Enhancing and Re-Purposing TV Content for Trans-Vector Engagement (ReTV)
H2020 Research and Innovation Action - Grant Agreement No. 780656

ReTV

Enhancing and Re-Purposing TV Content for Trans-Vector Engagement

Deliverable 4.2 (M24)

Trans-Vector Platform, Technology Roadmap and Revised Prototype

Version 1.0

This document was produced in the context of the ReTV project supported by the European Commission under the H2020-ICT-2016-2017 Information & Communication Technologies Call
Grant Agreement No 780656

DOCUMENT INFORMATION

Delivery Type	Report
Deliverable Number	4.2
Deliverable Title	Trans-Vector Platform, Technology Roadmap and Revised Prototype
Due Date	M24
Submission Date	31.12.2019
Work Package	WP4
Partners	WLT, GENISTAT
Author(s)	Arno Scharl, Max Göbel (WLT), Basil Philipp (Genistat), Kostas Apostolidis, Vasileios Mezaris (CERTH), Lyndon Nixon (MOD)
Reviewer(s)	RBB, NISV
Keywords	System Architecture, Integration
Dissemination Level	PU
Project Coordinator	MODUL Technology, Am Kahlenberg 1, 1190 Vienna, Austria.
Contact Details	Coordinator: Dr Lyndon Nixon (nixon@modultech.eu) R&D Manager: Prof Dr Arno Scharl (scharl@weblyzard.com) Innovation Manager: Bea Knecht (bea@zattoo.com)

Revisions

Version	Date	Author	Changes
0.1	26.09.2019	Arno Scharl	Initial version based on D4.1
0.2	04.10.2019	Max Göbel	Preparation for content updates
0.25	08.11.2019	Max Göbel	Refactoring, Sections 3,4,5,6
0.3	25.11.2019	Max Göbel	Added Section 7 (WP4)
0.4	26.11.2019	Basil Philipp	Updated Genistat content, Sections 3,4,5
0.5	26.11.2019	Kostas Apostolidis, Vasileios Mezaris	Updated CERTH content, Sections 2,3,5
0.6	26.11.2019	Lyndon Nixon	Updated MODUL content
0.7	09.12.2019	Max Göbel	Major restructuring
0.75	10.12.2019	Max Göbel	Proof-reading, flow optimisation
0.8	11.12.2019	Max Göbel	Abstract, conclusion, QA-ready
0.9	19.12.2019	Miggi Zwicklbauer, Rasa Bocyte, Lyndon Nixon, Arno Scharl	QA comments; reviews, edits and optimisations.
1.0	28.12.2019	Max Göbel	Final version

Statement of Originality

TABLE OF CONTENTS

GLOSSARY

Term	Description
Trans-Vector Platform (TVP)	An open architecture of components and subsystems that, in flexible configurations, enable a content owner or creator to optimise their content's impact and reach through contextualisation, prediction, and adaptation mechanisms.
Electronic Program Guide (EPG)	A metadata feed of scheduling information for upcoming video broadcasts.
Vector	A place where content can be published. This can be a social media vector like Facebook or Twitter or a more traditional vector like linear TV.
Application Programming Interface (API)	Well-defined technical machine-to-machine interface for data exchange.
Semantic Knowledge Base (SKB)	A knowledge graph service operated by MODUL and WLT.

EXECUTIVE SUMMARY

At the end of the second project year, ReTV has completed a first major revision of its central Trans-Vector Platform (TVP). This revision encapsulates all technical results from the project partners into a coherent modular platform, enabling broadcasters, media archives and other digital content owners to tap into the next generation of digital content publication through provision of prediction, personalisation and content transformation capabilities, with the common goal of maximising the reach and engagement with their audience.

In response to stakeholder feedback received from the use cases, the target architecture as initially drafted in the first prototype (reported in deliverable D4.1) has evolved both technically and conceptually into a comprehensive platform of loosely-coupled software components, organised along the three main functional capabilities: **listening**, **prediction**, and **adaptation**.

At the end of the second year all components have been integrated, extending the functional capacity of the TVP in line with the use case specifications. To validate the correctness and completeness of the TVP, we present four TVP configurations that showcase the added value of the TVP by solving industry limitations confirmed in stakeholder demonstrations.

The resulting roadmap for the final year is aligned with the deliverables and milestones of the *Description of Work* of the ReTV project. As such, it will support the technical project management in ensuring that TVP development proceeds according to plan, working towards the timely delivery of the final target architecture.

1 INTRODUCTION

This document presents the first major revision of the **Trans-Vector Platform** (TVP) at M24, a year after its initial prototype was reported in Deliverable D4.1 in M12. The TVP is a system of modular components that can be used together in different configurations by digital content stakeholders to recommend and repurpose their media semi-automatically and distribute it optimally across multiple publication vectors. It entails all technical outputs from ReTV, composed as a system of independent, distributed modules that communicate with each other through APIs. The end user accesses the functionalities of the TVP through applications that sit on top of the TVP, and which bundle the relevant TVP components together. In ReTV, four applications were devised, each addressing a distinct use case scenario developed and refined by the use case partners of the consortium.

The TVP consists of three types of technical components:

1. **Individual Web services** (delivered by WPs 1-3) which could also operate standalone via REST APIs. Those components provide data management (collection, annotation), predict usage patterns (based on events, Web and social media metrics, TV audience figures) and analyse and repurpose digital media for recommendation and (re)publication across vectors.

2. **Integration components** to combine individual services into a workflow and a data mapping between the data generated and consumed by distinct components (as delivered by WP4). Different integration choices define the configuration of multiple components into a TVP instance.

3. **TVP applications** for supporting different use cases, including scenario-specific user interfaces on top of TVP components, in order to provide access to TVP data and services to both professional users on the media stakeholder side as well as content consumers on the public side (WPs 5 and 6 define, design and evaluate a set of scenarios for the TVP to support).

The main focus of WP4 lies in the architecture design and integration of software components as developed by the technical consortium partners towards a working platform that addresses and solves the user needs as elaborated by the use case partners. Since the last reporting of WP4 in M12, the ReTV consortium has focused its initial vision of user needs into four precise applications with clear specifications of user workflows and user needs.[1] All four scenarios use the vast majority of TVP components in different configurations.

In this deliverable, we present the TVP as a fully integrated mesh of software components, ready to be utilised by the TVP applications. While we envision small-scale adaptations of the TVP to still be made during the final year, as more feedback is gathered from application development, we see the TVP as complete and functional at this time. We plan on spending the remainder of this project's runtime on high performance and high availability tuning of the TVP towards installments of first live TVP instances in form of stable, scalable TVP applications.

D4.2 has received a major restructuring since the last WP4 deliverable D4.1. In light of more refined technical specifications of the use case scenarios and applications during the last 12 months, we have chosen to depart from the work package-centric structure pursued in D4.1, and instead align the structure of D4.2 with the TVP as a cross-work package entity. To this end,

[1] See deliverables D5.2 and D6.2 for detailed descriptions of the professional user and consumer scenarios, respectively.

we start D4.2 with a technical overview of the TVP in Section 2, incorporating updates from all technical work packages where applicable. Section 3 then describes the integration efforts undertaken to bundle the various technical components into a holistic, coherent toolkit via the introduction of data exchange formats and interfaces. Section 3 also introduces processes for platform evaluation that allow us to assert the validity, correctness, and completeness of the TVP integration. Finally, Sections 4 and 5 demonstrate the use of the TVP by highlighting the technical integration of the two use case scenarios, *Engagement Monitoring* and *Personalisation*, as defined in WP5 and WP6, respectively. As such, Section 4 discusses the TVP as a back end for the TVP applications *Topics Compass* and *Content Wizard*, and Section 5 highlights the TVP as a back end for the TVP applications *4u2 Chatbot* and *Content sWitch*. We conclude this deliverable with an outlook on the planned roadmap for TVP development during the final year of the project.

During the reporting period M12-M24, all four WP4 tasks were active. Progress of Task 4.1, *System Architecture and Integration* is reported on in Sections 2, *Trans-Vector Platform*, and 3, *Platform Integration And Evaluation*. Task 4.2, *Visualization of Content and Audience Flows*, is covered in Sections 3.2, *Metadata APIs*, and 4.1, *Topics Compass*. Progress on Task 4.3, *TVP Visual Dashboard*, is reported in Section 4.1, *Topics Compass*, which is the ReTV brand name that has replaced *TVP Visual Dashboard*. Finally, Task 4.4, *Scalability and Distributed Content Processing Strategy*, is reported in Sections 2, *Trans-Vector Platform*, and 3, *Platform Integration And Evaluation*.

2 TRANS-VECTOR PLATFORM

The TVP is the back end system that enables all TVP applications as identified and specified in the use cases (WP5 and WP6), and as reported in D5.2, *First Validation of Engagement Monitoring Prototype*, and D6.2, *First Validation of Personalization Prototype*.

2.1 OVERVIEW AND OBJECTIVES

The TV content publication industry faces a diversification of its audience across a growing set of media vectors, while at the same time lacking the tools to efficiently adapt their content to these new publication vectors for optimised impact.

The TVP addresses the problem that adapting content for a growing set of publishing vectors is still a manual, labour-intensive task (c.f. Listing 1, excerpt from the ReTV Description of Work).

> ReTV aims to provide **broadcasters** and **content distributors** with technologies and insights to leverage the converging **digital media landscape.** By advancing the state of the art in the **analysis** of this media landscape and providing novel methods to dynamically **re-purpose** content for an array of **media vectors** (= all relevant digital channels), a **Trans-Vector Platform (TVP)** will provide these stakeholders with the ability to *"publish to all media vectors with the effort of one".* It will empower broadcasters and brands to continuously **measure** and **predict** the **success** of their content and advertisements in terms of **reach** and **audience engagement** across vectors, allowing them to optimize **decision making** processes.

Listing 1: Objective of the TVP, as described in the ReTV Description of Work.

In ReTV, the automation of content repurposing for cross-vector publication has been aligned within three distinct capacities that the consortium considers essential for an effective TVP to exhibit. In particular, the desired TVP capacities are:

- **Listening** - given an existing TV show, monitor audience feedback and the public online debate. This helps to contextualise content publication and transformation as a central KPI (success metric), and to repurpose archival content, for example to provide historical context about a particular topic currently discussed online.

- **Prediction** - given current world events, cultural trends, and personal preferences, predict the most relevant content for publication, at the best time (scheduling), and along the highest-impact vector.

- **Transformation** - given an existing TV format, allow fully-automatic content transformation for various audiences (personalisation) and different publication vectors.

The TVP as designed in WP4 and described in this deliverable is a toolkit of individual software components developed within WP1, WP2, and WP3, respectively, integrated into a closely-knit mesh of interconnected TVP subsystems that address the very core capacities outlined above. Through its flexibility – achieved by a loosely-coupled microservice architecture of single-functional components with strictly-defined interfaces – the TVP allows for a multitude of functional instances to be realised that address any of the above core capacities. In the scope of the ReTV project, we have defined four use case scenarios for which applications are being developed. The chosen use case applications are guided by the project's content partners, and have integrated multiple feedback rounds from industry stakeholders received during the first two years of the project. With the TVP (in the form of the professional scenarios) a media professional can maximise the impact of their media content publication guided step by step by AI-powered prediction and recommendation functionalities, without additional manual effort or any technical background required.

2.2 PLATFORM ARCHITECTURE

WP4 is concerned with the overall architecture of the TVP as well as the integration of the individual components into coherent, scalable, and adaptive workflows that optimally reflect the functional requirements of the ReTV applications and thereby best address the use case scenarios from WP5 and WP6, respectively. Over the course of the last two years, the TVP architecture has evolved into a tightly-integrated mesh of components that supports rapid prototyping of new integration ideas from user testing in the use case scenarios into the ReTV applications. This flexibility is achieved by strict adherence of all technical consortium partners to the microservice-based TVP architecture, where each TVP component is required to have a well-defined API and must be limited to a fixed number of related functionalities.

2.2.1 Technical Innovation

The core motivation behind the TVP is an industrial need for an open toolkit of easy-to-use tools for the content creator and scheduler to optimise the reach and impact of video content to be broadcast across an ever growing set of publication vectors. The TV industry is strongly relying on rigid and closed systems to organise video content creation and publication. With the TVP, we introduce an open and flexible architecture for the media industry that opens new possibilities to connect and integrate with existing and new components and thereby to freely accomodate for new uses as they arise in the future.

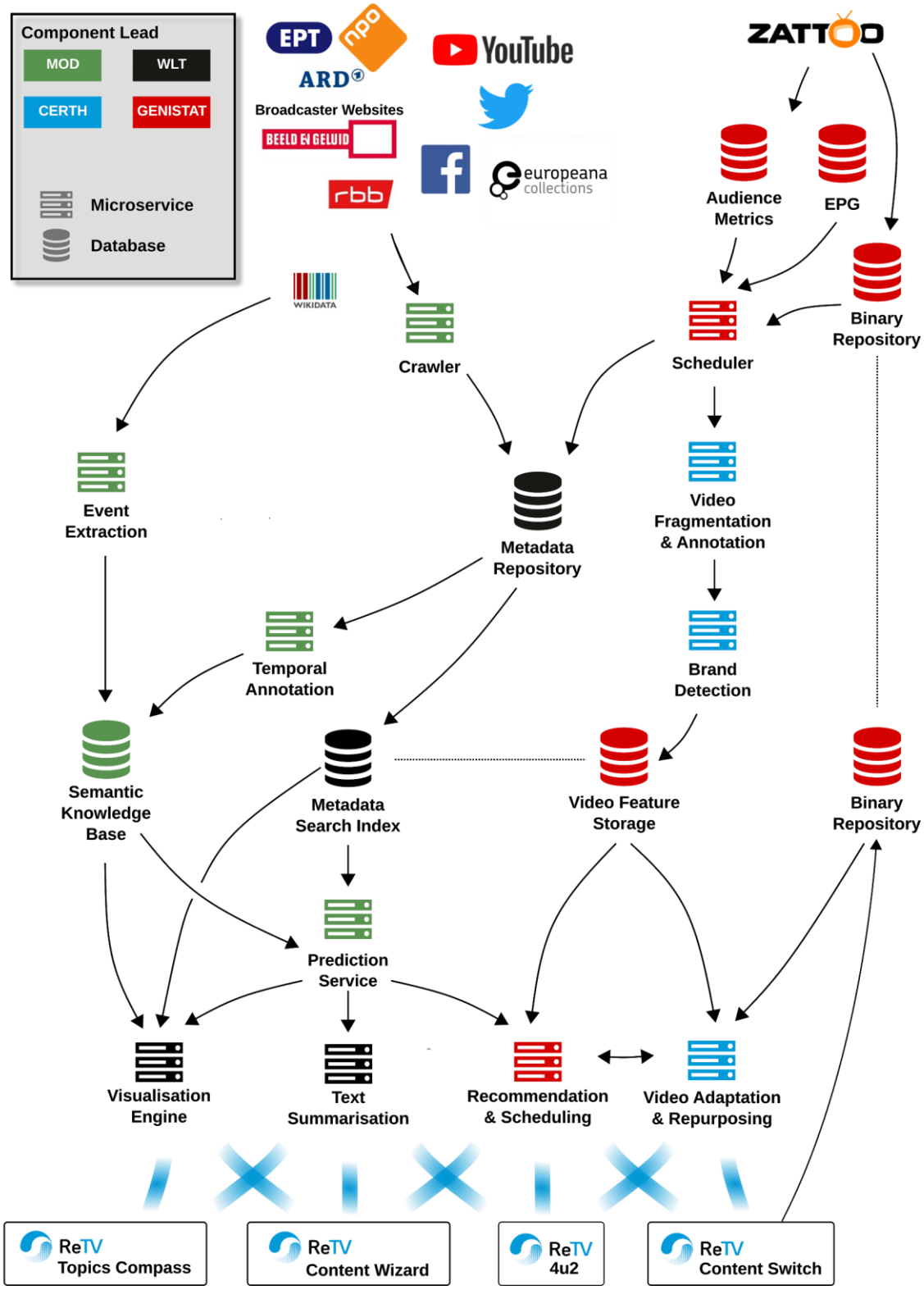


Figure 1: TVP architecture as a modular design of microservices, controllers and databases, colour-coded by lead developer of each component.

The TVP follows a modular design to allow for the highest flexibility in terms of component reuse across the various user-facing TVP applications. Its modular architecture is the technical realisation of the bold vision of reinventing TV, allowing for a reorganisation of modules akin to a toolkit to best match a wide variety of real-world TV content publication scenarios. It further facilitates the commercial dissemination of the TVP beyond the lifetime of the ReTV project by mitigating the risk of blockage of individual TVP components in larger, integrated deployments, with each component of the TVP being fully operational in isolation as a containerised micro application, independent of existing code bases.

Where possible, *application state* should be achieved via single-purpose data storages (both for binary and metadata content), and possibly equipped with full CRUD (Create, Retrieve, Update and Delete) API access. The microservice components operate stateless on the content streams. The microservice-based TVP architecture is shown in Figure 1. In this figure, the high integration across consortium members is particularly apparent in the many data flows that cross between the metadata workflows on the left of the figure, and the binary data workflows on the right. To demonstrate the evolution of the TVP since D4.1, Appendix A, *Initial TVP Architecture (D4.1)*, presents the TVP architecture diagram after M12. Since then, the data flows have become more precise through specification of APIs and respective assignment of functional responsibilities.

2.2.2 Data Flows

From a technical perspective, the TVP components are driven by two central processing workflows:

- A **binary processing workflow**, where TV video (binary) content is ingested, analyzed, transformed, and the results pushed into the TVP applications *Content Wizard*, *4u2 Chatbot*, and *Content sWitch*. The binary workflow is the backbone of content repurposing in ReTV.
- A **metadata processing workflow**, where TV content and news metadata is ingested, integrated, analyzed, and the results pushed into the TVP applications *Content Wizard*, *Topics Compass, 4u2 Chatbot*, and *Content sWitch*. The metadata workflow enables the prediction capabilities of ReTV via contextualization of binary data into a larger public topic debate.

Both workflows have data flow from raw formats towards enriched or transformed formats in a one-way direction. The exception is the *Content sWitch*, where the transformed binary data may be re-ingested into the binary workflow via a feedback loop. The two main workflows along the various TVP components into the applications are shown in Figure 1. In this figure, the core components of the TVP are described as either a microservice or a database component. A *microservice* here is an enrichment or transformation service that does not maintain state, and is therefore by design horizontally scalable. A *database* represents any form of persistent storage component, generally also providing APIs for retrieval or ingestion.

The left half of Figure 1 illustrates data flows for metadata (textual formats), whereas the right half of the figure shows the data flows for video data (binary format). Both data flows start with data retrieval from either public (Web, YouTube, Twitter, Wikidata, etc.) content providers or from our commercial industry partner Zattoo. As the data items flow downwards through the TVP, the data gets continuously refined into representations and patterns that make up the statistical support which in turn drive the insights as required by the TVP applications, shown at the very bottom of the same figure.

2.2.3 Platform Scalability

The TVP was designed and built with scalability in mind. In the TVP architecture, scalability considerations show in the microservice architecture, in the choice of database systems, and in the distribution of persistence and caching layers throughout the TVP. The system state of the TVP is maintained in the seven database systems that are distributed across the architecture, holding varying levels of data abstractions. The microservices are stateless and can be scaled horizontally as the data flow requires. All components interact via APIs, with all APIs having well-defined functional scopes and a consistent payload abstraction. Where possible, APIs maintain buffers of incoming requests, while processing their workload asynchronously via messaging queues. This design principle applies to the CERTH video segmentation and brand detection services, the CERTH video summarisation service, all WLT ingestion services, as well as the GENISTAT scheduler service for data pre-aggregation from Zattoo.

TVP Operations

The operation of the TVP is de-centralised, with every technical partner of the consortium overseeing all operations of their respective TVP contributions. The partners synchronise on best practices of service operations in the monthly WP4 meetings to guarantee an optimal knowledge transfer on operations, including the orchestration and monitoring of each component. Also, all technical partners of the ReTV consortium provide their services fully containerised through *Docker*, allowing for fully-managed service provisioning via orchestration frameworks such as *Kubernetes* or *Portainer*. *Load Balancing* and *Service Discovery* are used both on microservice and database level where fitting.

Software choices for persistence layers were made in favour of high-availability systems that are distributed across large hardware setups with high resilience to failure. In particular, at consortium partner WLT, we employ large Linux-based hardware clusters to run distributed, horizontally sharded CockroachDB[2] and ElasticSearch[3] systems as backbones to the metadata repositories. Similar setups are operated by Genistat for binary repositories.

TVP Monitoring

Multiple levels of monitoring in the TVP exist, each addressing a different class of failures: Hardware monitoring of the Metadata Repository and APIs as well as all prediction services are monitored via the *Open Monitoring Distribution*[4] (OMD) together with its flexible *CheckMK*[5] extension. This monitoring covers all potential hardware failure scenarios and allows for quick recoveries via battle-proven alerting policies. On top of hardware monitoring, software-level monitoring is employed via *Prometheus*[6] and *Logstash*[7], with *Grafana*[8] and *Kibana*[9] analytical front ends for detailed insights and alerting. Front end application monitoring is achieved via *Sentry*[10] application level monitoring as we as *Logstash/Kibana* for usability and access monitoring.

[2] https://www.cockroachlabs.com/
[3] https://www.elastic.co/
[4] https://omdistro.org/
[5] https://checkmk.com/
[6] https://prometheus.io/
[7] https://www.elastic.co/products/logstash
[8] https://grafana.com/
[9] https://www.elastic.co/products/kibana
[10] https://sentry.io/welcome/

2.3 TVP COMPONENTS

Name	Description	Lead
Semantic Knowledge Base	An RDF triple store (Apache Fuseki) containing entities and their descriptions as a knowledge graph. Includes the event entities from the event extraction service.	MOD
Metadata Repository	A distributed, high volume, high performance persistence layer for fully integrated, cross-vector metadata items.	WLT
Metadata Search Engine	A fast and expressive ElasticSearch index with a rich set of functional API abstractions.	WLT
Audience Metrics	Audience data from Zattoo. Consists of aggregated real-time data and detailed viewing sessions. Metrics are anonymised to preserve user privacy. Data is being pushed regularly into the Metadata Search Engine.	GENISTAT
EPG Repository	Program metadata for the relevant TV channels, regularly pushed into the Metadata Search Engine.	GENISTAT
Binary Repository	A distributed storage solution that holds video files for 10 TV channels since the beginning of 2019. Provides a REST API for search and retrieval of binary content.	GENISTAT
Video Feature Storage	The features that CERTH video analysis service extracts from video are stored in a database for quick later retrieval without the need for recomputation.	GENISTAT

Table 1: Overview of storage components of the TVP architecture.

Similarly, Table 2 lists all eight microservices in the TVP, describing their respective inputs and outputs, as well as denoting their respective development lead from the consortium.

Name	Input	Output	Lead
Temporal Annotation	textual content from Web pages or social media posts	detected temporal references (position of date-time values)	MOD
Event Extraction	event sources: WikiData, iCal files	event instances and their descriptions in RDF	MOD
Crawler	list of Websites or social media channels or (regexp) terms	documents each representing one content item (Webpage, social network post) and its description	MOD

Prediction Service	events, documents, time series success metrics from the documents, audience metrics	prediction of a value at a future time (success metric for a topic on a certain vector, topic for a success metric on a certain vector)	MOD
Video Fragmentation and Annotation	videos	video keyframes, video temporal segmentation information, concept annotations on keyframes, various supporting signals for subsequent components	CERTH
Brand Detection	video keyframes	brand/channel logo annotations on keyframes	CERTH
Recommendation and Scheduling	video segments from the binary repository, video metadata features from the video feature storage, topic predictions from the prediction service	a linear sequence of potentially adapted video segments tailored to the personal preference of the viewer	GENISTAT
Video Adaptation and Repurposing	features extracted by the Video Fragmentation and Annotation component	summary script (i.e. JSON structured document which lists segments of the original video that constitute a meaningful summary), or also the summary rendered to video files of various formats (mp4, webm and gif)	CERTH

Table 2: Overview of enrichment components of the TVP architecture.

Across the consortium, the TVP is comprised of seven database components and eight microservice components. All services provide RESTful APIs as an interface for data exchange. Within the consortium, we maintain documentation of all services in the shared GitLab instance we operate for technical communication between project partners. Table 1 gives a short overview of all storage components from the TVP, also denoting their respective development lead from the consortium.

2.3.1 Content Aggregation and Alignment

WP1 deals with the retrieval of video content and metadata context from a wide set of commercial and public resources. From industry partner *Zattoo*, we retrieve Electronic Program Guides (EPG) as metadata enrichments where available. This correlation of metadata to binary content via EPGs content contextualisation and indirect engagement monitoring through the monitoring of the online debate surrounding the aggregated TV programs both along news and social media vectors.

1. **TV Show content**: the broadcast video content as provided by industry partner Zattoo, together with metadata information extracted from it.
2. **Public metadata content on TV shows**: this includes descriptions and reactions online, as well as the broadcaster's schedule. It is retrieved by the *MOD crawler* from publicly accessible Web sources such as social media APIs and Web sites published by TV producers.
3. **Public metadata content from Europeana**: this relates to the collection of video content from consortium partner NISV that is being employed in the use cases. The content is accessible on the Europeana platform[11], from which the metadata descriptions are aggregated for special use case evaluations.
4. **Event data related to TV shows**: information on what event a TV show relates to. Examples of events are the FIFA World Cup or a royal wedding. This data is aggregated by the *MOD Event Extraction* component in structured format from the public Wikidata repository.

Metadata Aggregation and Alignment

The content aggregation dataflows required by the engagement monitoring scenarios for professional users as described in D5.2 (WP5) are summarized in Figure 2. We collect data from relevant Websites as well as from public social media platforms that are related to the EPG content as provided by the *scheduler* component. Details on this content aggregation can be found in WP1 Deliverable D1.2, *Data Ingestion, Analysis and Annotation*.

WP1 Component Update. Aside from document-based content, we also collect relevant events from the public Wikidata platform to drive the prediction components further downstream in the TVP. Details on event retrieval can be found in WP2 Deliverable D2.2, *Metrics-based Success Factors and Predictive Analytics, First Version*.

The *MOD Crawler* scrapes different data sources for information relevant to ReTV. Its targets are social media, websites and archives and it can be configured to listen for certain keywords through "term lists" or monitor for new content on Websites or social network accounts through "channel lists". It can, for example, listen to keywords or hashtags like *#Tatort*, which usually accompany posts on the popular German-language crime TV show. It analyses and annotates all Web and social media content found matching its term and channel lists and writes this content metadata into the *Metadata Repository*.

[11] https://www.europeana.eu/portal/en

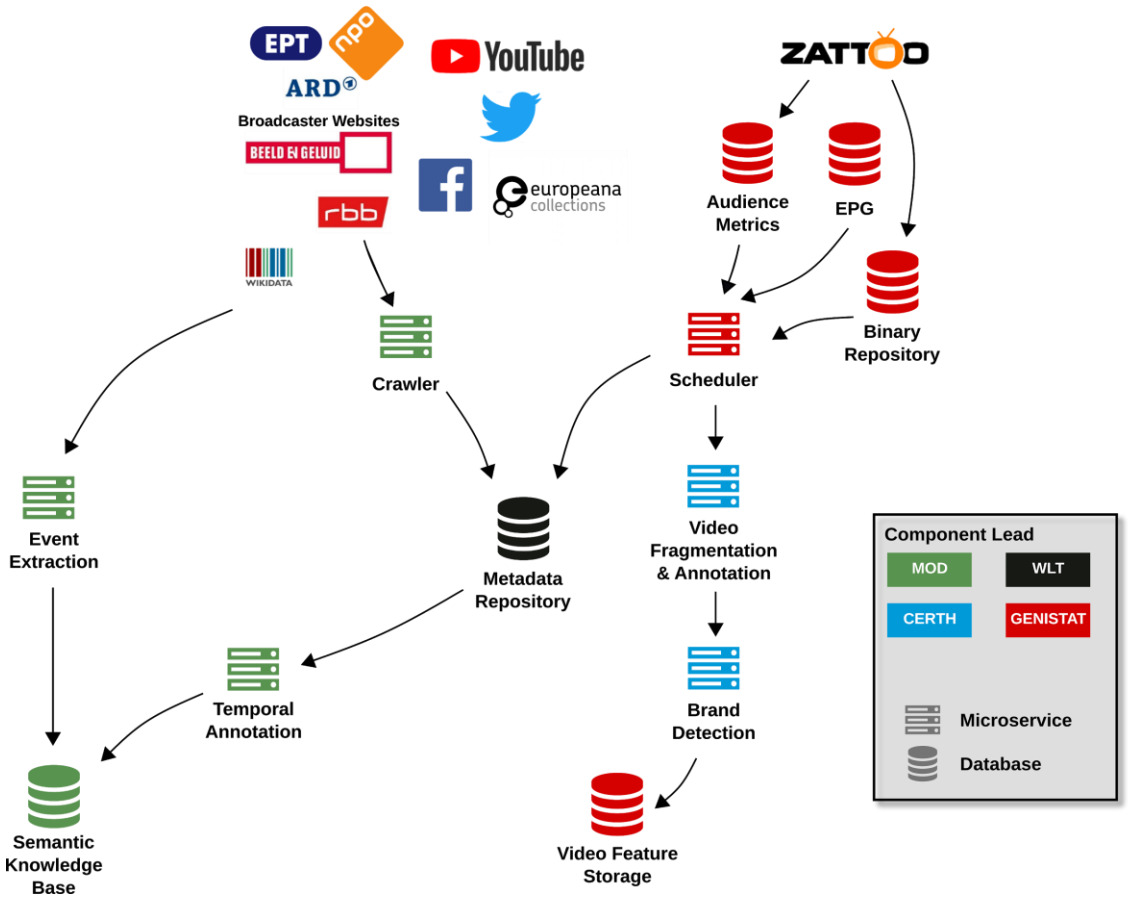


Figure 2: Content Aggregation for Engagement Monitoring (Topics Compass, Content Wizard), from data source (top) to the internal TVP components *Metadata Repository* and *Semantic Knowledge Base*.

Another source of content is the Electronic Program Guide (EPG) data that Genistat receives from Zattoo (stored in the *EPG* database). After is is enriched with metadata and start/end times have been added, the *Scheduler* pushes it into the *Metadata Repository* at regular intervals. Consortium partner Genistat also provides audience statistics on selected EPG publications to enhance the analytical capabilities of the prediction components of the TVP.

Binary Aggregation and Alignment

The source for the raw video material is Zattoo. It is stored in the *Binary Repository.* In compliance with copyright laws, this video data is treated as a private recording and is therefore not redistributed without the explicit approval of the copyright holder.

The *Video and Fragmentation and Annotation* component splits the videos into meaningful temporal fragments, selects some representative frames (keyframes) for each fragment, annotates the keyframes with concepts and extracts a set of additional auxiliary features to support the services of WP3. The *Brand Detection* component extracts information on brand logos detected in keyframes. It also performs advertisement detection. Together, the metadata results from the *Video and Fragmentation and Annotation* component are stored in the *Feature Storage* of ReTV (they include temporal segmentation data, concept annotations and other supporting signals). At this point, it is still unclear if the results from the feature storage component should eventually be integrated with the metadata repository (denoted by the dotted line in Figure 2, between the Metadata Repository and the Video Feature Storage components). The latest status update of WP1 components is compiled in Appendix B.

2.3.2 Content Enhancement and Predictive Analytics

The second major subsystem within the TVP is formed by those components that enable the predictive scenarios of the ReTV use cases (c.f. WP2). They incorporate the database systems *Semantic Knowledge Base* and *Search Index*, as well as the microservice component *Prediction Service*. This prediction subsystem of the TVP is highlighted in Figure 3. The *Semantic Knowledge Base* (SKB) stores all the information on relevant entities for the syntactic and semantic annotation of the crawled content. As a knowledge graph of entities and relations, it supports the accuracy of keyword and entity extraction from the content metadata by modelling synonyms, alternative written forms, lemmas, as well as entity relations (for graph-based disambiguation of entities in MOD's Named Entity Recognition tool *Recognyze*).

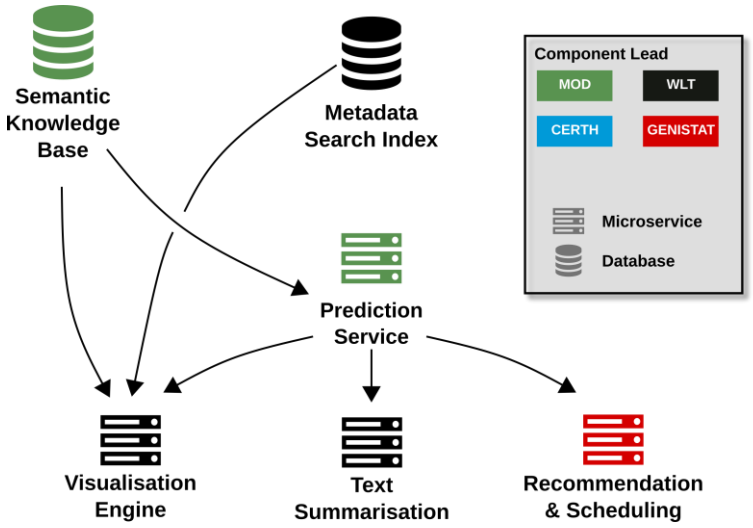


Figure 3: TVP subsystem to enable predictive analytics.

Prediction Service

WP2 delivered D2.2, *Metrics-based Success Factors and Predictive Analytics* in month 20. It has updated on the progress in event extraction and modelling (incl. updating), success metric measurement and audience measurement, and reported on evaluations of prediction models using events and metrics. In month 30 we will deliver the final version of *Metrics-based Success Factors and Predictive Analytics*. A first prototype of the *Prediction Service* incorporates the following prediction models:

1. Prediction of audience metrics based on past audience and including content and event features for improved accuracy. Input: past audience, content categorization and events per TV show/channel; Output: future audience per TV show/channel.
2. Prediction of success metrics for content based on past success metrics. Input: past success metrics per topic; output: future success metric per topic.
3. Prediction of future keyword popularity based on the *Temporal Annotation* component. Input: keyword or topic; output: success metric for that keyword or topic on future dates.
4. Prediction of future events of relevance based on the *Event Extraction* component. Input: future date; output: list of events on that date.

Recommendation and Scheduling

The *Temporal Annotation* and *Event Extraction* component updates the SKB at regular intervals with TV content publication-relevant information from WikiData[12] and other sources such as news articles or public calendars. The SKB is used by the *Prediction Service*. This service additionally takes into account *viewer segments* (created by aggregating groups of users in a manner that does not allow to identify individuals) and metadata on the content and the audience reaction. Based on those pieces of information it predicts the success of content for a particular audience on a particular vector.

The latest status update of WP2 components is compiled in Appendix C.

2.3.3 Recommendation and Scheduling

The *Viewer Segment Creator* uses the metadata on the content and the audience metrics from the *Metadata Repository* to create privacy-safe viewer segments. Those segments are stored in the *Viewer Segment Storage*. The viewer segments, as well as the combined metadata, are used by *Recommendation & Scheduling.* This service needs to decide on the content that should be recommended to viewers and how it should be scheduled. To predict the future success of a piece of content, it also uses the *Prediction Service.* When the *Recommendation & Scheduling* service finds a suitable combination of raw content and publication schedule, it calls *Video Adaptation and Repurposing*.

Video Adaptation

The *Video Adaptation* service fetches the analysis results for videos that were previously ingested by the WP1 services from the *Video Feature Storage* . Then, shorter versions of the original video are generated by employing video summarization methods. This service has undergone several revisions in a continuous effort to be adapted to newly presented challenges and requirements of content partners, as reported in Deliverable 1.2.

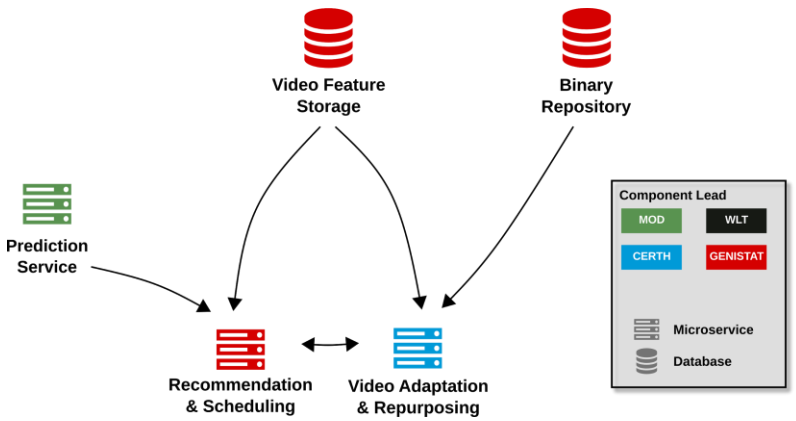


Figure 4: TVP subsystem to enable video adaptation and repurposing.

Text Summarisation

A new component in the TVP since the last WP3 report D3.1, the *Text Summarisation* component is the consortium's answer to stakeholder feedback received during the first half of ReTV's duration: there exists a need for content creators to efficiently optimise the textual content accompanying (re-purposed) video assets to increase its impact on target audiences. In analogy

[12] www.wikidata.org

to what the *Video Adaptation and Repurposing* component applies to binary content, the Text Summarisation component adapts and repurposes a piece of textual content in preparation for dissemination across multiple publication vectors, e.g. press release, blog posting, social media posting, or a long read.

The latest status update of WP3 components is compiled in Appendix D.

3 Platform Integration And Evaluation

In Section 2 of this deliverable we have presented the TVP from a high-level technical perspective as a microservice architecture, interleaving both binary and metadata workflows to uncover content transformations and publication triggers that support content creators for state-of-the art TV experiences. Section 3 discussed the technical integration of the TVP components into a singular platform for content enhancement, prediction and repurposing. A central aspect of component integration lies in the definition of clear interchanges for data exchange between the individual TVP components. In the TVP, the metadata repository with its metadata exchange format defines this very exchange format that is used by all ReTV partners to connect to the TVP.

3.1 Metadata Exchange Format

All TVP components dealing with textual metadata content were specified at the outset of the ReTV project to be compatible with the WLT Metadata Exchange Format. This is a JSON based document description model, specifying the set of required fields as well as the set of optional fields to be expected in metadata items as passed through the TVP. Whereas the model has been originally developed outside of the ReTV project, amendments have been made in order to make it suitable to the TVP data space. The amendments include:

1. an optional field **`media_brand`**.
2. an optional field **`media_broadcaster`**.
3. an optional field **`temporal_start`**, as a UNIX Timestamp.
4. an optional field **`temporal_end`**, as a UNIX Timestamp.

These changes allow for a close alignment of the EPG data with the audience data, along the temporal, the channel and the brand dimensions. The extended metadata model is shown in the sample document *document.json*, as shown in Listing 2. This model can be used in all the metadata API examples to follow in this chapter.

3.2 Metadata APIs

ReTV ensures a high degree of flexibility of the TVP by offering a rich set of *single-purpose APIs*, allowing for a maximum level of combinations for heterogeneous data flows. The metadata repository takes a central role in the TVP, since it integrates all non-binary contents and analytical results into a single persistence service with a rich set of APIs. The metadata APIs contain:

- Authentication API, one-stop token valid for all metadata APIs.
- Ingestion APIs (Document API and Statistics API) for multiple metadata vectors.
- A Search API, for fast aggregations across mult-vector metadata and vast dimensions.
- A Visualisation API, for high-quality, fully interactive visualisations across large, multi-vector metadata.

All metadata APIs are documented online under *api.weblyard.com*, and special programming tutorials are being shared within the consortium for fast access.

```
{
          "content": "Die Kulturstiftung Pro Helvetia soll das Schweizer
Kunstschaffen im Ausland bekannt machen. Sie verhilft hiesigen
Kulturschaffenden nicht nur zu Ausstellungen und Veranstaltungen in aller
Welt, sie gewährt ihnen auch Atelier- und Recherchestipendien auf vier
Kontinenten. \"Kulturplatz\" fragt, wie nachhaltig diese Fördermaßnahmen
sind.",
          "content_type": "text/plain",
          "repository_id": "retv.weblyzard.com/api_retv_epg",
          "uri": "http://api.bee.genistat.ch/program/zattoo/158",
          "title": "Was bringt Kulturaustausch?",
          "meta_data": {
             "published_date": "2018-03-08T16:10:20.335472",
             "language_id": "de",
             "media_brand": "Kulturplatz",
             "media_broadcaster": "sf1",
             "temporal_start": "1520219400",
             "temporal_end": "1520220900",
             "source": "zattoo_program"
          }
}
```

Listing 2: File *document.json*, an example of a GENISTAT EPG document encoded in the WLT metadata format (JSON).

3.2.1 WLT Authentication API

At consortium partner WLT, all API access is restricted to user login via *Java Web Token*. For this purpose, a stand-alone *WLT Authentication API* is employed, with login credentials provided to all ReTV consortium partners. The Authentication API exposes a single GET endpoint for token generation, with the validity of a single token being eight hours. No quota restrictions are used for the ReTV users.

```
#!/bin/bash
TOKEN=$(curl -s -u username:password https://api.weblyzard.com/0.3/token)
```

Listing 3: Creating an access token valid for all webLyzard APIS, with validity of 8 hours.

3.2.2 WLT Search API

Exclusively for ReTV, the WLT Search API has been extended to cater for the various prediction aspects of WP2 and WP3.

At this time, the WLT Search API supports querying for *key entities* (Person, Organisation, GeoLocation, Event) as defined by the MOD *Semantic Knowledge Base* according to the annotations in the WLT *Metadata Repository*. Due to the integration of future timeline projection into the WLT *Metadata Repository*, this query not only supports historic or current date ranges, but also retrieves predicted future mentions of said entity types.

```
curl -H "Authorization: Bearer $TOKEN" -H "Content-Type: application/json"
-X POST
-d @template_request.json https://api.weblyzard.com/1.0/_retv/keyentities
```

Listing 4: Querying the WLT Trending API for key entities using a query template.

```
"query": {
          "bool": {
            "should": [
                  {
                        "text": {
                              "phrase": "Sandmännchen"
                        }
                  },
                  {
                        "text": {
                              "phrase": "Sandmaennchen"
                        }
                  },
                  {
                        "text": {
                              "phrase": "Kalle Kuchenzahn"
                        }
                  },
                  {
                        "text": {
                              "phrase": "Raketenflieger Timmi"
                        }
                  },
                  {
                        "text": {
                              "phrase": "Lennart im Grummeltal"
                        }
                  },
                  {
                        "text": {
                              "phrase": "Pondorondo"
                        }
                  },
                  {
                        "text": {
                              "phrase": "Antjes Fischkoppgeschichten"
                        }
                  },
                  {
                        "text": {
```

```
                                "phrase": "Lola Langohr"
                        }
                },
                {
                        "title": {
                                "phrase": "Sandmännchen"
                        }
                },
                {
                        "title": {
                                "phrase": "Sandmaennchen"
                        }
                }
            ]
        },
    "beginDate": "2020-01-13",
    "endDate": "2020-01-19"
}
```

Listing 5: A JSON-based topic template designed by use case partner RBB to restrict the search space of the contextual WLT *Metadata Repository* to the RBB TV Show *Sandmännchen*, with a time window projected to the third week in January, 2020.

3.2.3 WLT Document API

The WLT Document API provides a simple HTTP wrapper for data ingestion into the WLT metadata repository. It allows for standard CRUD functionality, and has been configured and activated for data ingestion from both the *MOD Crawler* and the *GENISTAT Scheduler* components. The former pushes metadata content as retrieved from various public news, TV and social media sources into the Metadata Repository, whereas the latter pushes EPG program data provided by Genistat into the TVP (c.f. Section 5, *Content Aggregation and Alignment, WP1*). Sample *cURL* requests for the Document API are shown in Listing 6-9.

```
curl -H "Authorization: Bearer $TOKEN" -H "Content-Type: application/json"
-X POST

-d @document.json
https://api.weblyzard.com/0.3/documents/retv.weblyzard.com/api
```

Listing 6: Push (POST) a JSON document to the Metadata Repository.

```
curl -H "Authorization: Bearer $TOKEN" -X GET

https://api.weblyzard.com/0.3/documents/retv.weblyzard.com/api/446202312674
1424383
```

Listing 7: Retrieve (GET) a JSON document to the Metadata Repository.

```
curl -H "Authorization: Bearer $TOKEN" -H "Content-Type: application/json"
-X PUT
```

```
--data @new_document.json
https://api.weblyzard.com/0.3/documents/retv.weblyzard.com/api/446202312674
1424383
```

Listing 8: Update (PUT) a JSON document to the Metadata Repository.

```
curl -H "Authorization: Bearer $TOKEN" -X DELETE
https://api.weblyzard.com/0.3/documents/retv.weblyzard.com/api/446202312674
1424383
```

Listing 9: Remove (DELETE) a JSON document to the Metadata Repository.

3.2.4 WLT Visualization API

The ReTV Topics Compass in its "full" form offers a comprehensive, fully-fledged and feature-rich Data Analytics and Visualisation dashboard customised to the needs of ReTV stakeholders. In order to support use cases that require a more granular approach, the webLyzard Visualization API enables the integration of distinct dashboard components into third-party Web applications. Version 1 uses *<iframe>* tags to embed these components. While this approach ensures ease of use and widespread compatibility across platforms, it also comes with shortcomings if a deeper integration is desired (e.g. incompatibility with certain browser navigations). This will be addressed by additional features in future versions of the API, complementing (but not replacing) the *<iframe>* approach.

Configuration of visualizations (i.e. the exact topics/search terms behind the visualization) are managed by webLyzard and can be defined on request by the use case partners. Access to a configuration for embeddings is achieved via a token in the URL, which tells the webLyzard Visualization API what configuration to apply to the rendered iframe embedding. The required *<token>* is provided by webLyzard and might change over time. This token ensures that the correct search configuration is used for the visualizations.

The iframe should be provided with all necessary attributes:

- *width* - the iframe's width in pixels (int)
- *height* - the iframe's height in pixels (int)[13]
- *src* - URL of the visualization, see also next section "URL Schema" (string)
- *frameborder* - should be "0" to avoid rendering standard iframe border (string) (deprecated in HTML5 and will later be replaced in CSS)
- *scrolling* - should be "no" to avoid scrolling bars (no,yes,auto) (deprecated in HTML5 and will later be replaced in CSS)
- *sandbox* - not required, but if present a value of "allow-same-origin allow-scripts" is required

Geo Map Visualisation

The Geo Map sets documents from the WLT metadata repository within their location of publication as well as within the locations referenced within their content. It allows the user of the *Topics Compass* to determine the geographic scope of their topics of interest quickly and effectively, e.g. to investigate the regional distribution of news articles or social media postings.

[13] Width and height can be preset to the desired dimensions, similar to the approach of e.g. YouTube.

Figure 5 shows an iframe embed of the GeoMap for a given ReTV topic. This embed can be rendered with the HTML embedding shown in Listing 10.

Figure 5: Geomap <iframe> embed for HTML integration of a predefined ReTV Topic.

```
<iframe width=960 height=600 frameborder="0" scrolling="no"
src="https://api.retv.weblyzard.com/embed/ump7lbw1OTInsgfyiuYoClQZFY/geomap
"></iframe>
```

Listing 10: Geomap <iframe> embed for HTML integration of a predefined ReTV Topic.

Tag Cloud Visualisation

The *Tag Cloud* uses dynamic transitions to adapt its content to the current search. It arranges the most relevant keywords alphabetically, uses color coding to show their positive or negative sentiment and adjusts the font size according to their relative importance (= number of occurrences in the list of search results). In Year 2, the color coding of various widgets including the tag cloud has been made more flexible. The component now supports both the drill down and comparison modes, for example to not only visualise positive versus negative sentiment, but also multi-dimensional emotions based on Plutchik's *Wheel of Emotions*. Figure 6 shows an iframe embed of the Tag Cloud for a given ReTV topic. This embed can be rendered with the HTML embedding shown in Listing 11.

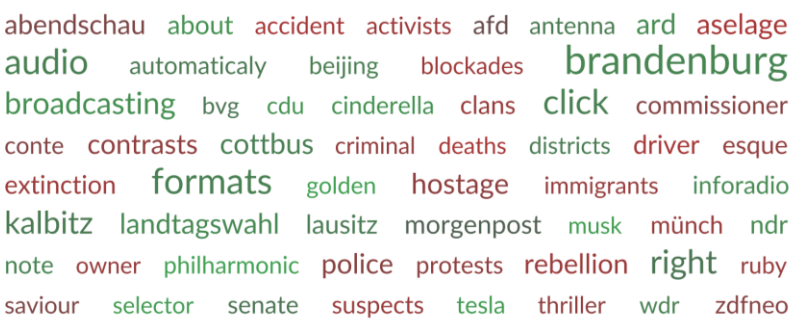


Figure 6: Tag cloud <iframe> embed for HTML integration of a predefined ReTV Topic.

```
<iframe width=340 height=280 frameborder="0" scrolling="no"
src="https://api.retv.weblyzard.com/embed/ump7lbw1OTInsgfyiuYoClQZFY/tagclo
ud"></iframe>
```

Listing 11: Tag cloud <iframe> embed for HTML integration of a predefined ReTV Topic.

Keyword Graph Visualisation

The keyword graph shows the search term together with its strongest *associations* within the search results. The resulting *semantic network* is a powerful tool to guide the development and ongoing adaptation of an organization's communication strategies. The keyword graph allows investigating how different stakeholder groups perceive brands or topics, what negative and positive associations they have with a topic (the color of graph nodes encodes their sentiment), and how the meaning of specific terms changes over time.

Figure 7 shows an iframe embed of the Keyword Graph for a given topic. This embed can be rendered with the HTML embedding shown in Listing 12.

```
<iframe width=600 height=400 frameborder="0" scrolling="no"
src="https://api.retv.weblyzard.com/embed/ump7lbw1OTInsgfyiuYoClQZFY/keywor
d"></iframe>
```

Listing 12: Keyword Graph <iframe> embed for HTML integration of a predefined ReTV Topic.

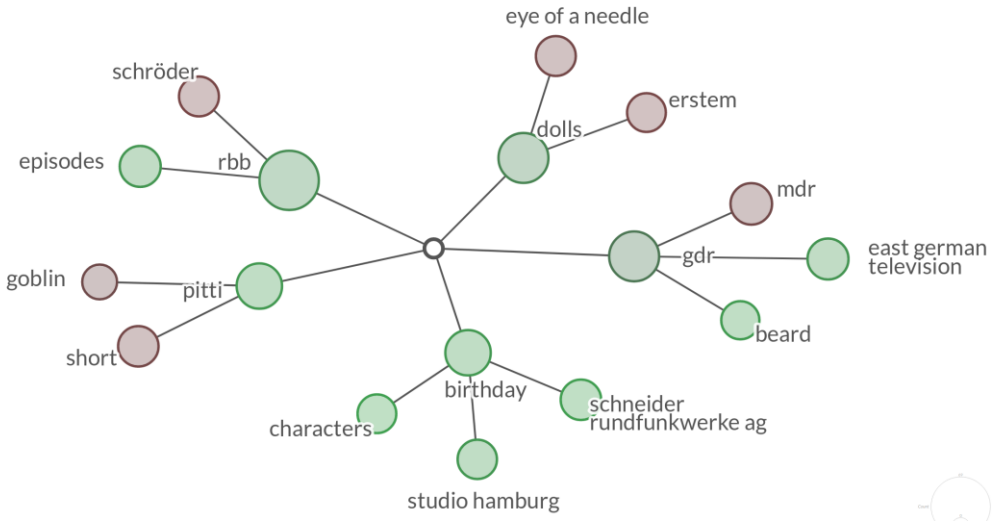


Figure 7: Keyword Graph <iframe> embed for HTML integration of a predefined ReTV Topic.

WYSDOM Visualisation

The webLyzard *Stakeholder Dialog and Opinion Model* (WYSDOM) provides a dynamic assessment that goes beyond sentiment and allows real-time insights into the success of marketing and public outreach activities. Measuring attention and sentiment is descriptive in nature, whereas the hybrid WYSDOM success metric, reveals whether communication targets

have been reached. It measures to what extent the chosen strategy has an impact on observable patterns in online coverage. The success metric also reflects how consistently a message is being conveyed, and whether this message helps to reinforce brand positioning. The WYSDOM trend chart is shown as an embeddable widget in Figure 8.

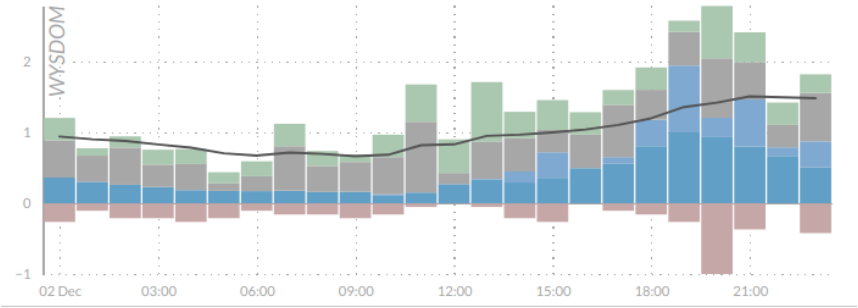


Figure 8: WYSDOM chart <iframe> embed for HTML integration of a predefined ReTV Topic.

3.2.5 WLT Statistics API

The WLT Statistics API provides a simple wrapper to ingest statistical time-series data into a fast ElasticSearch persistence, with full CRUD abstraction. Any content ingested in the WLT Statistical API can be correlated against content from the WLT Metadata Repository, and the resulting insights can in turn be visualised using the WLT Visualisation API. Figure 8 shows an iframe embed of the WYSDOM longitudinal chart for a given ReTV topic. Within ReTV, the WLT Statistics API is used for ingestion of Genistat audience data points into the WLT Metadata Repository. Audience data is shown by the blue bars in Figure 8, correlated with desired (green) and undesired (red) keyword associations as defined by use case partner RBB.

3.3 Prediction Components

The prediction capabilities of the TVP are realised through the prediction components *Prediction Service* and *Semantic Knowledge Base*. Together, these offer a rich set of freely-configurable search queries against time frames lying in the future.

3.3.1 MOD Events API

The Event API grants query access to the events (EventEntities) stored in the MOD *Semantic Knowledge Base* (SKB). It supports different search functionalities, such as a search for events in a given timespan, and an advanced search for events that match given properties with the use of filters.

Property names that can be used with filters are:

- `skbprop:eventCategory`, the general event category, e.g. Sport, NewsEvent, Politics
- `skbprop:confirmation_status`, status of the event, e.g. confirmed
- `skbprop:temporal_start`, starting datetime
- `skbprop:temporal_end`, ending datetime
- `skbprop:location`, location where the event takes place, either exact location, e.g. Emirates Stadium or country, e.g. Bolivia (wd:Q750), the Kingdom of Spain (http://sws.geonames.org/2510769/)

- `skbprop:country`, country code of the location country, e.g. DE, AT
- `skbprop:mdDate`, date value as a string, e.g. 12-10
- `skbprop:year`, year the event takes place as an integer value, e.g. 2017
- `skbrop:next_instance`, for repeating events, such as elections or holidays
- http://weblyzard.com/skb/holidayIn, for holidays, e.g. DE-Augsburg
- `schema:description`, a short description of the event, e.g. 1. Bundesliga, 14. matchday
- `dct:source`, the source, e.g. ical, https://www.timeanddate.com/holidays/belgium (holidays),
- `dct:type`, the event type or subtype, e.g. Premier League (wd:Q9448)
- `dct:date`, calculated (future) dates for repetitive events, e.g. 2063-07-28
- `skos:altLabel`, an optional alternative label, e.g. Christi Himmelfahrt(DE)

Full-Text Search

Full-text search filters need to have `property_format:text` and `filter_operator:contains`. A sample *cURL* request for the full-text search on the MOD Event API is shown in Listing 13.

```
curl -X POST -H "Authorization: Bearer $TOKEN" -H "Content-Type: application/json"
--data '{"filters": [{"property_name":"rdfs:label", "property_value":"wien",
"property_format":"text", "filter_operator":"contains"}]}'
https://api.weblyzard.com/1.0/skb/events
```

Listing 13: Full-text search, returning all events with arbitrary date that contain the query term “wien” in a `rdfs:label` property.

Exact Match Search

Exact match search filters need to have `property_format:null` and `filter_operator:term`. A sample *cURL* request for the exact match search on the MOD Event API is shown in Listing 14.

```
curl -X POST -H "Authorization: Bearer $TOKEN" -H "Content-Type: application/json"
--data '{"filters": [{"property_name":"dct:type", "property_value":"wd:Q16466010",
"property_format":null, "filter_operator":"term"}]}'
https://api.weblyzard.com/1.0/skb/events
```

Listing 14: Exact-match search, search, returning all events of type Association football match.

Integer Search

Integer search allows one to search the MOD Event API by year directly. A sample *cURL* request for the integer year search on the MOD Event API is shown in Listing 15.

```
curl -X POST -H "Authorization: Bearer $TOKEN" -H "Content-Type: application/json"
--data '{"filters": [{"property_name":"skbprop:year", "property_value":"2001",
"property_format":"long", "filter_operator":"lte"}]}'
https://api.weblyzard.com/1.0/skb/events
```

Listing 15: Integer search query, returning all events that happened before and including the year 2001.

Conjunctive Search

Any number of filters can be combined, i.e. the following curl example returns association football matches in March 2019 with a description containing premier league. A sample *cURL* request for the conjunctive search capabilities on the MOD Event API is shown in Listing 16.

```
curl -X POST -H "Authorization: Bearer $TOKEN" -H "Content-Type: application/json"
--data '{"from_date":"2019-03-01", "to_date": "2019-03-31",
"filters": [{"property_name":"dct:type", "property_value":"wd:Q16466010",
"property_format":null, "filter_operator":"term"},
{"property_name":"schema:description", "property_value":"premier league",
"property_format":"text", "filter_operator":"contains"}]}'
https://api.weblyzard.com/1.0/skb/events
```

Listing 16: Conjunctive search query, using a combination of search constraints in conjunction.

Language Filters

There exists an optional filter to only retrieve labels and descriptions in a certain language. In that case the properties rdfs:label, schema:description and skos:altLabel are only returned in that language, instead of a list with all known labels and descriptions. Labels and descriptions that with unknown language are always returned. A sample *cURL* request for the language filters for the MOD Event API is shown in Listing 17.

```
curl -X POST -H "Authorization: Bearer $TOKEN" -H "Content-Type: application/json"
--data '{"language": "de", "filters": [{"property_name":"rdfs:label",
"property_value":"Weltmeisterschaft", "property_format":"text",
"filter_operator":"contains"}]}'
https://api.weblyzard.com/1.0/skb/events
```

Listing 17: Query with output language specified to be German.

Provenance Filters

An optional filter allows to retrieve events of a certain source (provenance). This is most useful to filter events that originate from wikidata from other events extracted through, for example, *iCal*. Provenance uses substring match, so using `provenance:wiki` will work as well. Similarly it is also possible to filter for e.g. *HTTP*. A sample *cURL* request for the language filters for the MOD Event API is shown in Listing 18.

```
curl -X POST -H "Authorization: Bearer $TOKEN" -H "Content-Type: application/json"
'{"provenance": "wiki", "filters": [{"property_name":"skbprop:eventCategory",
"property_value":"Politics", "property_format":null, "filter_operator":"term"}]}'
https://api.weblyzard.com/1.0/skb/events
```

Listing 18: Provenance query, returning only events in the category *Politics*, with provenance *wikidata*.

3.3.2 MOD Anniversaries API

Querying the *Semantic Knowledge Base (SKB)* for anniversaries, i.e. events and significant occurrences that take place on a given day, where the day is provided as a month-day pair. The

response returns events such as birthdays, dates of death, inception dates of organizations, holidays or sporting events that match that day. A sample *cURL* request for the anniversary search on the MOD Anniversaries API is shown in Listing 19.

```
curl -X POST -H "Authorization: Bearer $TOKEN" -H "Content-Type: application/json"
'{ "date": "04-01",
  "entity_type": "PersonEntity"}'
https://api.weblyzard.com/1.0/skb/anniversaries
```

Listing 19: Anniversaries query, returning all events that represent an anniversary happening on 01 April.

3.3.3 MOD Temporal Annotation API

The *Temporal Annotation API* from ReTV partner MODUL allows to identify and annotate mentions of date, both relative and absolute, within a document. This component constitutes one of the core steps that enable prediction. A sample *cURL* request for the text annotation with temporal information via the MOD Annotation API is shown in Listing 20.

```
curl -H "Authorization: Bearer $TOKEN" -H "Content-Type: application/json" -X POST
-d @document.json https://api.weblyzard.com/0.3/annotate/temporal
```

Listing 20: Annotate (POST) a JSON document with temporal information for prediction.

3.4 Current Status And Roadmap

Table 3 summarises the achievements of the second reporting period for WP4. Please note that results achieved and reported in Deliverable D4.1 are kept for completeness.

Component	Current version	Final version
REST API Framework	An updated version of the REST API framework, compatible with changes and new developments of the technical WPs 1-3 and supporting predictive analytics.	*D4.3.* Optimized system architecture in terms of throughput and response time, tested in concrete use case scenarios. Final API specification.
Visualisation Components	Integration of the Genistat audience metrics into they WYSDOM charts; Extension of all charts to support multi-colour codings as provided by the sentiment extension *Plutchik's Wheel of Emotions;* Extension of all charts to support *prediction mode*.	*D4.3.* Ongoing refinement of the dashboard based on the feedback of use case partners in the Topics Compass scenario, with a special focus on the new prediction mode and the multidimensional display of emotions; addition of new PDF report types, e.g. to compare multiple markers selected by the user, or a dedicated WYSDOM report.

TVP Visual Dashboard	Support of predictive analytics by offering a selector to use either *published dates* or *referenced dates* to anchor the documents contained in the search results along the temporal axis. The referenced dates will enable the display of predicted topics, as they can point to both past and future events. User interaction design has been improved based on feedback from the use cases.	*D4.3.* Revised versions of all the embedded visual tools. The dashboard will support cross-lingual exploration and visualisation of content streams across *vectors*, *languages* (English, Dutch, German), and other *context dimensions*. Advanced interactive controls will support drill down operations and on-the-fly query refinements.

Table 3: Overview of updates for WP4 components since D4.1.

3.5 SOFTWARE QUALITY

Software quality management is overseen by WP8 (c.f. D8.1, *Quality and Data Management Plan*). WPs 1-3 maintain their own software quality management in accordance to these recommended software quality guidelines, and these are reported respectively in D1.2, D2.2, and D3.2.

With regards to software quality, all work accomplished in WP4 is following similar recommendations on software quality. All software produced in this work package adheres to:

- Strict implementation of issue tracking and code review.
- Test-driven design principles. To achieve best test coverage of software at unit-level, all developers are trained and reminded to adhere to state-of-the-art principles from Test-Driven Development (TDD).
- Latest development infrastructures. Programming languages in use are constantly updated to their latest stable release versions (Python3.7, Java11) to optimise security, stability, features, and support. Build, test, and development infrastructures are kept up-to-date for similar reasons.
- Static code analysis tools run both on Java and Python environments to counter software defects and code smells as early as possible (Sonarqube, Sentry).
- Continuous Delivery allows for high volume of release cycles, establishes a sense of software ownership.
- Detection of integration defects through full-fledged staging environment.

For integration testing, we define a set of integration tests in the next subsection 3.6, *Platform Evaluation*. While this evaluates the live TVP, similar tests are performed in staging setups, where possible. Note that due to the distributed nature of the TVP, not all integration tests can be run isolated in staging environments.

3.6 PLATFORM EVALUATION

The TVP is a complex and distributed software architecture that integrates a wide range of service components and database repositories from multiple technical partners, tied together via data exchanges and abstractions. For such complex architecture to work both in evaluation as well under load in the field, we are taking multiple measures to guarantee the platform's correctness and performance.

We evaluate the TVP via integration and scalability testing and individual TVP components are evaluated via unit testing and software quality assessments.

3.6.1 Integration Testing

Success criteria for the correctness and completeness of the TVP is asserted via the four use case scenarios described in WPs 5 and 6.

Data Retrieval Subsystem

Since data retrieval drives all of the TVP platform, it is of utmost importance to guarantee a steady flow of data on the binary streams as well as on the metadata streams. In WP4, we have established tight data quality monitoring and alerting on the storage tier following the data ingestion. This allows us to assert the correctness of the integration of the complete data retrieval subsystem from data intake to persistence.

The monitoring system is designed to detect anomalies in any data intake component within at most an hour, and to alert the responsible service operator to guarantee small response times on errors. Table 4 shows that all four data retrieval components are working in a satisfactory manner.

Data Source	Component	Status	Outage/Week
WikiData Events	Temporal Annotation	Running	Low
Websites/Social Media APIs	Crawler	Running	Low
EPG	Scheduler	Running	Low
Audience Metrics	Scheduler	Running	Low

Table 4: Operational overview of the data retrieval subsystem of the TVP. *Low Outage*: less than 1 hour of downtime per week; *Medium Outage*: between 1 and 3 hours of downtime per week; *High Outage*: above 3 hours of downtime per week.

Prediction Subsystem

The prediction subsystem requires the availability of data inputs for the predictive analytics training and calculation from the SKB and the Metadata Search Index as well as the data output's active connection to consuming components for the scenarios (the Visualisation Engine for the Topics Compass or Content Wizard, Text Summarization for the Content Wizard or the Recommendation & Scheduling for the Content Wizard, 4u2 Chatbot or Content sWitch).

1. Prediction of audience metrics uses audience metrics from the Metadata Search Index and events from the Semantic Knowledge Base, outputs to the Visualisation Engine and the Recommendation & Scheduling components. Correctness of operation is shown by response to queries from these components (e.g. channel or show + future time period) with predicted audience.

2. Prediction of success metrics uses success metrics from the Metadata Search Index and outputs to the Visualisation Engine, the Text Summarization and the Recommendation & Scheduling components. Correctness of operation is shown by response to queries from these components (topic + success metric + future time period) with predicted success metric value.

3. Prediction of future keyword popularity uses the *Temporal Annotation* through the SKB and outputs to the Visualisation Engine, the Text Summarization and the Recommendation & Scheduling components. Correctness of operation is shown by response to queries from these components (future time period, optional keywords) with predicted keyword popularity at that future time.

4. Prediction of future events of relevance uses the *Event Extraction* via the SKB and outputs to the Text Summarization component. Correctness of operation is shown by response to queries from these components (future time period, optional topics of interest) with future events in that time period.

Integration testing of the TVP prediction configuration is planned to commence with M24 and run for the remaining time of the ReTV project. Results of the integration testing are projected to be reported in D4.3, after the final iteration of the TVP.

3.6.2 Scalability Testing

Another important aspect of the TVP's evaluation concerns the non-functional assessment of its scaling out capabilities. Under heavy-load scenarios (much concurrent content to be processed in parallel), the TVP should be capable to be updated both by hardware and software nodes to retain operation with data loss, system failure, or negative impact on the usability of any of the ReTV applications.

For scalability, we treat the three functional TVP layers *listening*, *prediction*, and *adaptation* as separate entities, and plan a separate scalability evaluation for each subsystem. A sudden peak in any of the three subsystems will remain isolated to the subsystem, without affecting the other subsystems' operations. In particular, we foresee to answer the following three questions regarding scalability planning of the TVP:

1. What is the monetary cost of doubling the capacity of a subsystem of the TVP?
2. What is the operational cost of doubling the capacity of a subsystem of the TVP?
3. What are the bottlenecks of each subsystem, and how can these bottlenecks effectively eliminated/reduced in case of peaks?

Scalability testing is planned to commence with M24 and run for the remaining time of the ReTV project. Results of the scalability testing are projected to be reported in D4.3, after the final iteration of the TVP.

4 TVP FOR PROFESSIONAL USER SCENARIOS

Since Deliverable D4.1, where the first prototype of the TVP was outlined, the use cases have furthered through recurrent internal and external stakeholder feedback loops into professional user applications with clear focus on addressing the market needs voiced from the use case scenarios. This section provides a brief overview of ReTV's two use case applications for the professional user, with a particular focus on how they are technically enabled through the TVP as outlined in the previous sections.

4.1 RETV TOPICS COMPASS

The *ReTV Topics Compass* is the flagship version of a Data Exploration and Visualisation System used for the TV content and news metadata and its analyses within the TVP. As such, the *Topics Compass* makes sole use of the metadata analysis workflow in the TVP, and it is the driving vehicle for the definition and configuration of all content parameters used in the metadata

visualisations that are shared via API with the *Content Wizard*. Whereas the pre-tailored visualisation components as integrated into the *Content Wizard* via the WLT Visualisation API are limited in their exploratory functionalities, the *Topics Compass* allows for full-fledged, multi-modal context explorations over arbitrary topics, date ranges, and across all five ReTV languages.

The advantages of the *Topics Compass* over the integrated visualisation in the *Content Wizard* are its exploratory capabilities, supporting drill down operations on specific aspects of a story. The context of the story is shown within the public debate - by associated keywords, observed agreement and disagreement, as well as entity and geolocation co-occurrence patterns.

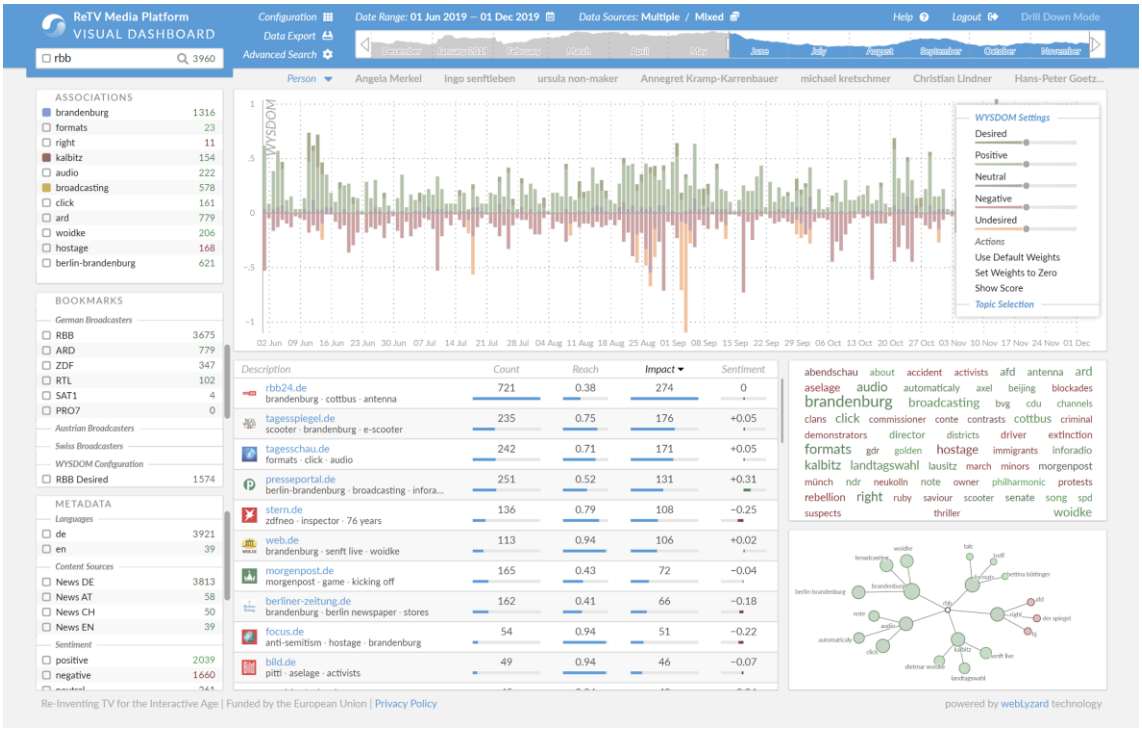


Figure 9: The ReTV Topics Compass, one of the TVP applications, highlighting how the integration of the TVP components enables each TVP application.

The ReTV *Topics Compass* is a JavaScript-based Dashboard implementation that is directly connected to the WLT Search Index (ElasticSearch), providing full access to the WLT Metadata Repository as well as the MOD Prediction Service, as highlighted in Figure 9.

4.1.1 TVP Integration

The *Topics Compass* is the front end system to the metadata repository, abstracted by the metadata search index for fast aggregations. It uses the visualisation engine for rendering of the analytical charts. Also, the SKB provides entity-related metadata for the visualisation engine in the form of entity thumbnails and internationalised label and descriptions. Another integration for the *Topics Compass* comes from the textual summarisation and prediction services. A schematic overview of TVP integration for the *Topics Compass* is shown in Figure 10.

The Topics Compass benefits from the TVP as follows:

- **Prediction Mode**, the new prediction capabilities of the TVP allow for the rich analytical visualisation to be projected into future time frames. This opens completely new doors to the analytical capacity of the Topics Compass.

- **Text Summarisation**, the Topics Compass makes use of the Text Summarisation service to better abstract news article content into relevant content.

- **Semantic Knowledge Base**, with the integration of the SKB into the Visualisation Engine, the Topics Compass gains the possibility of a whole new level of metadata overlays from the LOD space to elevate its keyword-centric visualisations into real-world entities.

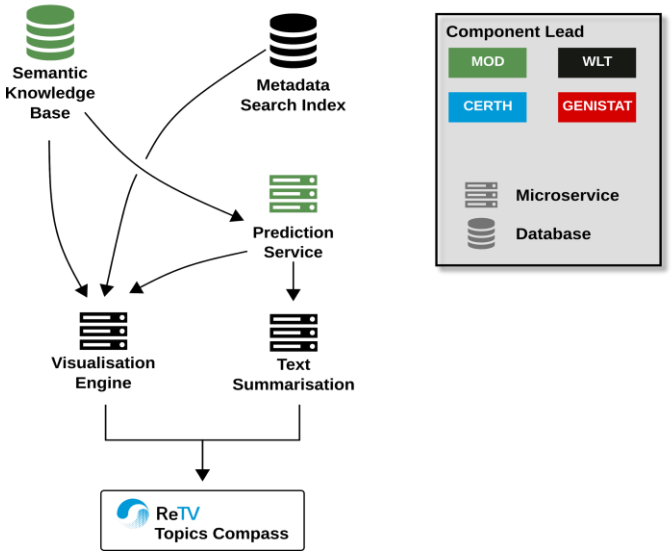


Figure 10: The TVP configuration of the ReTV Topics Compass.

4.1.2 Year 2 Innovations

Since development of the *Topics Compass* is covered by Task T4.3, *TVP Visual Dashboard*, we report on the updates and innovations of the *Topics Compass* in Year 2 of the ReTV project:

- **WYSDOM Chart.** Activating the hybrid success metrics now opens a sidebar category element with the configuration dimensions as selectable markers (specific selections not only impact the WYSDOM chart, but the dashboard's other visualizations as well). The new version provides higher granularity, i.e. hourly instead of daily values, and provides access to statistical indicators as additional dimensions. As long as the WYSDOM mode is active, markers of other sidebar elements are not available. The latest TVP release also supports the display of audience metrics for specific programs and channels (see Figure 11).

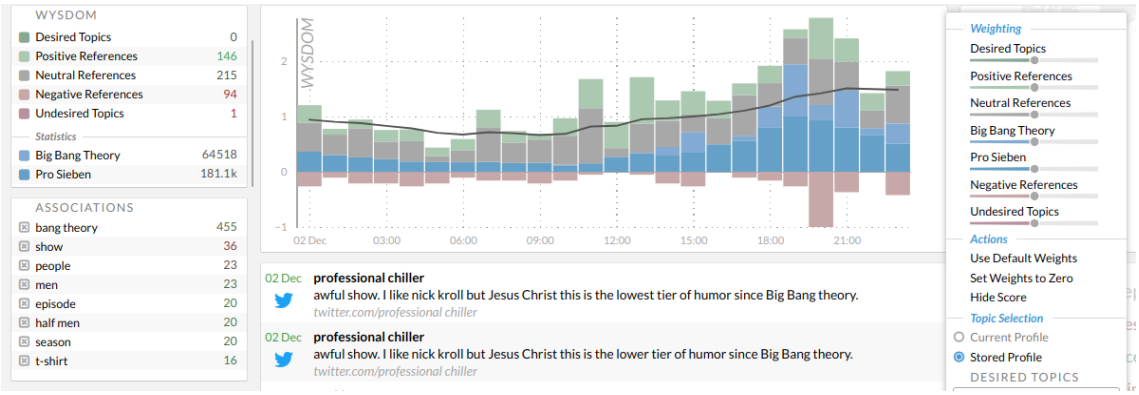


Figure 11: Audience metric integration for programs ("Big Bang Theory") as well as channels ("Pro Sieben") into the hybrid WYSDOM chart.

- **Prediction Mode.** The dashboard configuration menu has been extended with a "date processing" selector to switch between regular mode and the new *prediction mode*. The latter uses date references found in the documents instead of publication dates to project documents along the temporal axis in the trend chart, unlocking the ability to apply the full analytic capabilities of the dashboard to future topics.

- **Multi-Color Mode.** Alternative color scheme based on the selected topics or metadata attributes, visible in the portal mode "Comparison"; including stacked horizontal bar chart to reflect result composition in the source and entity lists. This allows to render benchmarks of channels and programs, metadata classifications as well as the multi-dimensional emotional categories of WP2, e.g. *Plutchik's Wheel of Emotions*.

- **Charting Library.** Two floating menu options have been added: (i) data granularity in terms of days, minutes or hours, including "automated" that adapts to the selected time interval; (ii) impact of a selected topic, similar to frequency but weighing each occurrence by the reach of the source (we have also replaced frequency with impact as the default sorting attribute in the source list to pinpoint opinion leaders).

- **Story Detection and Visualisation.** Revised algorithm to rank documents within a given story, increasing performance and addressing a few minor bugs in terms of document assignment and labeling. The embedded document links now provide direct access to the original content.

Search Results Structure. We redesigned the Topics Compass and introduced a hierarchical layout with five main categories – Documents, Sentences, Sources, Entities and Relations. Typically, each of these categories contains at least one list view and one visualization (e.g. word tree for sentences or scatter plots for entities and sources).

- **Sidebar Structure.** The look and feel of the left sidebar has been aligned with the visual tools of the right sidebar - including support of multiple categories and drag and drop operations. The metadata category has been revised and now includes source country as an additional attribute.

- **SKB Viewer and Editor.** Complementing the dashboard, this separate user interface delivered by MOD (WP1) allows *Topics Compass* users to search for entities in the Elasticsearch index (which is synced with entities in MOD's SKB), supporting filtered search for entity type, time range, anniversary date, as well as property types and property values, and allows to modify thumbnails in addition to property values in multiple languages.

4.2 ReTV Content Wizard

The *ReTV Content Wizard* is the user-facing application that builds upon all aspects of the TVP, integrating results from both the metadata and the binary analyses. As such, it directly or indirectly makes use of the WLT Content Recommendation API, the WLT Search and Visualisation APIs from the metadata workflows, as well the Genistat Recommendation and Scheduling API and the CERTH Video Adaptation and Repurposing API from the binary workflows. The integration of the ReTV Content Wizard into the TVP is shown in Figure 12.

In particular, the following functional aspects are provided by the TVP in the form of microservice API requests from the *Content Wizard*:

- **Text Summarisation**: Smart, textual content creation support and impact optimisation, based on trends from the public debate (i.e. metadata context) and prediction.

- **Visualisation Engine**: rich, interactive trend visualisations and top trending keywords derived from the public debate (i.e. metadata context) and prediction.
- **Recommendation & Scheduling**: Smart video scheduling and adaptation for multi-vector impact optimisation.

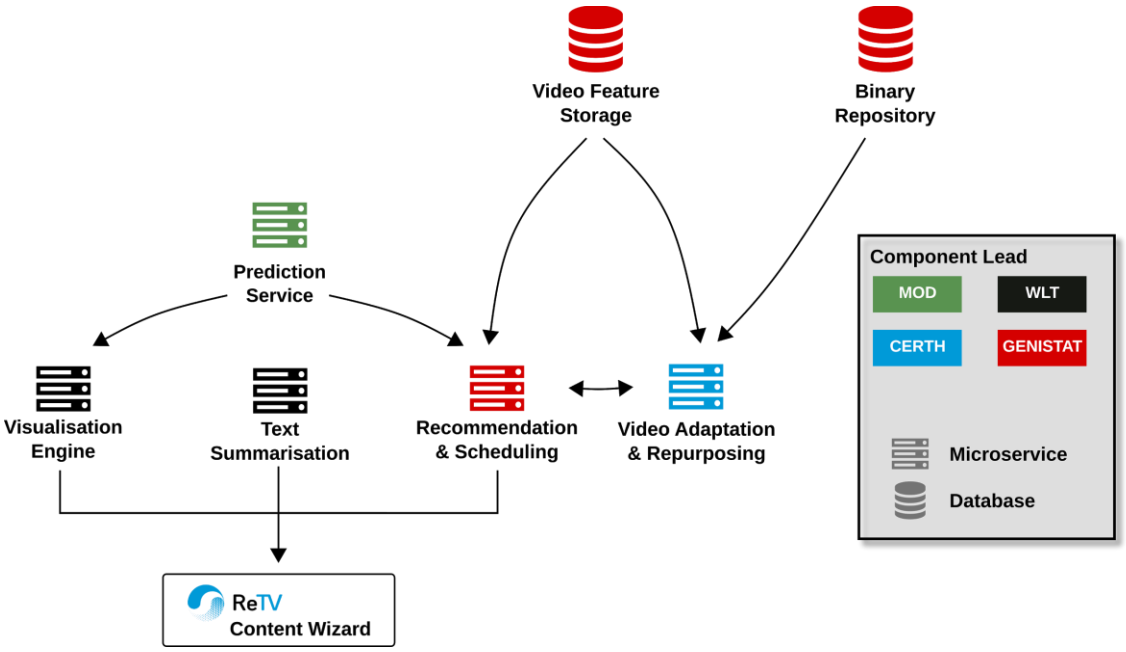


Figure 12: The TVP configuration of the ReTV Content Wizard.

Integration with the WLT Search API

The WLT Search API allows to query the WLT metadata repository comprised of multi-modal television-specific as well as general-interest content collected by MOD in an expressive and efficient way. The API identifies the top stories, their opinion leaders, and the most sought-after keywords occurring in the current debate, but also, thanks to the prediction capabilities provided by partner MODUL, in the near future. This allows content creators such as users of the Content Wizard to guide their idea creation and content refinement process towards more effective publication turnover.

Each search result (stories, opinion leaders, keywords) is queried with precise date range selectors provided in the application to the user's choice. The results are presented as ranked lists *Top Stories*, *Opinion Leaders*, and *Top Keywords*.

Integration with the WLT Visualisation Engine

All visualisations that make up the *ReTV Topics Compass* are available as embeddable widgets via the WLT Visualisation API. In particular, this interface allows for the integration of contextual metadata analytics including prediction into the use case application *Content Wizard*. Requests against this API allow precise configurations regarding the data sources, search queries, as well as data ranges to be defined. In addition, the WLT Visualisation API further broadens the context shown in a visualisation widget via similarity search based on specific URIs. This is useful if the seed of the explorative context analysis is provided by a single document rather than a topic definition or search terms.

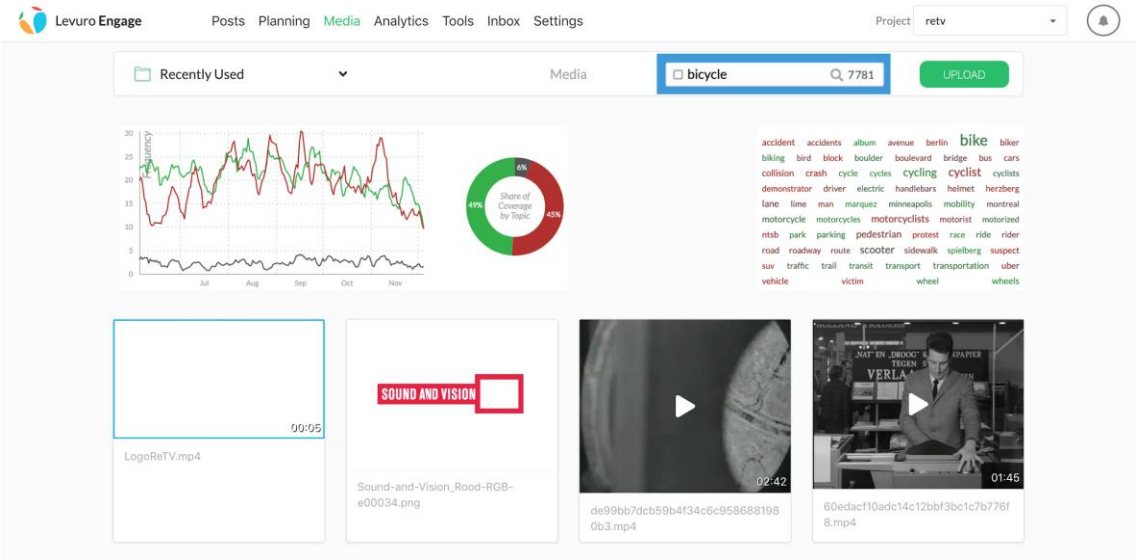


Figure 13: Mockup of the integration of the WLT search into the media assets view of *Content Wizard*.

WLT Text Editor

The WLT Text Editor refers to embeddable UI elements for the text summarisation component that will be used in the Content Wizard. Based on *TinyMCE* editing platform (www.tiny.cloud), it is composed of a stand-alone Web application and a rich set of RESTful Web Services (APIs) for TVP integration. The editor is a content optimisation, text summarization and ambient search tool that helps content creators maximise the impact of their textual content across multiple vectors (press releases, twitter, facebook, blog, etc.) by suggesting lexical, stylistic, and contextual text replacements that can increase the impact of publications. The replacement suggestions are determined by measuring the impact of past publications and observing trending stories as well as keywords and references. The MOD prediction service will provide additional evidence to finetune the data-driven content recommendations. As part of its individual exploitation plan, WLT plans to release a stand-alone version of the editor under the name *StoryPact*.

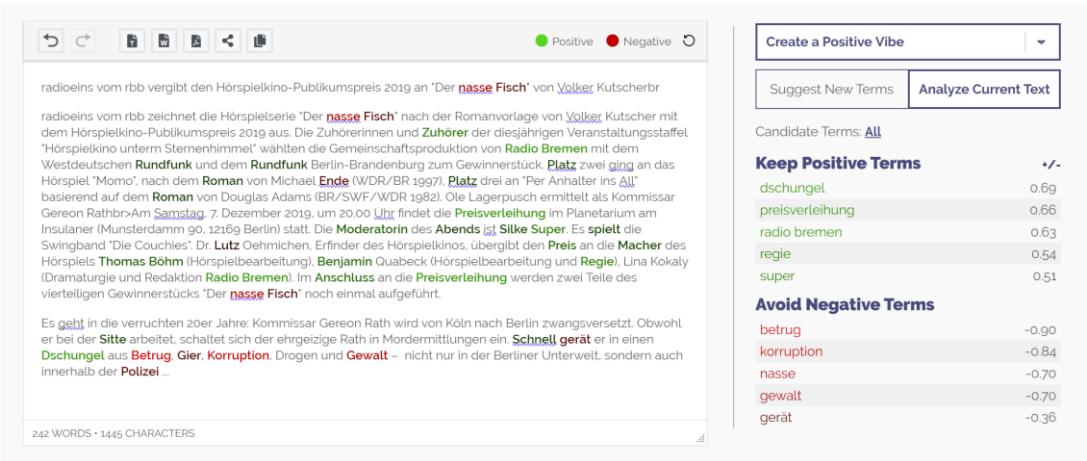


Figure 14: Screenshot of the WLT Text Editor, showing an annotated press release and sentiment analysis to highlight positive and negative terms.

Integration with Video Adaptation & Repurposing

When the user selects a video for editing in the Content Wizard, they are given the option to have it automatically summarised (see Figure 15). The summarisation is then requested from the Video Adaptation & Repurposing service. Since the video analysis is run in advance, storing extracted features in the Video Feature Storage, the summarisation is almost instant. Figure 15 also shows how the video editor cuts the video into the scenes that the Video Adaptation & Repurposing proposed. The user can still make manual adjustments if so desired.

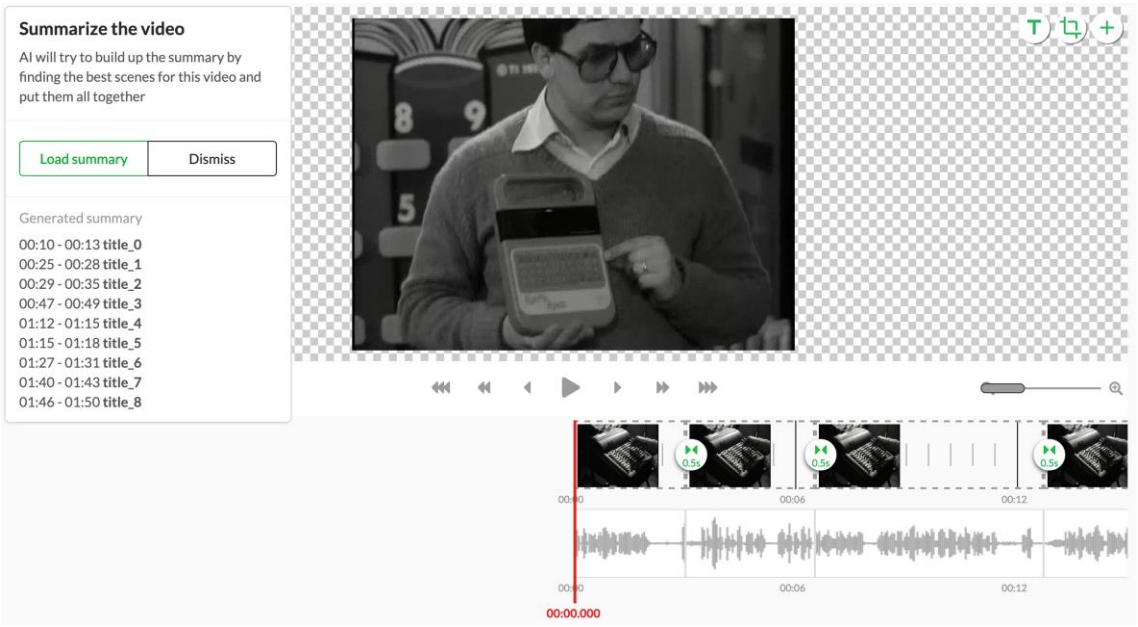


Figure 15: "Load Summary" button for a video in the *Content Wizard*.

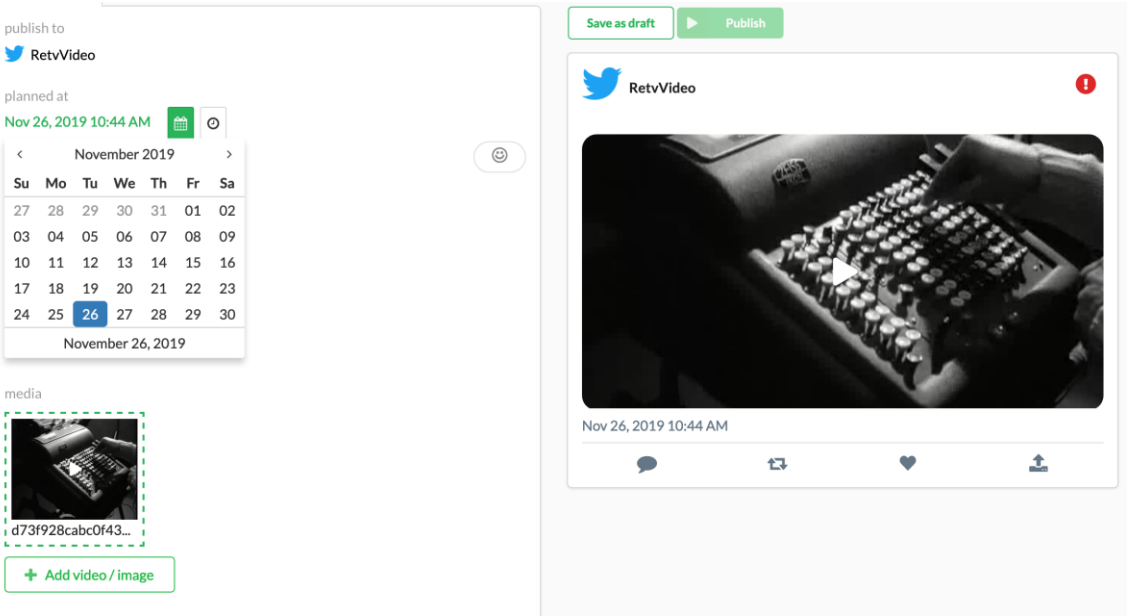


Figure 16: Choosing a publication time for a social media post.

Integration with Recommendation & Scheduling

When publishing a video or any other type of content to social media, the Content Wizard allows the user to schedule the post (see Figure 16). Currently the user needs to set the publishing time themselves. However, the Recommendation & Scheduling API is ready to be integrated into the Content Wizard and set the publication date and time to our algorithms best estimate.

5 TVP For Personalisation Prototypes

Similar to the professional user scenarios, ReTV has also developed specific user scenarios for the prosumer user in the form of personalisation prototypes. In particular, the two use cases *4u2 Chatbot* and *Content sWitch* address the professional consumer (single person content creator and consumer) to optimise their production workflows within a fast-paced environment of vectors and formats.

5.1 4u2 Chatbot

The goal of the 4u2 Chatbot use case is to surface relevant video content to viewers. Based on the results of user research, we decided on chatbots and smart speakers as a publication vector. The chatbot is built using the *Rasa* chatbot framework to abstract from different types of messengers (e.g. Telegram and WhatsApp). We have built two chatbots, one for NISV and one for RBB. They share the majority of their code, but are configured differently. See Figure 17 for the components of the chatbot use case itself and Deliverable D6.2, *First Validation of Personalization Prototype*, for a detailed description of the use case.

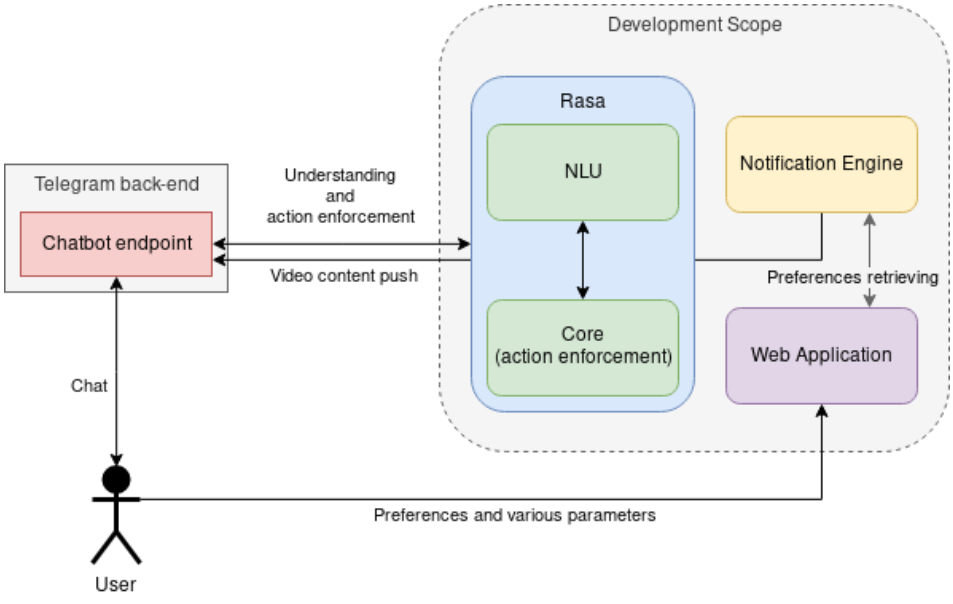


Figure 17: Architecture of the *4u2 Chatbot*.

The chatbots use the Recommendation & Scheduling component to decide which videos to show that which users based on their preferences.

5.2 Content sWitch

The Content sWitch is a ReTV use case scenario that was used to investigate the feasibility and appeal of replacing parts of the video stream. Instead of replacing ads, as is done in Dynamic Ad

Insertion, we insert trailers. See Deliverable D6.2, *First Validation of Personalization Prototype*, for a detailed description of the use-case.

The Content sWitch uses the Recommendation & Scheduling component to decide on the ideal trailer for the person watching. The EPG data from the Metadata repository is used to pinpoint program slots where the trailer can be inserted.

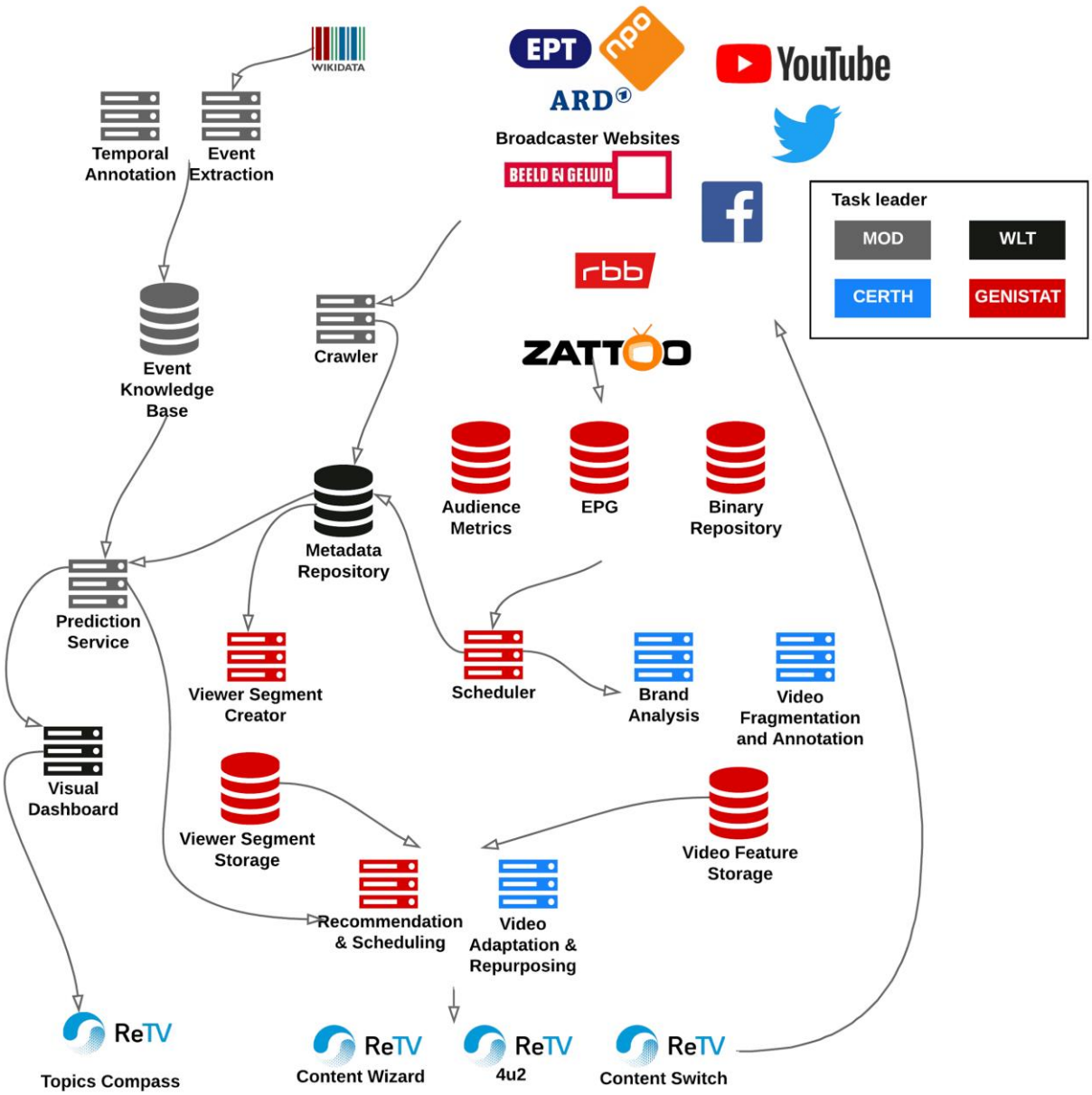
The actual insertion of the trailer into the video stream happens in the browser, by having the video player switch from the stream to the trailer and then back again. In a production setup, an additional back end component that splices the video files would be required.

6 Conclusion And Outlook

This deliverable presents the first major iteration of the TVP a year after its initial prototype release in M12. At M24, the TVP is fully functional with all components of the target architecture being integrated.

For the final year, the focus of WP4 lies in continuation of elevating the TVP from early proof-of-concept configurations into a commercially viable technical platform, with particular focus on high-availability, and rapid scaling out capabilities. To this end, we will continue with the platform evaluation with a stronger focus on the downstream subsystems *Prediction* and *Recommendation/Scheduling*.

APPENDIX A: INITIAL TVP ARCHITECTURE (D4.1)



APPENDIX B: WP1 COMPONENTS UPDATE

Component	Current version	Final version
EPG	The current version loads EPG data from Zattoo and stores it in an internal database of Genistat. The top 10 TV channels in Germany and Switzerland are currently loaded. Those are ARD, ZDF, Prosieben, RTL 2, Sat 1, SRF 1, SRF zwei, VOX, ZDF. Additionally, we also push the EPG data of consortium partner RBB. We decided to focus on those channels, as their popularity means that there will be a significant amount of Zattoo audience data to match to the content.	Operational. Will be extended as needed.
Binary repository	The binary repository is being fed with the the video data from ARD, ZDF, RBB, Prosieben, SRF 1 and SRF zwei.	Completed.
Scheduler	The scheduler is pushing audience data from *Audience Metrics* to *Audience Metrics* and from *EPG* to the *Metadata Repository.*	Completed.
Video Annotation and Fragmentation	The video temporal segmentation to shots/scenes was updated to use a DCNN-based method (inspired by the literature), which is approximately 700% faster than the previously used one. A novel and improved deep learning architecture was designed for concept annotation aiming at more accurate detection by employing a sub-class partitioning strategy during the training phase. The collection of the employed concepts pool was expanded and now includes: a) “YouTube8M”, b) ”ImageNet”, c) “places365”, d) “trecvid SIN task”, e) “Sandmannchen and friends”, a custom concept pool employed after a request from a content partner (RBB) due to the	Increase speed, accuracy and robustness of the component, with changes at both the algorithmic and software implementation levels, to facilitate video analysis.

	inclusion of “Sandmännchen and friends” TV series in the TVP ingestion pipeline. A set of additional auxiliary features to support the services of T3.3 (Content Adaptation and Repurposing) of WP3 are also extracted in this stage. The video’s temporal structure information, the concept-based annotations, and all the additional features extracted are stored in *Video Feature Storage* in order to be readily available for WP3 services.	
Brand Detection	The set of brands that can be detected was extended after discussions with partners. The *Brand Detection* component can currently detect more than 320 logos in video keyframes. The RetinaNet object detection framework was selected to replace the previous used FasterRCNN framework. The decision was made after collecting evaluation results from the literature on standard object detection benchmark datasets. New models utilizing the new framework were trained on the extended set of brands logos. An evaluation conducted by CERTH shows a significant increase both in terms of accuracy and speed. We introduced a first version of an ad-detection framework to distinguish between three classes of TV program: a) content, b) advertisement and c) promo. The detections and the ad-detection results are stored in *Video Feature Storage.*	Brand logos pool will be extended as needed, based on the requirements of WPs 2, 3, 5 and 6. Research will be conducted on further optimizing the ad-detection framework. Research will also be conducted on accurate EPG alignment utilizing other analysis results of WP1 (i.e. channel logo detection and ad-detection).
Crawler	Crawling has continued as planned. An extended list of websites, particularly TV/Radio websites as well as ‘hybrid’ (both news and TV/radio related content), is now supported. Dutch content was added and the Dutch	*D1.3* Data collection is continuing and will be extended or modified as required by the use cases. The annotation pipeline is to be extended to more

	language now supported in the NLP part of the pipeline. Our NER/NEL annotation capability is iteratively improved and remains state of the art, supported by an error classification approach and a visual error correction interface called ORBIS.	accurately support annotations of Works (such as TV programmes). As part of this, iterative extensions of the entity coverage of the Semantic Knowledge Base which is used in the semantic annotation will take place, as well as iterative training and improvements in annotation accuracy.
Semantic Knowledge Base	The knowledge base is populated with the latest dumps of WikiData and OmegaWiki for multiple languages. It is updated as required when entities are found to be missing or incomplete. It is being extended by events daily provided by the *Event Extraction* component. We added the capability to update event descriptions via the same component.	*D1.3* We will look at extending the type and range of entities - especially works and events - collected according to use case requirements. We will release a Web interface to the KB that allows adding, checking and correcting entity descriptions, allowing expert end users to curate available entities.
Video Feature Storage	Features extracted from *Video Annotation and Fragmentation* and *Brand Detection* are stored as binary files to speed up the creation of video summaries.	Completed.

APPENDIX C: WP2 COMPONENTS UPDATE

Component	Current version	Final version
Event Extraction	Events from WikiData and selected iCals are extracted and stored in the *Semantic Knowledge Base*. Event descriptions are also updated from the same sources.	*D2.3* Maintained up to date and including support for extracting events specifically needed by the use cases.
Temporal Annotation	Temporal reference detection in documents is implemented (both absolute and relative references) and available to the *Prediction Component,* currently using a subset of the huge News document corpus in the *Metadata Repository*.	*D2.3* Full coverage of temporal references extracted from past documents across all vectors.
Content-based Success Metrics	Trend detection for the success metrics based on historical audience and viewer data. Improved keyword extraction with a special focus on compound nouns and part-of-speech validity checks. Daily intervals replaced by a shorter timespan granular enough to track the impact of short-term interventions, e.g. changes in advertising strategies.	*D2.3* Success factors calculated based on multiple emotional categories (in addition to bipolar sentiment annotations) and measures of disagreement calculated for arbitrary time intervals and with an on-the-fly reconfiguration of weights.
Genistat Audience Metrics	The Zattoo audience data for Switzerland and Germany is being aggregated into 5-minute slices and then pushed to the *Metadata Repository*	Completed.
Prediction Service	A first version incorporates the following as separate prediction models: 1. prediction of audience metrics based on past audience and including content and event features for improved accuracy 2. prediction based on historical data (i.e., past success metrics) 3. prediction of future keyword popularity based on the *Temporal Annotation* component	*D2.3* A hybrid prediction model based on the most accurately evaluated analytics for each combination of vector and metric. This combined model is retrained dynamically depending on the amount of new data.

	4. prediction of future events of relevance based on the *Event Extraction* component	

APPENDIX D: WP3 COMPONENTS UPDATE

Component	Current version	Final version
Metadata Model	The metadata model is described in *D3.2,* encompassing datasets collected and extracted in WP1 and WP2. The Elasticsearch cluster of WP4 is used to store the data.	Final version already delivered in D3.2.
Recommendation & Scheduling	First deployment and evaluation of repurposing and scheduling on a best effort basis using the *D2.2* version of the *Prediction Service.* Focused on use in the ContentWizard for scheduling and the 4u2 Chatbots to push the ideal content to viewers.	*D3.3* The final version of repurposing and scheduling, using the latest version of the *Prediction Service* deployed for *D2.3.*
Video Adaptation and Repurposing (aka Video Summarization)	The *Video Summarization* (VS) service fetches from the *Video Feature Storage* the analysis results for videos that were previously ingested by the WP1 services. Then, shorter versions of the original video are generated by employing video summarization methods. This service has undergone several revisions in a continuous effort to be adapted to newly presented challenges and requirements of content partners. Research was conducted on a more elaborate learning-based video summarization method that takes into consideration in the temporal relation of frames and already achieves state of the art results on standard benchmark datasets. We also developed an early version of text-based video retrieval for matching media assets with ads, which is a baseline for further improvements.	Improve the learning-based summarization approach by a) introducing editor-specific rules, b) extending it to different summary types (e.g. trailer), and c) adapting summaries to specific target vectors. Integrate the learning-based summarization approach in the VS service. Use text-based video retrieval for matching media assets with ads and integration in ReTV.