation report v3; LinkedTV deliverable; 2015

[D8.7] H. Horstmann, K. Stanoevska-Slabeva, R. Fricke; LinkedTV business models, V2; LinkedTV deliverable; 2014

[IEEE1471] IEEE 1471:2000, Recommended Practice for Architectural Description of Software-Intensive Systems, IEEE Press, 2000.

[SAAM96] R. Kazman, G. Abowd, L. Bass, M. Webb; Scenario-based Analysis of Software Architecture; IEEE Software, pp. 47-55, 1996

[SARA02] H. Obbink, P. Kruchten, W. Kozaczynski, H. Postema, A. Ran, L. Dominic, R. Kazman, R. Hilliard, W. Tracs and E. Kahane; Software Architecture Review and Assessment (SARA) Report, Version 1.0, tech. report, SARA WG, 2002

[TARA11] E. Woods; Industrial Assessment using TARA; Ninth Working IEEE/IFIP Conference on Software Architecure, 2011

9 Appendix

The following excerpt from a log file gives an example of the steps of the processing workflow and presents details on the duration of each single step and on the complete workflow. It shows that a 30-minute video content can be completely processed by the LinkedTV workflow in real-time. This of course depends on the selection of analysis services performed on the content.

```
[2015-03-20_06:00:01] Script started.

[2015-03-20_06:00:01] Fetching mediafiles for '20150319'.

[2015-03-20_06:00:01] running RBB Fetch Script ...

writing File: /root/rbb/files/rbbaktuell_20150319.html - OK.

writing File: /root/rbb/files/rbbaktuell_20150319_sdg_m_16_9_512x288.mp4 - OK.

[2015-03-20_06:01:26] Copying Mediafiles to NFS-Share...

[2015-03-20_06:01:57] MEDIALINK: Sucessfully created Mediaresource
<http://api.linkedtv.eu/mediaresource/3c1e2134-cebe-11e4-951c-f8bdfd0abfbd>!

[2015-03-20_06:01:57] MEDIALINK: performing REST-Call for visual Analysis (without
ASR):

[2015-03-20_06:01:58] MEDIALINK: Call returned: The REST call has been received.
Processing will start as soon as the video file is downloaded.

[2015-03-20_06:01:59] MEDIALINK: Media file
<http://api.linkedtv.eu/data/RBB/tva/rbbaktuell_20150319.html> successfully linked!

[2015-03-20_06:01:59] looking for corresponding subtitle ...

[2015-03-20_06:01:59] TOSRT: successfully created
'/mnt/data/RBB/srt/rbbaktuell_20150319.srt'.

[2015-03-20_06:02:01] MEDIALINK: Media file
<http://api.linkedtv.eu/data/RBB/srt/rbbaktuell_20150319.srt> successfully linked!

[2015-03-20_06:02:01] TOSRT: successfully created
'/mnt/data/RBB/srt/rbbaktuell_20150319.srt'.

[2015-03-20_06:02:01] MEDIALINK: MediaFile
<http://api.linkedtv.eu/data/RBB/srt/rbbaktuell_20150319.srt> already linked!

[2015-03-20_06:02:01] deleting local files ...

[2015-03-20_06:02:01] calling ingest script ...

('2015-03-20 06:02:02.796833 INFO Trying to ingest ', 'rbbaktuell_20150319')

2015-03-20 06:02:02.803425 INFO
/mnt/data/RBB/mp4/rbbaktuell_20150319_sdg_m_16_9_512x288.mp4 OK

2015-03-20 06:02:02.803722 INFO New UUID = 3f3d20e0-cebe-11e4-90b3-005056a40191

Connecting to linkedtv mysql db

2015-03-20 06:02:02.901694 INFO Connected to MySQLDB linkedtv

2015-03-20 06:02:02.901775 INFO Try SQL:  INSERT INTO MediaResourceUUID (id,
locator, publisher, titleName, linkedTVStatusId, dateInserted, frameSizeWidth,
frameSizeHeight, lang, collection) VALUES ( '3f3d20e0-cebe-11e4-90b3-005056a40191',
'http://api.linkedtv.eu/data/rbb/mp4/rbbaktuell_20150319_sdg_m_16_9_512x288.mp4',
'rbb', 'rbb AKTUELL tv.ed.link 21:45', 1,'2015-03-20 06:02:02.901761', 512, 288,
'German', 'rbb Aktuell')

2015-03-20 06:02:02.901788 INFO Try SQL:  INSERT INTO MediaResourceUUID (id,
locator, publisher, titleName, linkedTVStatusId, dateInserted, frameSizeWidth,
frameSizeHeight, lang, collection) VALUES ( '3f3d20e0-cebe-11e4-90b3-005056a40191',
'http://api.linkedtv.eu/data/rbb/mp4/rbbaktuell_20150319_sdg_m_16_9_512x288.mp4',
'rbb', 'rbb AKTUELL 19.03.2015 21:45', 1,'2015-03-20 06:02:02.901761', 512, 288,
'German', 'rbb Aktuell')
```

```
2015-03-20 06:02:02.925638 SUCCESS MediaResource successfully created

Connecting to linkedtv mysql db

2015-03-20 06:02:02.927581 INFO Connected to MySQLDB linkedtv

2015-03-20 06:02:02.927631 INFO Try SQL:  INSERT INTO
MediaResourceRelationUUID(relationTarget, relationType, mediaResourceId) VALUES
('http://api.linkedtv.eu/data/rbb/mp4/rbbaktuell_20150319_sdg_m_16_9_512x288.mp4',
'original-video', '3f3d20e0-cebe-11e4-90b3-005056a40191')

2015-03-20 06:02:02.931858 SUCCESS MediaResourceRelation successfully created

2015-03-20 06:02:02.931944 INFO Trying to ingest TVA file

2015-03-20 06:02:02.932826 INFO Linking TVA file
/mnt/data/RBB/tva/rbbaktuell_20150319.html

Connecting to linkedtv mysql db

2015-03-20 06:02:02.945426 INFO Connected to MySQLDB linkedtv

2015-03-20 06:02:02.945484 INFO Try SQL:  INSERT INTO
MediaResourceRelationUUID(relationTarget, relationType, mediaResourceId) VALUES
('http://api.linkedtv.eu/data/rbb/tva/rbbaktuell_20150319.html', 'tva', '3f3d20e0-
cebe-11e4-90b3-005056a40191')

2015-03-20 06:02:02.954403 SUCCESS MediaResourceRelation successfully created

2015-03-20 06:02:02.995136 INFO Trying to ingest SRT file

2015-03-20 06:02:02.995997 INFO Linking SRT file
/mnt/data/RBB/srt/rbbaktuell_20150319.srt

Connecting to linkedtv mysql db

2015-03-20 06:02:02.997234 INFO Connected to MySQLDB linkedtv

2015-03-20 06:02:02.997286 INFO Try SQL:  INSERT INTO
MediaResourceRelationUUID(relationTarget, relationType, mediaResourceId) VALUES
('http://api.linkedtv.eu/data/rbb/srt/rbbaktuell_20150319.srt', 'srt', '3f3d20e0-
cebe-11e4-90b3-005056a40191')

2015-03-20 06:02:02.997601 SUCCESS MediaResourceRelation successfully created

2015-03-20 06:02:02.997716 INFO checking if exb file is ready ...

http://160.40.50.201:8000/status/rbbaktuell_20150319_sdg_m_16_9_512x288

2015-03-20 06:02:03.176959 FILE_DOWNLOAD_STARTED

2015-03-20 06:05:34.283065 INFO checking if exb file is ready ...

http://160.40.50.201:8000/status/rbbaktuell_20150319_sdg_m_16_9_512x288

2015-03-20 06:05:34.441695 SHOT_CONCEPT_CHAPTER_DETECTION_ANALYSIS_STARTED

2015-03-20 06:06:04.469617 INFO checking if exb file is ready ...

http://160.40.50.201:8000/status/rbbaktuell_20150319_sdg_m_16_9_512x288

2015-03-20 06:21:09.883064 INFO checking if exb file is ready ...

http://160.40.50.201:8000/status/rbbaktuell_20150319_sdg_m_16_9_512x288

2015-03-20 06:21:10.035745 ANALYSIS_COMPLETED

2015-03-20 06:21:10.035833 INFO fetching video result:
http://160.40.50.201:8000/result/rbbaktuell_20150319_sdg_m_16_9_512x288

2015-03-20 06:21:16.582898 SUCCESS EXB file successfully written (size: 1376256):
/mnt/data/RBB/exb/rbbaktuell_20150319.exb

2015-03-20 06:21:16.582958 INFO: Getting UUID for rbbaktuell_20150319 Connecting to
linkedtv mysql db

2015-03-20 06:21:16.584907 INFO Connected to MySQLDB linkedtv

2015-03-20 06:21:16.584971INFO Try SQL SELECT a.id FROM MediaResourceUUID a,
MediaResourceRelationUUID b WHERE a.id = b.mediaResourceId AND b.relationType =
'original-video' and b.relationTarget like '%rbbaktuell_20150319%'

2015-03-20 06:21:16.804859 SUCCESS UUID = 3f3d20e0-cebe-11e4-90b3-005056a40191

2015-03-20 06:21:16.805013 INFO: UUID = 3f3d20e0-cebe-11e4-90b3-005056a40191
```

```
Connecting to linkedtv mysql db

2015-03-20 06:21:16.806935 INFO Connected to MySQLDB linkedtv

2015-03-20 06:21:16.806987 INFO Try SQL:  SELECT LinkedTVStatusID FROM
MediaResourceUUID WHERE Id = '3f3d20e0-cebe-11e4-90b3-005056a40191'

2015-03-20 06:21:16.807633 INFO status is TRANSFERRED, continue Connecting to
linkedtv mysql db

2015-03-20 06:21:16.808649 INFO Connected to MySQLDB linkedtv

2015-03-20 06:21:16.808704 INFO Try SQL:  INSERT INTO
MediaResourceRelationUUID(relationTarget, relationType, mediaResourceId) VALUES
('http://api.linkedtv.eu/data/RBB/exb/rbbaktuell_20150319.exb', 'exmaralda',
'3f3d20e0-cebe-11e4-90b3-005056a40191')

2015-03-20 06:21:16.809101 INFO id: 11940 Connecting to linkedtv mysql db

2015-03-20 06:21:16.810318 INFO Connected to MySQLDB linkedtv

2015-03-20 06:21:16.810376 INFO Try SQL:  UPDATE MediaResourceUUID SET
linkedtvStatusID = 3 WHERE id = '3f3d20e0-cebe-11e4-90b3-005056a40191'

Connecting to linkedtv mysql db

2015-03-20 06:21:16.812315 INFO Connected to MySQLDB linkedtv

2015-03-20 06:21:16.812377INFO Try SQL SELECT locator FROM MediaResourceUUID WHERE
id = '3f3d20e0-cebe-11e4-90b3-005056a40191'

2015-03-20 06:21:16.828895 SUCCESS locator =
http://stream19.noterik.com/progressive/stream19/domain/linkedtv/user/rbb/video/1810/

2015-03-20 06:21:16.829060 INFO: creating mediaresource at tv2rdf
http://linkedtv.eurecom.fr/tv2rdf/api/mediaresource/3f3d20e0-cebe-11e4-90b3-
005056a40191?locator=http://stream19.noterik.com/progressive/stream19/domain/linked
tv/user/rbb/video/1810/

2015-03-20 06:21:16.986473 SUCCESS: Mediaresource successfully created at tv2rdf:
3f3d20e0-cebe-11e4-90b3-005056a40191

2015-03-20 06:21:22.177120 INFO fetching
http://linkedtv.eurecom.fr/tv2rdf/api/mediaresource/3f3d20e0-cebe-11e4-90b3-
005056a40191/serialization?metadataType=legacy

2015-03-20 06:21:22.943960 SUCCESS TTL file successfully written (size: 14651):
/mnt/data/RBB/ttl/rbbaktuell_20150319_tva.ttl

2015-03-20 06:21:31.770382 INFO fetching
http://linkedtv.eurecom.fr/tv2rdf/api/mediaresource/3f3d20e0-cebe-11e4-90b3-
005056a40191/serialization?metadataType=exmaralda

2015-03-20 06:21:33.454365 SUCCESS TTL file successfully written (size: 1719359):
/mnt/data/RBB/ttl/rbbaktuell_20150319_exb.ttl

Connecting to linkedtv mysql db

2015-03-20 06:22:08.567185 INFO Connected to MySQLDB linkedtv

2015-03-20 06:22:08.567254 INFO Try SQL:  UPDATE MediaResourceUUID SET
linkedtvStatusID = 4 WHERE id = '3f3d20e0-cebe-11e4-90b3-005056a40191'

2015-03-20 06:22:08.947748 INFO Now waiting 300 seconds ....

subtitle

2015-03-20 06:27:09.035804 INFO fetching
http://linkedtv.eurecom.fr/tv2rdf/api/mediaresource/3f3d20e0-cebe-11e4-90b3-
005056a40191/serialization?metadataType=subtitle

404

2015-03-20 06:27:09.129735 ERROR No serialization result available for
http://linkedtv.eurecom.fr/tv2rdf/api/mediaresource/3f3d20e0-cebe-11e4-90b3-
005056a40191/serialization?metadataType=subtitle


2015-03-20 06:27:09.375918 INFO Now waiting 300 seconds ....

subtitle
```

```
2015-03-20 06:32:09.476226 INFO fetching
http://linkedtv.eurecom.fr/tv2rdf/api/mediaresource/3f3d20e0-cebe-11e4-90b3-
005056a40191/serialization?metadataType=subtitle

200

2015-03-20 06:32:18.092372 SUCCESS TTL file successfully written (size: 311839):
/mnt/data/RBB/ttl/rbbaktuell_20150319_srt.ttl

2015-03-20 06:32:21.961188 INFO: java -jar ~/TV2Lucene/script/TV2Lucene.jar -
provider RBB -exm /mnt/data/RBB/exb/rbbaktuell_20150319.exb -srt
/mnt/data/RBB/srt/rbbaktuell_20150319.srt -metadata
/mnt/data/RBB/tva/rbbaktuell_20150319.html -videoId 3f3d20e0-cebe-11e4-90b3-
005056a40191

TRYING TO PARSE

PARSED

srt: /mnt/data/RBB/srt/rbbaktuell_20150319.srt

exm: /mnt/data/RBB/exb/rbbaktuell_20150319.exb

metadata: /mnt/data/RBB/tva/rbbaktuell_20150319.html

provider: RBB

videoId: 3f3d20e0-cebe-11e4-90b3-005056a40191

http://localhost:8983/solr/RBBindex

------- connection to solr server ok ----- metadata processed subtitles processed

------- input file processed ------------- /home/linkedtv/TV2Lucene/script there
are 323 concepts

DATE: Thu Mar 19 22:45:00 CET 2015

added docs of type : Shot

added docs of type : Chapter

------- documents added to solr server --- [2015-03-20_06:34:05] script finished.
```

Excerpt from log file of the annotation and enrichment workflow