



SELMA

Stream Learning for Multilingual Knowledge Transfer

<https://selma-project.eu/>

D1.4 Final Prototype Report

Work Package	1
Responsible Partner	Priberam
Author(s)	Sebastiao Miranda
Contributors	Guntis Barzdins, Andreas Giefer
Reviewer	Andreas Giefer
Version	1.0
Contractual Date	31 March 2024
Delivery Date	28 March 2024
Dissemination Level	Public

Version History

Version	Date	Description
0.1	01/03/2024	Initial Input
0.3	06/03/2024	Internal Review
1.0	26/03/2024	Publishable version

Executive Summary

SELMA's NLP research on transfer learning, user feedback learning and stream learning has been applied into three main use cases, Multilingual Media Monitoring (UC1), Multilingual News Content Production (UC2) and SELMA NLP Service Orchestration (UC0), through the development of different software prototypes.

This document provides an overview of the prototype development related to each of the three main SELMA use cases, including the final status of the implementation of the requirements previously listed in D1.3. For ease of reading, this document consolidates the information presented in the previous two Prototype Reports, D1.2 and D1.3, and serves as the final version of SELMA's Prototype Report.

Table of Contents

- Executive Summary..... 3**
- 1. Introduction 8**
- 2. Multilingual Media Monitoring (UC1)..... 9**
 - 2.1 Named Entity Recognition and Entity Linking9**
 - 2.1.1 User Entity Correction 12
 - 2.2 Multilingual News Clustering 13**
 - 2.3 Multilingual Topic Detection 14**
 - 2.4 Multilingual Multi Document Summarization..... 15**
 - 2.5 A/V Content..... 16**
 - 2.6 Diversity Use Case Application 17**
 - 2.7 Integration of Grounded Large Language Models for Explainability..... 18**
- 3. Multilingual News Content Production (UC2)..... 21**
 - 3.1 plain X user interface 21**
 - 3.2 plain X Audio Wave Form 28**
 - 3.3 plain X Goal Oriented Workflow..... 28**
 - 3.4 Models integrated in plain X 31**
 - 3.4.1 Speech Recognition 32
 - 3.4.2 Speech Translation 34
 - 3.4.3 Custom Voiceover models..... 35
 - 3.4.4 plain X Automations API 36
 - 3.4.5 Other Integrations 37
- 4. SELMA NLP Service Orchestration (UC0) 39**
- 5. External Use Cases: LETA 41**
 - 5.1 PiniTree: Rule-based Stream Learning for NEL..... 41**
 - 5.2 Description of the NEL Stream Learning process in the PiniTree Editor 41**
 - 5.3 Word Sense Disambiguation in the PiniTree Editor..... 44**
- 6. External Use Cases: Podcast Producer..... 45**

6.1	Goal	45
6.2	Motivation	45
6.3	Production Steps	45
6.4	The structure of a typical news podcast	46
6.5	Developing a stand-alone app	47
6.6	Using SELMA components to increase efficiency	47
6.7	The Podcast Creator App in Screenshots	48
6.8	Interaction with SELMA components	51
6.9	Breaking out sub-functionalities into separate apps	52
7.	<i>Research Prototypes</i>	55
7.1	M-PHANTOM Transcription Prototype	55
7.2	Post Editing Prototype	57
7.3	Story Segmentation Prototype	58
7.4	Speaker Identification Prototype	59
7.5	Diarization Prototype	60
8.	<i>Requirement Implementation Status</i>	62
9.	<i>Conclusion</i>	86
	<i>Bibliography</i>	87
	<i>Annex 1 – plain X User Guide</i>	88
	<i>Annex 2 – plain X Automations API</i>	89

Table of Figures

FIGURE 1 NAMED ENTITIES (LINKED TO WIKIPEDIA) IN THE MONITIO DOCUMENT PAGE, AS DETECTED BY THE NAMED ENTITY RECOGNITION AND LINKING MODEL DEVELOPED WITHIN SELMA 10

FIGURE 2 NAMED ENTITIES (LINKED TO WIKIPEDIA), AS DETECTED BY THE NAMED ENTITY RECOGNITION AND LINKING MODEL DEVELOPED WITHIN SELMA, TRENDING IN A DATE RANGE OF 30 DAYS. ENTITIES ARE CONSIDERED TO BE “TRENDING” ACCORDING TO A “POINTWISE MUTUAL INFORMATION” CRITERION ([HTTPS://EN.WIKIPEDIA.ORG/WIKI/POINTWISE_MUTUAL_INFORMATION](https://en.wikipedia.org/wiki/Pointwise_mutual_information)). 11

FIGURE 3 NAMED ENTITIES (LINKED TO WIKIPEDIA) AND THE CONNECTIONS FOUND BETWEEN THEM THROUGH DOCUMENT CO-OCCURRENCE, IN THE MONITIO ENTITY NETWORK PAGE. THEY ARE ALSO SHOWN ON THE RIGHT SIDE OF THE FILTER PANE, AS DETECTED BY THE NAMED ENTITY RECOGNITION AND LINKING MODEL DEVELOPED WITHIN SELMA. 12

FIGURE 4 USER FEEDBACK COLLECTION UI FOR ENTITY LINKING 13

FIGURE 5 “STORYLINES” DASHBOARD FROM THE MONITIO PLATFORM, SHOWING THE CLUSTERS FROM THE CROSS-LINGUAL CLUSTERING MODEL DEVELOPED IN SELMA. ALTHOUGH AN ENGLISH TRANSLATION IS AVAILABLE FOR THE ARTICLES SHOWN, THEY ARE PRESENTED ABOVE IN THE ORIGINAL LANGUAGE TO EMPHASIZE THE MULTILINGUALITY OF THE PLATFORM. 14

FIGURE 6 IPTC TOPICS DETECTED ON A RUSSIAN DOCUMENT. THE TAGGING WAS DONE IN THE ORIGINAL LANGUAGE BY A MULTILINGUAL MODEL, WHEREAS TRANSLATION IS ONLY USED FOR SHOWING THE RESULT TO THE USER. 15

FIGURE 7 MULTILINGUAL MULTI DOCUMENT SUMMARY OF A CLUSTER RELATED TO THE WAR IN GAZA. AFTER BEING SELECTED, THE SUMMARY SENTENCES ARE TRANSLATED TO ENGLISH ACCORDING TO THE PREFERENCES OF THE USER (COULD BE ANOTHER DESIRED LANGUAGE). 16

FIGURE 8 SPANISH VIDEO INGESTED BY MONITIO, INCLUDING AN AUTOMATICALLY EXTRACTED TEXT TRANSCRIPT, TOPIC DETECTION AND ENTITY RECOGNITION AND LINKING. A TRANSLATION TO ENGLISH IS ALSO AVAILABLE. 17

FIGURE 9 DIVERSITY FILTER COUNTS IN THE DWNEWS SCENARIO IN MONITIO FOR A PERIOD OF 30 DAYS BETWEEN OCT-22-2022 AND NOV-21-2022. 18

FIGURE 10 PLAIN X LIBRARY SHOWING AVAILABLE VIDEOS TO WORK ON, AND PLAIN X BOARD PAGE. 22

FIGURE 11 PLAIN X TRANSCRIPTION TASK PAGE 23

FIGURE 12 PLAIN X TRANSLATION TASK PAGE, SHOWING A VIDEO TRANSCRIPT IN THE ORIGINAL AND TARGET LANGUAGE 24

FIGURE 13 PLAIN X VOICE-OVER TASK PAGE, SHOWING THE GUI COMPONENT TO HIGHLIGHT TEXT SEGMENTS AND CUSTOMIZE SYNTHETIC VOICE GENERATION PARAMETERS 25

FIGURE 14 PLAIN X VOICE-OVER VOICE INTONATION CUSTOMIZATION INTERFACE 26

FIGURE 15 PLAIN X SETTINGS PAGE WHICH ALLOWS CUSTOMIZING THE WORKSPACE IN TERMS OF PREFERRED NLP ENGINES FOR CERTAIN LANGUAGE PAIRS 27

FIGURE 16 PLAIN X SHORTCUT LIST WHICH CAN BE CONSULTED BY PRESSING “SHIFT+?” OR IN THE HELP MENU IN ANY TASK PAGE 27

FIGURE 17 PLAIN X AUDIO WAVE FORM COMPONENT. 28

FIGURE 18 PLAIN X GOAL ORIENTED WORKFLOW ADD MEDIA FORM. 29

FIGURE 19 PLAIN X ADD TASK FORM WITH “SELMA” MODEL SELECTED. 30

FIGURE 20 PLAIN X’S GUIDED TASK PROGRESS. 31

FIGURE 21 PLAIN X'S JOB DEPENDENCY GRAPH.....	31
FIGURE 22 PLAIN X ADD TASK FORM WITH "WHISPER" MODEL SELECTED.....	32
FIGURE 23 PLAIN X ADD TASK FORM WITH "LIA" TRANSCRIPTION MODEL SELECTED.....	33
FIGURE 24 PLAIN X ADD TASK FORM WITH "FHG" TRANSCRIPTION MODEL SELECTED.....	34
FIGURE 25 PLAIN X ADD TASK FORM WITH THE SELMA "LIA" SPEECH TRANSLATION MODEL SELECTED	35
FIGURE 26 PLAIN X ADD TASK FORM WITH THE SELMA "LIA" PORTUGUESE TEXT2SPEECH MODELS	36
FIGURE 27 PLAIN X AUTOMATIONS API SWAGGER DOCUMENTATION.	37
FIGURE 28 SELMA BASIC TESTING AND CONFIGURATION INTERFACE GUI (UC0).....	39
FIGURE 29 SELMA PLATFORM ARCHITECTURE.....	40
FIGURE 30 PINITREE ONTOLOGY EDITOR INCLUDES NEL STREAM LEARNING FUNCTIONALITY.....	42
FIGURE 31 LETA ONTOLOGY USED IN THE PINITREE ONTOLOGY EDITOR.....	44
FIGURE 32 GENERALIZED STRUCTURE OF A NEWS BULLETIN.....	47
FIGURE 33 THE MAIN INTERFACE OF THE PODCAST CREATOR APP.....	48
FIGURE 34 THE SETTINGS SCREEN ALLOWING TO SELECT THE SUMMARIZATION ENGINE.....	49
FIGURE 35 THE STRUCTURE EDITOR.....	50
FIGURE 36 THE MONITIO IMPORTER.....	51
FIGURE 37 INTERACTING WITH SELMA COMPONENTS.....	52
FIGURE 38 THE DW SPEAKER APP.....	53
FIGURE 39 THE DW SUMMARIZER APP.....	54
FIGURE 40 M-PHANTOM.....	55
FIGURE 41 M-PHANTOM KEYWORD MANAGER.....	56
FIGURE 42 M-PHANTOM.....	56
FIGURE 43 STORY SEGMENTATION INTERFACE SHOWCASING TOPIC CHANGE DETECTION IN TEXT STREAMS	57
FIGURE 44 STORY SEGMENTATION INTERFACE SHOWCASING TOPIC CHANGE DETECTION IN TEXT STREAMS	58
FIGURE 45 INTERACTIVE DASHBOARD OF THE SPEAKER IDENTIFICATION SERVICE.....	59
FIGURE 46 ENHANCEMENT MODULE DESIGNED FOR ADDING NEW SPEAKER PROFILES TO THE SYSTEM'S DATABASE.....	60
FIGURE 47 INTERFACE FOR SPEAKER DIARIZATION DISPLAYING THE SEGMENTATION OF A SAMPLE YOUTUBE URL.....	61

Table of Tables

TABLE 1 PRODUCTION STEPS IN PODCAST CREATION.....	46
--	----

1. Introduction

SELMA's NLP research on transfer learning, user-feedback learning and stream learning is applied to three main use cases, Multilingual Media Monitoring (UC1), Multilingual News Content Production (UC2) and SELMA NLP Service Orchestration (UC0), through the development of different software prototypes.

The UC1 requirements as described in D1.1 and updated in D1.2 and D1.3 cover broadly the requirements of a Media Monitoring platform which Priberam is taking to market, named *Monitio*. In SELMA, we leveraged the efforts from the commercial *Monitio* platform, allowing us to focus on the NLP research aspects of the Media Monitoring problem, mainly the activities related to Natural Language Processing, Transfer Learning, User-Feedback Learning and Stream Learning. Also in focus are the activities related to performance scalability for processing massive streams.

A similar approach was used on the development of the *plain X* platform for UC2, where the more commercial aspects are being financed jointly between Priberam and Deutsche Welle.

Use cases UC1 and UC2 were developed by integrating SELMA's research output into two commercial software platforms: *Monitio* (UC1) and *plain X* (UC2). Use case UC0 is a third use case meant to build and share an Open-Source platform core, incorporating SELMA research results which will allow the community to build stream processing software pipelines. This platform core is currently in use in *plain X* (UC2), as well as in *Monitio* (UC1). By applying the SELMA platform core to these two different use cases, we are also proving that it is generic enough to be useful for the Open-Source community in other NLP applications/products.

There are two additional use cases described in this document which are being pursued in SELMA which leverage the project's technologies in additional applied environments, the PiniTree use case (at the Latvian News Agency - LETA), the Podcast use case (at DW) as well as three additional standalone apps / prototypes at DW: A summarizer app and a synthetic speaker app as well as a prototype on Diversity Indication (Binary Gender).

2. Multilingual Media Monitoring (UC1)

Within Use Case 1 (UC1), we integrated results from the SELMA research tasks into the *Monitio* product, a Media Monitoring platform under development by Priberam, available at <https://app.monitio.com>. The SELMA models integrated in *Monitio* are Named Entity Recognition and Entity Linking, Entity User Correction, News Clustering, Topic Detection and Summarization (see WP2 deliverables). The SELMA maestro orchestrator has also been integrated as *Monitio*'s NLP job orchestrator, which allows *Monitio* to scale (see WP4 deliverables). In addition, open-source models for transcription (whisper) and translation (m2m100) have been integrated in *Monitio*.

In Section 7 we report the final status of the prototype's requirements last updated in D1.3.

2.1 Named Entity Recognition and Entity Linking


The Entity Linking model available in *Monitio* UC1 supports 39 languages ("pt", "es", "de", "fr", "en", "it", "ru", "hu", "ar", "zh", "lv", "fa", "pl", "nl", "uk", "id", "tr", "cs", "sv", "fi", "hi", "bn", "el", "ca", "no", "bg", "sr", "sq", "mk", "sw", "bs", "ha", "ro", "ja", "he", "am", "hr", "ps", "ur"). The limiting factor in this language list is the Wikipedia dumps we have downloaded and processed and trained the model with. We keep extending the number of supported languages by using additional Wikipedia languages to the limit of the Named Entity Recognition model, which supports ~100 languages: *af, sq, am, ar, hy, as, az, eu, be, bn, bn, bs, br, bg, my, ca, zh, zh-TW, hr, cs, da, nl, en, eo, et, tl, fi, fr, gl, ka, de, el, gu, ha, he, hi, hi, hu, is, id, ga, it, ja, jv, kn, kk, km, ko, ku, ky, lo, la, lv, lt, mk, mg, ms, ml, mr, mn, ne, no, or, om, ps, fa, pl, pt, pa, ro, ru, sa, gd, sr, sd, si, sk, sl, so, es, su, sw, sv, ta, ta, te, te, th, tr, uk, ur, ug, uz, vi, cy, fy, xh, yi*. See the WP2 deliverables for details on these models.

In the *Monitio* Platform, the results from this model are directly visible in the "Document" page, which shows the detected named entities within a specific document – see the right-side pane in Figure 1 and the highlights in the text. These entities are linked to a Wikidata code and a Wikipedia page.

Views > DW > Articles > [Веслі Кларк: Україна потребує сотні сучасних літаків](#)

Веслі Кларк: Україна потребує сотні сучасних літаків

DW Germany | web
1 hour ago



Summary

Веслі Кларк: Україна потребує сотні сучасних літаків Україна потребує більшої кількості зброї для оборони від російської агресії, адже якщо вона не втримається, війна може перекинутися далі у Європу, попереджає екскомандувач НАТО

Екскомандувач Об'єднаними збройними силами НАТО в Європі Веслі Кларк Якщо Україна не витримає оборону проти Росії, ризики війни в Європі зростуть

Article

[Веслі Кларк: Україна потребує сотні сучасних літаків](#)

Translate to English

Mark as Unread

Save Article

Topics

- politics
- diplomacy
- unrest, conflicts and war

Entities

- Wesley Clark (People)
- Ukraine (Places)
- Europe (Places)
- NATO (Organization)
- Supreme Headquarters Allied Powers Europe (Organization)
- Russia (Places)
- Deutsche Welle (Organization)

Figure 1 Named Entities (linked to Wikipedia) in the Monitio Document page, as detected by the Named Entity Recognition and Linking model developed within SELMA

The named entities are also shown in many other pages, such as Trending (see Figure 2), Entity Network and the filtering side pane (see Figure 3).



Figure 2 Named Entities (linked to Wikipedia), as detected by the Named Entity Recognition and Linking model developed within SELMA, trending in a date range of 30 days. Entities are considered to be “trending” according to a “pointwise mutual information” criterion (https://en.wikipedia.org/wiki/Pointwise_mutual_information).

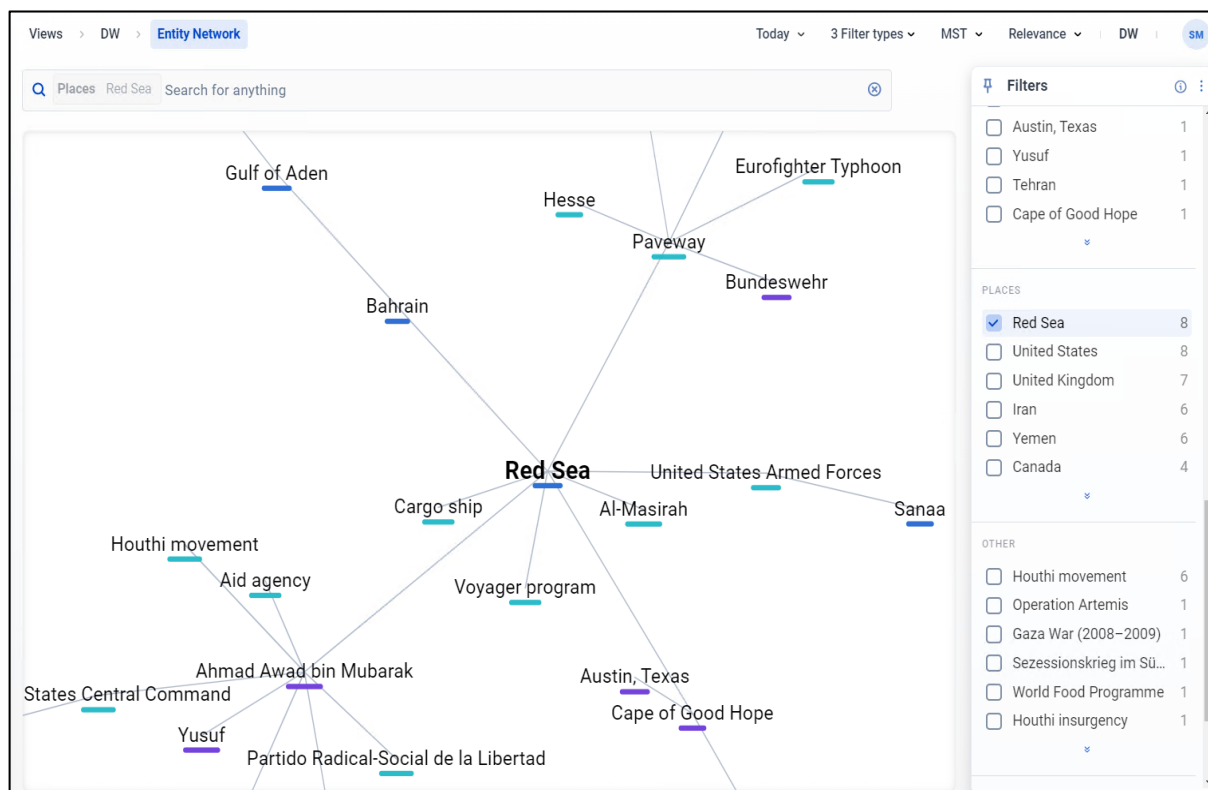


Figure 3 Named Entities (linked to Wikipedia) and the connections found between them through document co-occurrence, in the Monitio Entity Network page. They are also shown on the right side of the filter pane, as detected by the Named Entity Recognition and Linking model developed within SELMA.

2.1.1 User Entity Correction

The *Monitio* platform can receive user feedback on the results of the Entity Linking model. Users can go to the Document page in *Monitio*, select an entity and change the knowledge base link that the system has chosen (See Figure below). This user feedback is used to correct the Entity Linking decisions that the model does in future documents ingested into the platform.

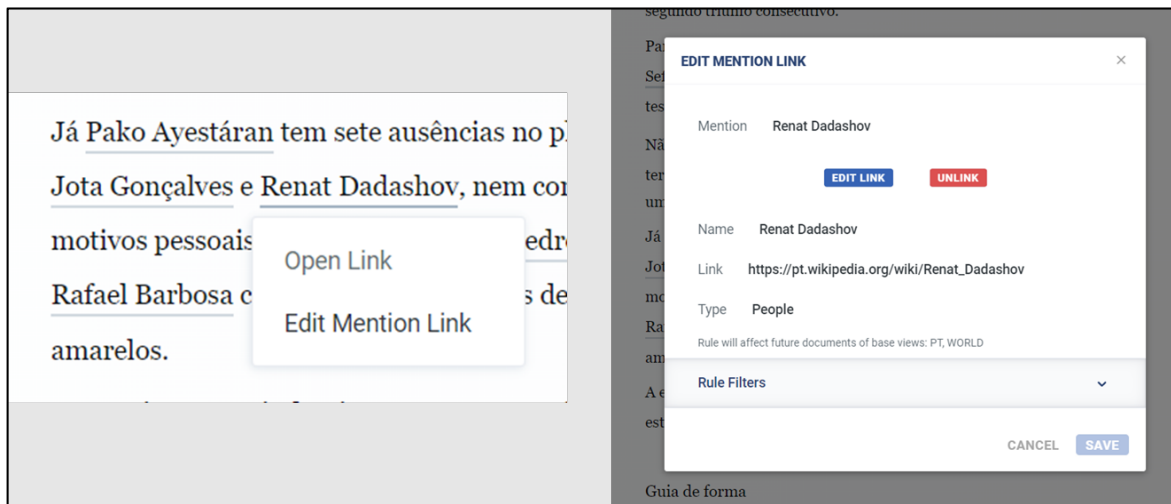


Figure 4 User feedback collection UI for entity linking

2.2 Multilingual News Clustering

The Multilingual News Clustering component leverages transfer learning by developing a new model capable of leveraging pre-trained crosslingual sentence embeddings in over 50 languages (*en, ar, bg, ca, cs, da, de, el, es, et, fa, fi, fr, fr-ca, gl, gu, he, hi, hr, hu, hy, id, it, ja, ka, ko, ku, lt, lv, mk, mn, mr, ms, my, nb, nl, pl, pt, pt-br, ro, ru, sk, sl, sq, sr, sv, th, tr, uk, ur, vi, zh-cn, zh-tw*). See WP2 deliverables for details.

In the *Monitio* Platform, the results from this model are directly visible in the “Storylines” dashboard page, which shows the most relevant aggregated news stories within a specific time range and after applied filtering. See Figure below:

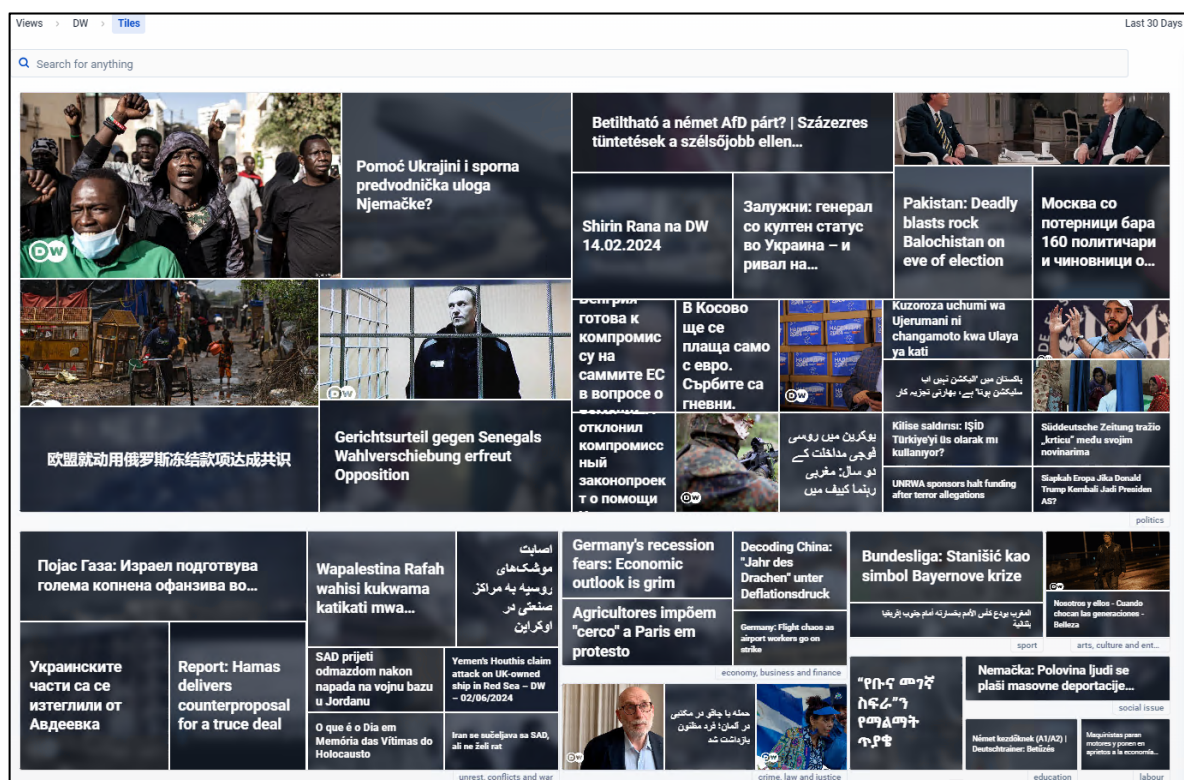


Figure 5 "Storylines" dashboard from the Monitio platform, showing the clusters from the cross-lingual clustering model developed in SELMA. Although an English translation is available for the articles shown, they are presented above in the original language to emphasize the multilinguality of the platform.


2.3 Multilingual Topic Detection

The IPTC topic detection model, described in the WP2 deliverables, is in the streaming pipeline of Monitio to associate an IPTC subject code to each incoming document (<https://www.iptc.org/standards/subject-codes/>). It supports ~100 languages: *af, sq, am, ar, hy, as, az, eu, be, bn, bs, br, bg, my, ca, zh, zh-TW, hr, cs, da, nl, en, eo, et, tl, fi, fr, gl, ka, de, el, gu, ha, he, hi, hu, is, id, ga, it, ja, jv, kn, kk, km, ko, ku, ky, lo, la, lv, lt, mk, mg, ms, ml, mr, mn, ne, no, or, om, ps, fa, pl, pt, pa, ro, ru, sa, gd, sr, sd, si, sk, sl, so, es, su, sw, sv, ta, ta, te, te, th, tr, uk, ur, ug, uz, vi, cy, fy, xh, yi*. See the WP2 deliverables for details on this model.

In the Monitio Platform, the results from this model are directly visible in the "Document" page, as seen in Figure below (IPTC Topics). The IPTC topics can also be used to filter documents in other pages, such as the "Entity Network Page" page (see the filter pane in Figure 3).

**Сколько страны НАТО тратят на оборону?
Инфографика**

DW Germany web
Today at 03:00



Summary

Сколько страны НАТО тратят на оборону? Инфографика 25

На саммите НАТО в Уэльсе в 2014 году странами-участницами был

Translate to English

Mark as Unread

Save Article

Topics

- politics
 - defence
 - military equipment
 - armed Forces
 - firearms
 - diplomacy

Entities

- NATO Organization
- Wales Places

Figure 6 IPTC Topics detected on a Russian document. The tagging was done in the original language by a multilingual model, whereas translation is only used for showing the result to the user.


2.4 Multilingual Multi Document Summarization

The Multilingual Multi Document Summarization component supports ~50 languages (*en, ar, bg, ca, cs, da, de, el, es, et, fa, fi, fr, fr-ca, gl, gu, he, hi, hr, hu, hy, id, it, ja, ka, ko, ku, lt, lv, mk, mn, mr, ms, my, nb, nl, pl, pt, pt-br, ro, ru, sk, sl, sq, sr, sv, th, tr, uk, ur, vi, zh-cn, zh-tw*). See WP2 deliverables for details. In the *Monitio* Platform, the results from this model are directly visible in the “Cluster” page, which shows the most relevant sentence highlights of a particular story cluster within a specific time range and after applied filtering. The final summary can be presented in the user’s language of preference. See Figure below.

Views > GLOBAL > Tiles > Hundreds of truckloads of aid for Gaza stuck as more lives lost to malnutrition, some aid...


Hundreds of truckloads of aid for Gaza stuck as more lives lost to malnutrition, some aid organizations say

70 articles from 43 feeds in 9 languages




Hundreds of truckloads of aid for Gaza stuck as more lives lost to malnutrition, some aid organizations say

ABC News
1 hour ago



Más camiones de ayuda humanitaria entran en Gaza desde el inicio de la guerra - Informe

The Jerusalem Post
Today at 11:24



Unterstützung vor Ort: Hungersnot in Gaza: Können Spenden aus Deutschland helfen?

STERN
Today at 10:21

Summary

- 1 "Hundreds of truckloads of aid for Gaza stuck as more lives lost to malnutrition, some aid organizations say" ABC News
- 2 "Israeli forces opened fire on people waiting for aid on Monday night in northern Gaza, eyewitnesses told CNN." Egypt Independent
- 3 "Amidst the humanitarian crisis that causes deaths and suffering in the Gaza Strip, the United Nations World Food Programme (WFP) said on Tuesday that an aid convoy was blocked by the Israeli Army in Palestinian territory and then looted by several "desperate people." Folha de S. Paulo

Figure 7 Multilingual Multi Document summary of a cluster related to the war in Gaza. After being selected, the summary sentences are translated to English according to the preferences of the user (could be another desired language).

2.5 A/V Content

We have added video ingestion and Speech2Text capabilities in *Monitio*. This allows users to search and filter A/V media items as they would any other textual media item in the *Monitio* platform. The Figure below shows an example.

Terremoto político en Ucrania por tensiones entre la cúpula militar y política

DW Germany YouTube
31/01/2024

Translate to English

Mark as Unread

Save Article

Topics

- unrest, conflicts and war
- politics
 - defence
 - armed Forces

Entities

- Volodymyr Zelenskyy (People)
- Ukraine (Places)
- Verkhovna Rada (Organization)
- The Kyiv Independent (Organization)
- Kyiv (Places)

Pulso en Zelenski y su jefe militar

Video Transcription

El presidente y su general el verano pasado, inspeccionando los sistemas antimisiles recién entregados. Una muestra pública de unidad. Pero a medida que la guerra se alargaba y una contraofensiva muy esperada contra los invasores rusos no conseguía torcer el rumbo, surgieron las tensiones. Al final de año, Zelensky dijo que el ejército le había presentado un plan, movilizar hasta 500.000 soldados más. Una medida políticamente arriesgada. Los soldados en primera línea llevan meses, algunos de ellos años, de servicios sin descanso. La opinión

Figure 8 Spanish video ingested by Monitio, including an automatically extracted text transcript, topic detection and entity recognition and linking. A translation to English is also available.

2.6 Diversity Use Case Application

One use case developed within SELMA is studying the diversity balance of People in the news. During the project this use case was analyzed due to potential ethical concerns and its final implementation was limited to an internal demonstrator with limited attributes and it was removed from the final UC1 prototype.

This use case was made possible by the Entity Linking component developed within SELMA, which links the entities to Wikipedia and thus also Wikidata (<https://wikidata.org/>). Wikidata provides many different attributes for each Entity (curated by Wikidata editors). We had selected 9 attributes to represent “diversity”: Gender, Ethnic Group, Medical Condition, Occupation, Country, Religion, Political Party, Education and Sexual Orientation. These appear in the filtering side pane of the platform, as seen in the Figure below.

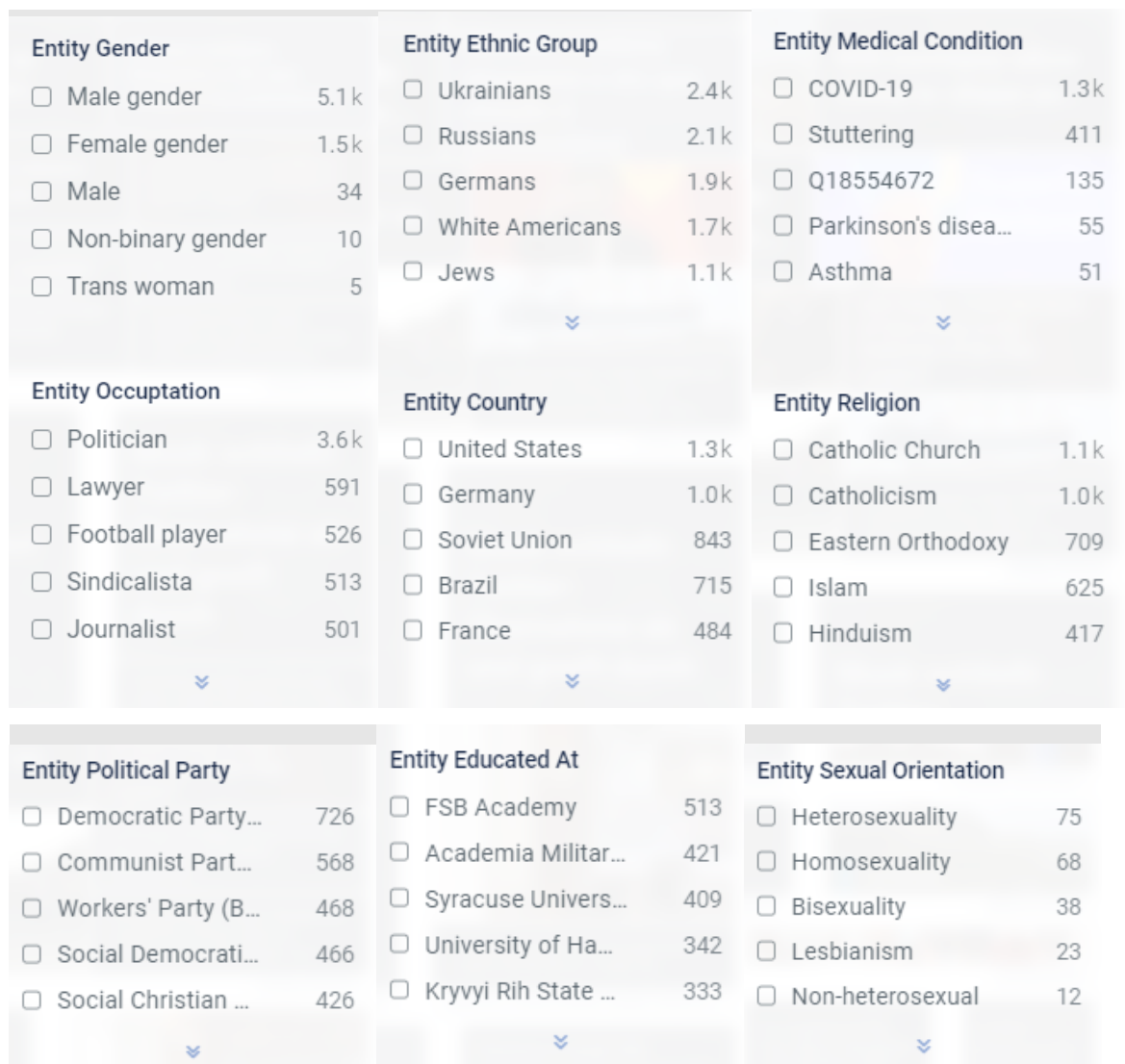


Figure 9 Diversity filter counts in the DWNEWS scenario in Monitio for a period of 30 days between Oct-22-2022 and Nov-21-2022

These filters allow users to filter news content using these attributes and to see the attributes' distribution when filtering by other fields.

2.7 Integration of Grounded Large Language Models for Explainability

The analytical results obtained in Monitio's interface for Trending Entities and Entity Network are sometimes difficult to quickly interpret without reading clicking through and reading the document which produced the trending signal, which is a slow and cumbersome process. To

address this issue, we've experimented integrating Large Language Models (LLMs) in Monitio to produce a textual explanation grounded in the documents retrieved through a RAG (Retrieval Augmented Generation) process – where a prompt enriched with the documents that produced the trending or network signal are passed to an LLM. Grounding the LLM in this way and forcing the model to cite original sources reduces the probability of information hallucination, which is a known shortcoming of LLMs. We show in the two figures below examples of this implementation. We've experimented with using both self-hosted models (e.g., LLAMA2, Mixtral) and external LLM APIs (e.g., ChatGPT).

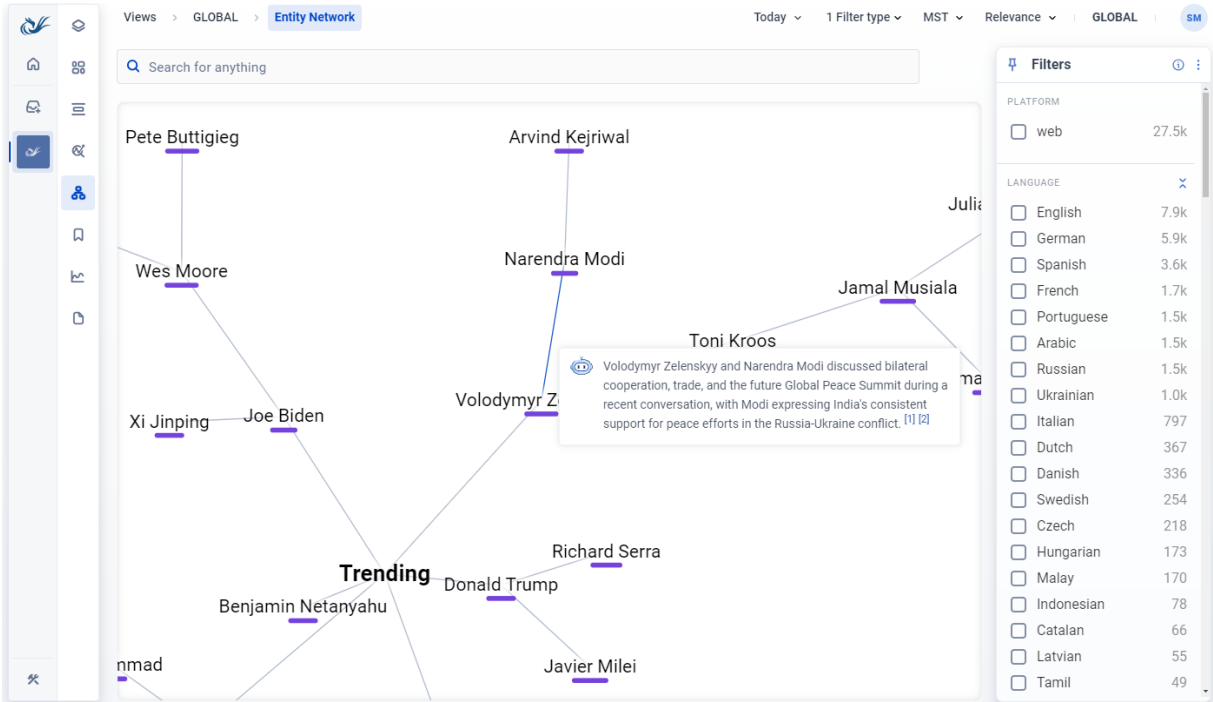


Figure 9a An LLM generated explanation of a connection between two entities ("Volodymyr Zelensky" and "Narendra Modi") in the Entity Network visualization, in the tool tip modal

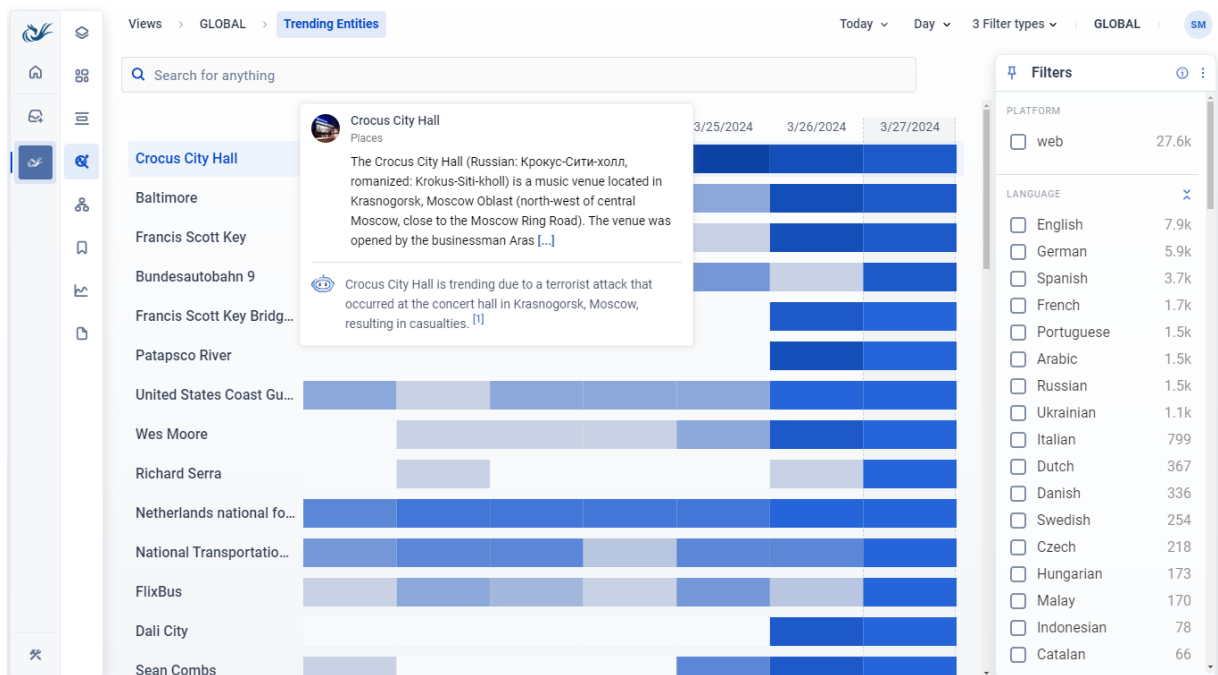


Figure 9b An LLM generated explanation of why the entity "Crocus City Hall" is trending (in the modal, below), along side it's wikipedia description (in the model, above)

3. Multilingual News Content Production (UC2)

Within Use Case 2 (UC2), we integrated results from SELMA into the *plain X* product, a Multilingual News Media Content Production platform under development by Priberam and Deutsche Welle, available at <https://app.plain-x.com>. *plain X* has been mostly developed within SELMA, started from a first prototype named “news.bridge”.

To achieve SELMA’s research goals on learning from user feedback, *plain X* stores original transcripts and user-edited versions such that this data can be used to improve models (if approved by the corresponding user).

plain X is using the SELMA orchestration to schedule and execute NLP jobs (D4.4), the ASR, Speech Translation and Text to Speech SELMA models (D3.7), and the MT model integrations (D4.4). In the case of *plain X*, the SELMA orchestration allows to execute NLP jobs not only from self-hosted APIs (e.g., the ones developed within SELMA through DockerSpaces – see D4.4) but also from many cloud providers (Azure, Google, etc).

Attached in Annex 1 there’s the *plain X* User Guide which showcases the most common functionalities of *plain X*.

Attached in Annex 2 there’s the *plain X* Automations API documentation which showcases how to integrate *plain X* in other software or workflows using the API.

3.1 *plain X* user interface

The *plain X* user interface was developed to meet SELMA’s UC2 requirements (see section 5). In this section we provide an overview of some of the most important views of the platform.

The entry point of the platform is the Library, a place where a user can find items they wish to work on (transcribe, translate, subtitle or voiceover). The Library can be filtered by repository or source language. See Figure below.

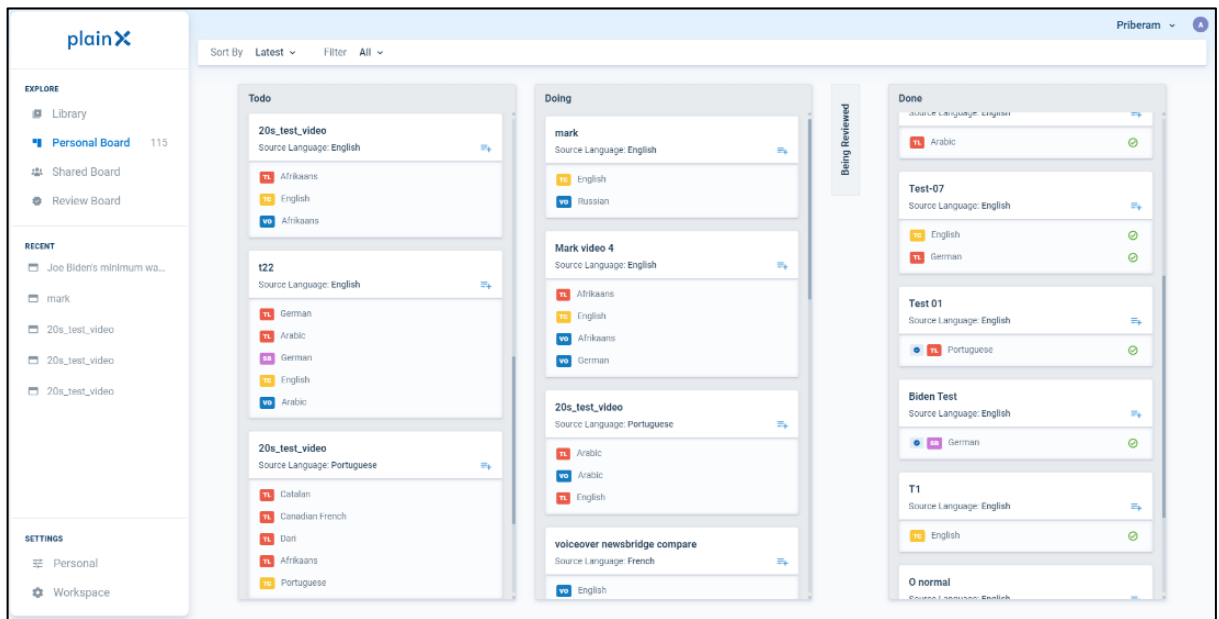
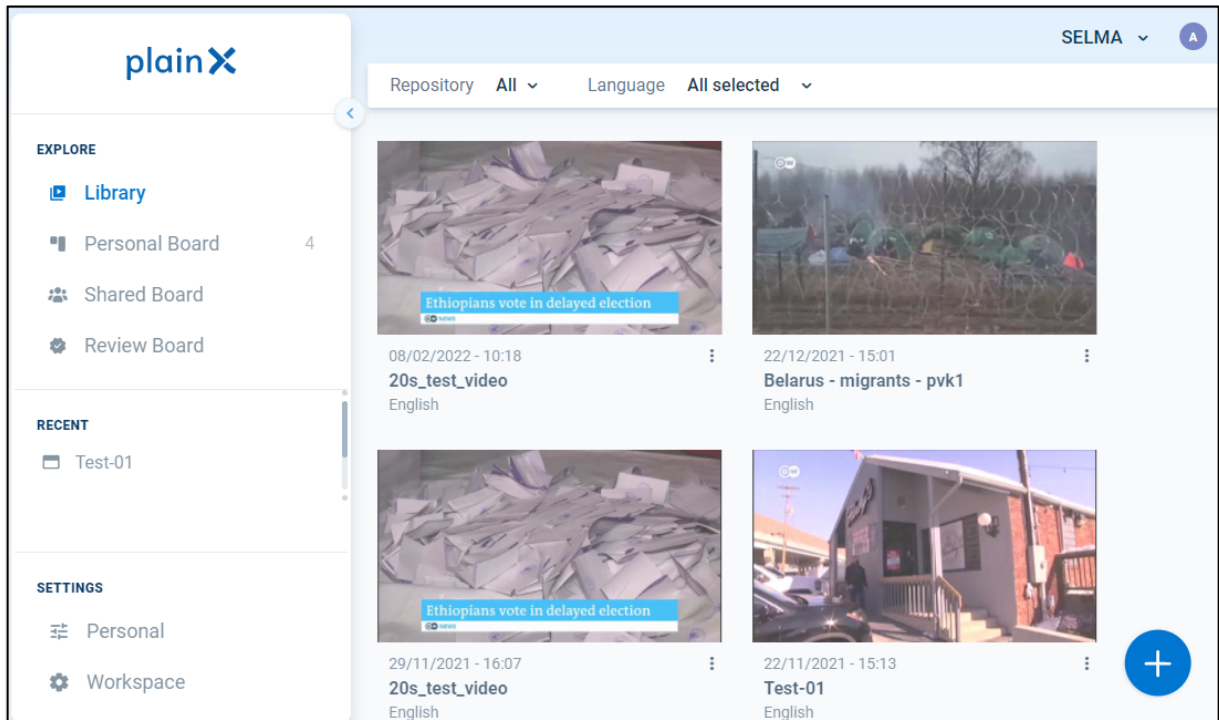


Figure 10 plain X library showing available videos to work on, and plain X board page

Besides allowing to store and find video items to work on, *plain X* also allows one person or a team of people to coordinate their tasks using a task board, as seen in Figure 10.

For each editing task, *plain X* offers a specific task page. In the Figure below, we show the Transcription task page which starts by showing to the user an automatic transcription and then allows them to edit this transcription. User edits are saved for future model improvement.

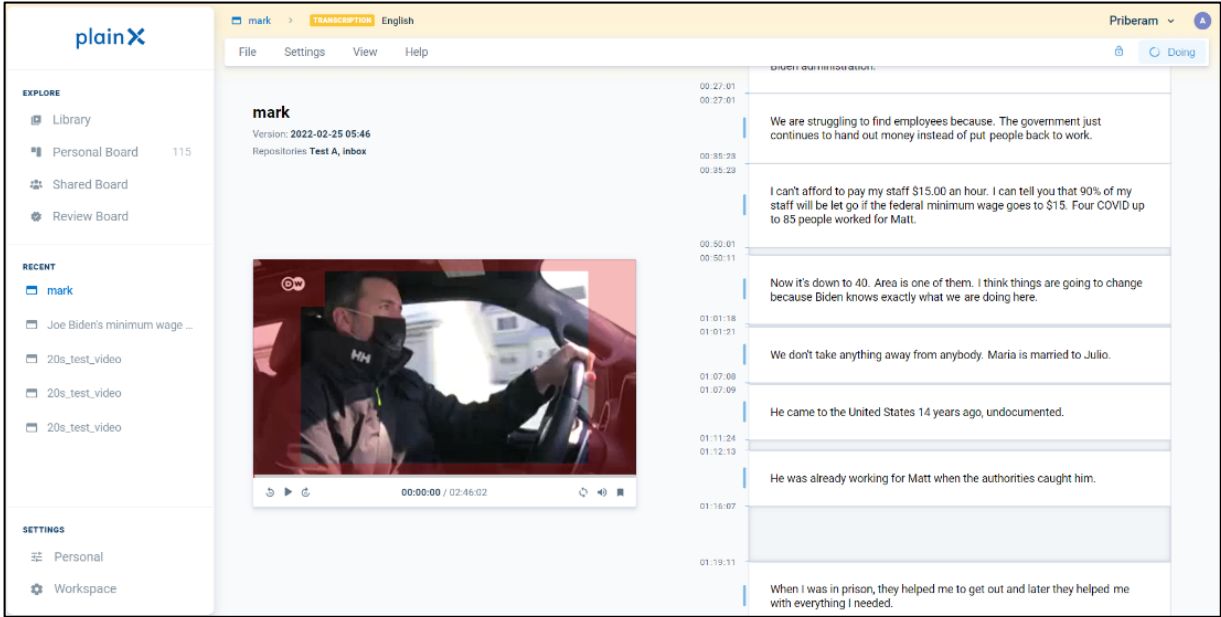


Figure 11 *plain X* Transcription task page

In the Figure below, we show the Translation task page which starts by showing to the user an automatic translation and then allows them to edit this translation. User edits are saved for future model improvement.

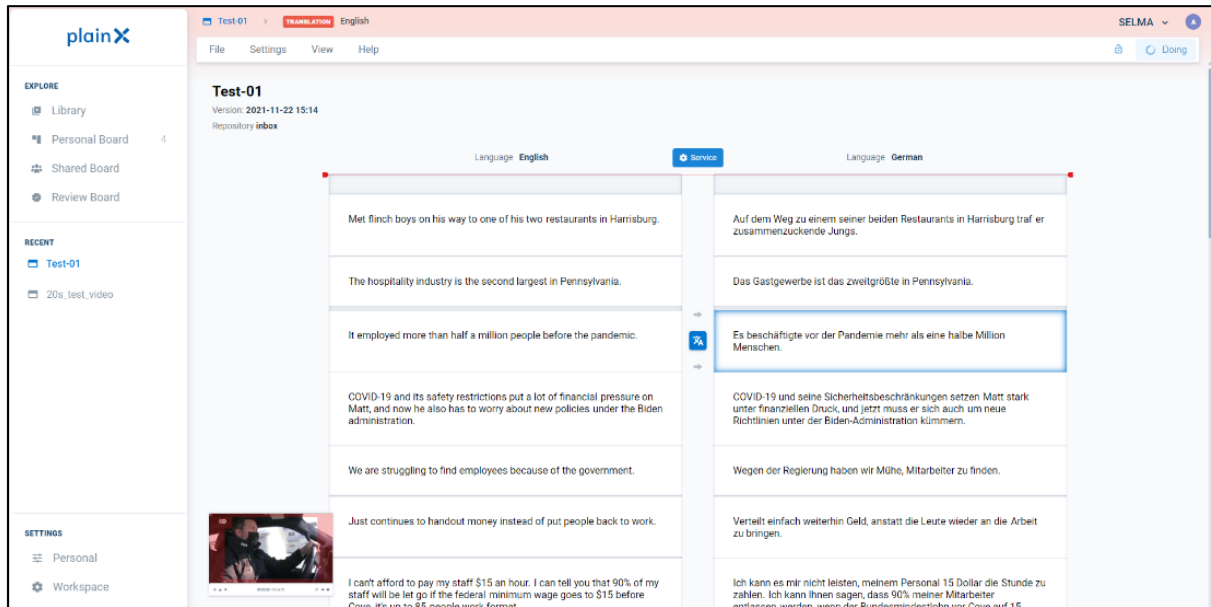


Figure 12 plain X Translation task page, showing a video transcript in the original and target language

In the Figure below, we show the Voice-over task page. A first automatic synthetic voice is generated and merged to the video's audio track. The user can customize the synthetic voice

generation by using a GUI tool to select text segments and change volume, pitch, pronunciation, etc.

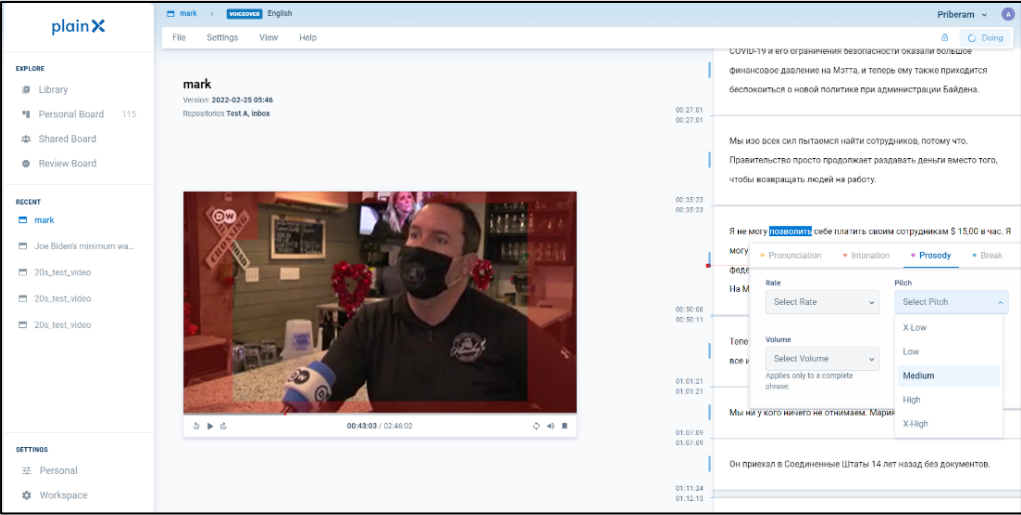


Figure 13 plain X Voice-over task page, showing the GUI component to highlight text segments and customize synthetic voice generation parameters

We have also developed a GUI to customize text intonation in a more fine-grained way, using a visual editor as shown in the Figure below.

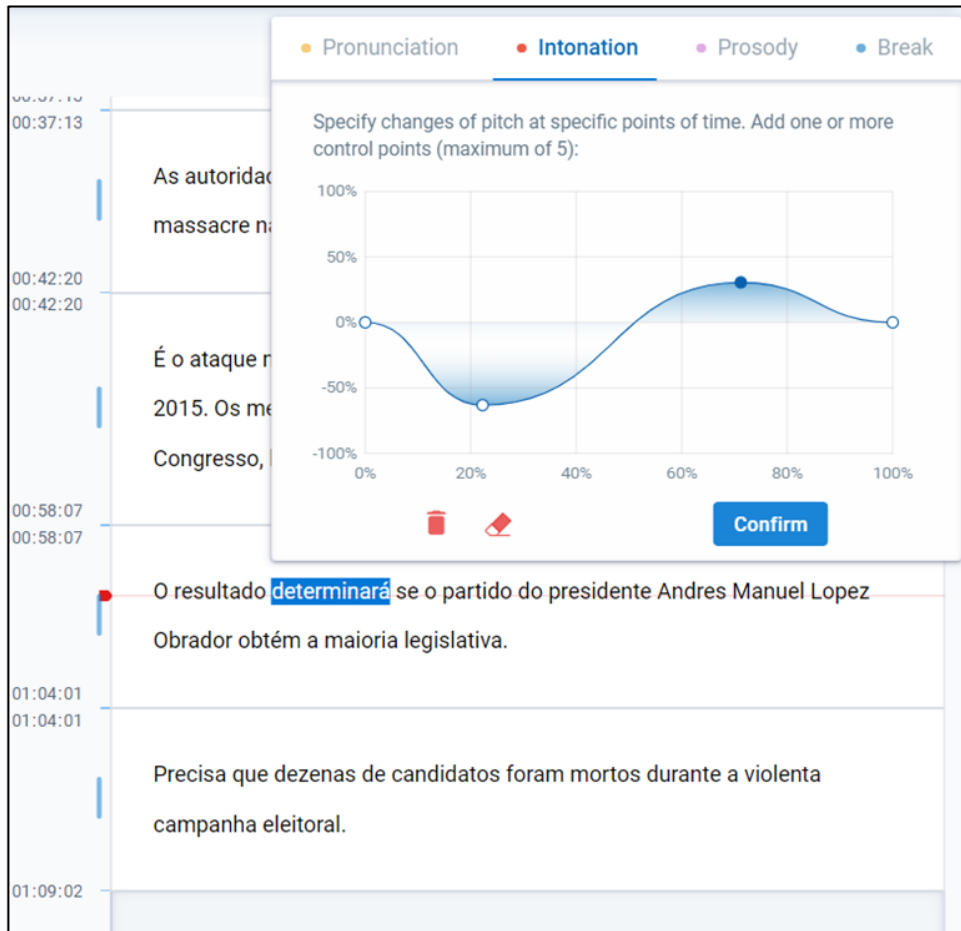


Figure 14 plain X Voice-over voice intonation customization interface

plain X also allows the creation and management of Users, Teams and Repositories (i.e. the Folders & Sources of news items). Another parameter which can be customized is which default engines should be used for each language or language pair, as shown in the Figure below.

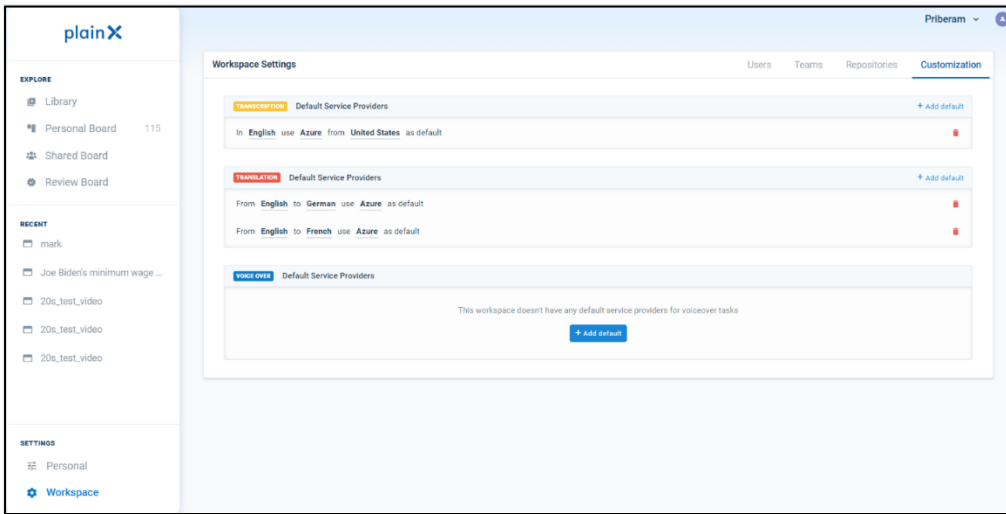


Figure 15 plain X Settings page which allows customizing the workspace in terms of preferred NLP engines for certain language pairs

In terms of accessibility, *plain X* supports a vast range of keyboard shortcuts to enhance user productivity and enable higher accessibility of the platform.



Figure 16 plain X shortcut list which can be consulted by pressing “SHIFT+?” Or in the Help menu in any Task page

3.2 plain X Audio Wave Form

To facilitate adjusting the timecodes of the transcription, plain X has an optional wave form preview component, as seen in the Figure below.

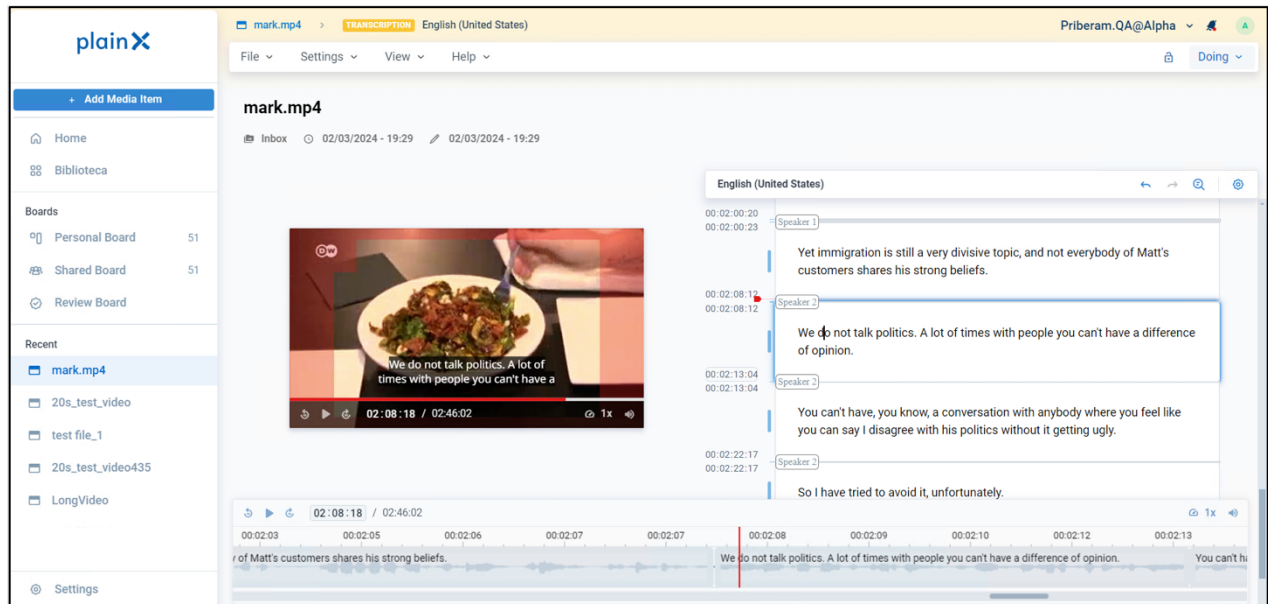


Figure 17 plain X Audio Wave form component.

3.3 plain X Goal Oriented Workflow

In *plain X* there's a streamlined task creation process named "Goal Oriented Workflow" which prompts the user for its final processing goal (Transcription, Translation, Subtitle, Voiceover) and configures the task dependencies automatically (alternatively it allows deeper customization). The Figure below shows this workflow for the case of a desired Subtitling output – in the same integrated Upload Media Form, the user only had to input the file and language and choose the desired output, in this case a subtitling in French. *plain X* will automatically configure the dependency tasks (an English transcription and an English->French translation) based on the defaults configured by the user in the platform.

Upload Media Item

Choose a media item to add and setup your tasks.


File

Title


Source Language

Add to


What task do you want to create?




TRANSCRIPTION



TRANSLATION



SUBTITLING



VOICE OVER

What is your target language?
Choose the language to which plainX should translate


ADVANCED CONFIGURATION
EDIT 

Figure 18 plain X Goal Oriented Workflow Add Media Form

Alternatively, the user can open the “Advanced Configuration” options and follow-up with the configuration of each dependency in detail, as shown in the figure below.

Setup dependencies

French
SUBTITLING

1. Transcription
2. Translation

How should plainX run this task?
This task can be automatically Marked as Done or you can mark it manually.

Automatically

▼

Upload text file
Use a text file with the transcript. It must be a text file using UTF8 encoding.

Import an existing L

↑

Select service provider
Choose which service plain X should use to run this task

SELMA

▼

Select service variant
Choose the variant for the service

Select Variant

▼

ADVANCED CONFIGURATION

EDIT ⚙

Cancel

Next

Figure 19 plain X Add Task Form with “SELMA” Model Selected

After initiating the tasks, *plain X* prompts the user with a guided progress modal dialogue to wait for the dependencies and finally open the Output task page (Figure below). This is optional, as the user can close the modal dialogue and check back on the progress later.

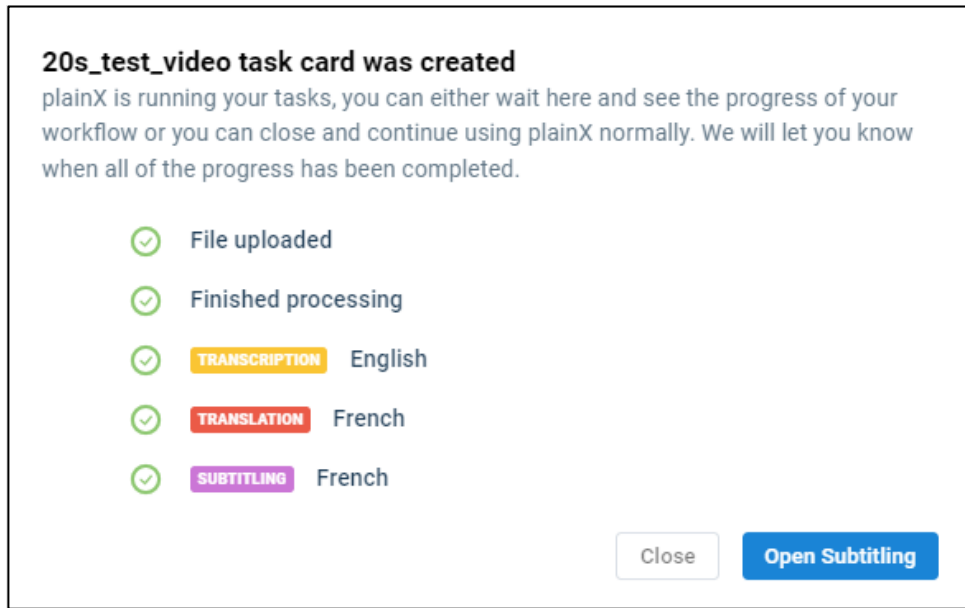


Figure 20 plain X's guided task progress

In terms of the Backend, this form produces a graph of NLP jobs that need to run on the media item. This graph is run by two SELMA components – it is orchestrated by the Maestro component and executed by a cloud provider, local infrastructure, or DockerSpaces (see D4.4 for details). The figure below shows the possible job nodes and their dependencies.

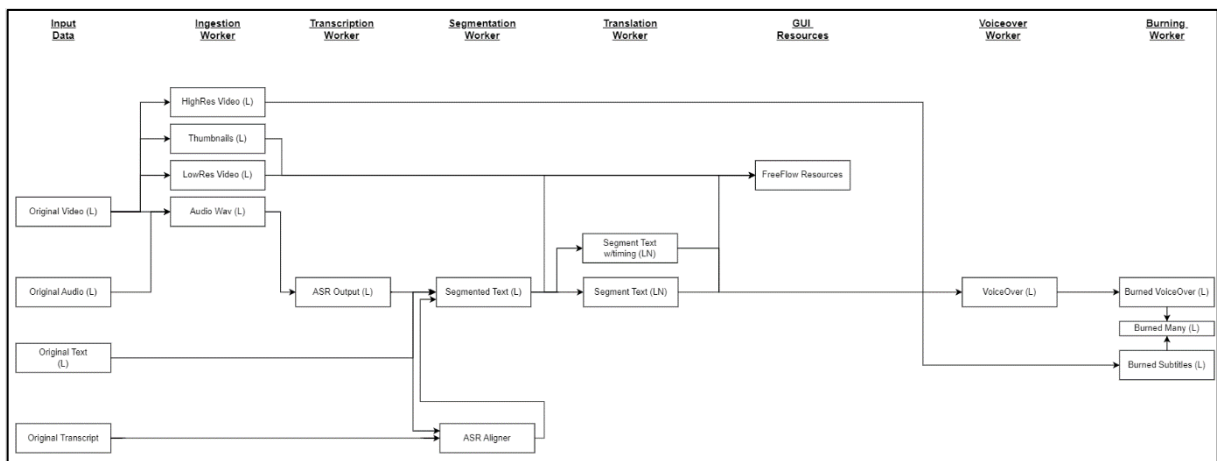


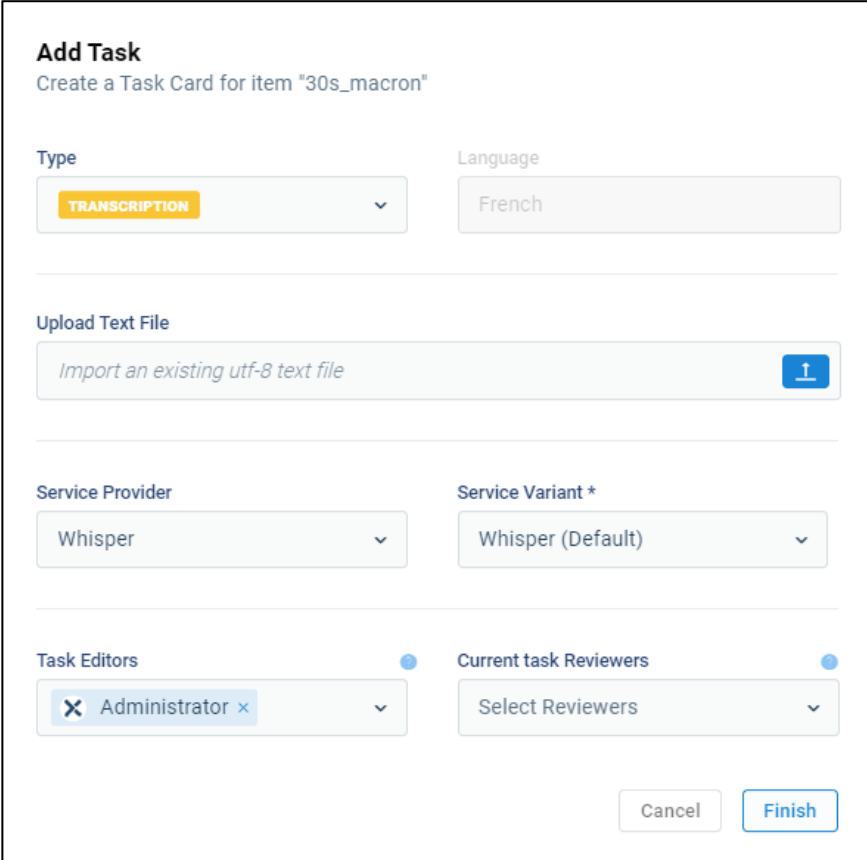
Figure 21 plain X's Job dependency graph

3.4 Models integrated in plain X

3.4.1 Speech Recognition

We have integrated in *plain X* a French transcription model developed by LIA (see WP3 deliverables), several FhG models for English, Russian, Spanish and Turkish, and the Open-Source Whisper model with additional modules (See Section 3.4.5).

As an example, you can select the Whisper and LIA models within plain X when configuring a transcription, as seen in the three figures below.



The screenshot shows the 'Add Task' form in plain X. The form is titled 'Add Task' and has a subtitle 'Create a Task Card for item "30s_macron"'. The form is divided into several sections:

- Type:** A dropdown menu with 'TRANSCRIPTION' selected.
- Language:** A text input field with 'French' entered.
- Upload Text File:** A text input field with the placeholder text 'Import an existing utf-8 text file' and a blue upload icon button.
- Service Provider:** A dropdown menu with 'Whisper' selected.
- Service Variant *:** A dropdown menu with 'Whisper (Default)' selected.
- Task Editors:** A dropdown menu with 'Administrator' selected, indicated by a blue dot.
- Current task Reviewers:** A dropdown menu with 'Select Reviewers' selected, indicated by a blue dot.

At the bottom right of the form, there are two buttons: 'Cancel' and 'Finish'.

Figure 22 *plain X Add Task Form with “Whisper” Model Selected*

Add Task
Create a Task Card for item "30s_macron"

Type: TRANSCRIPTION Language: French

Upload Text File
Import an existing utf-8 text file

Service Provider: SELMA Service Variant *: LIA (France)

Task Editors: Administrator Current task Reviewers: Select Reviewers

Cancel Finish

Figure 23 plain X Add Task Form with "LIA" Transcription model selected

Add Task

Create a Task Card for item "20s_test_video"

Type

TRANSCRIPTION
▼

Language

German

Upload Text File

Import an existing utf-8 text file

↑

Service Provider

SELMA
▼

Service Variant *

FhG
▲

FhG

Task Editors

✕ Administrator ✕
▼

Current task Reviewers

Select Reviewers
▼

Cancel
Finish

Figure 24 plain X Add Task Form with “FhG” Transcription model selected

3.4.2 Speech Translation

Another model type developed within SELMA is “end-to-end” Speech Translation, meaning that the audio in one language is converted directly to text in another language without passing through text. We have integrated in *plain X* LIA’s French to English Speech Translation and Whisper’s Speech Translation into English models. The figure below shows this option available in *plain X* for the LIA model.

Add Task

Create a Task Card for item "30s_macron"

Type

TRANSCRIPTION
▼

Language

English
▼

Upload Text File

Import an existing utf-8 text file
↑

Service Provider

SELMA
▼

Service Variant *

LIA (French->English)
▼

Task Editors

✕ Administrator ✕
▼

Current task Reviewers

Select Reviewers
▼

Cancel
Finish

Figure 25 plain X Add Task Form with the SELMA “LIA” Speech Translation model selected

3.4.3 Custom Voiceover models

In *plain X* the user can also use the custom synthetic voices developed within WP3, in particular for Brazilian Portuguese and Urdu. These are selectable in the Add Task modal, as shown in the figure below.

Add Task
Add a new task to your media item.

What task do you want to create?

TRANSLATION SUBTITLING VOICE OVER

Target language
Choose the language to which plain X should translate. Portuguese (Br... ▾

ADVANCED CONFIGURATION EDIT ⚙

How should plain X run this task?
This task can be automatically Marked as Done or you can mark it manually. Manually ▾

Service provider
Choose which service plain X should use to run this task. SELMA ▾

Service variant
Choose the variant for the service. Bruno Lupion ▾

Select who can edit this task
Are you working with anyone else on this task? Select Editors ▾

Select who will review this task.
Does this task need to be approved by someone? Select Reviewers ▾

Cancel Finish

Figure 26 plain X Add Task Form with the SELMA “LIA” Portuguese Text2Speech Models

3.4.4 plain X Automations API

The plain X automations API enables the creation of transcription, translation, subtitling and voice-over jobs and the corresponding export of the final results through REST HTTP calls. The figure below shows the deployed swagger documentation for this API. Additionally, more detailed documentation and several examples are attached in Annex 2.

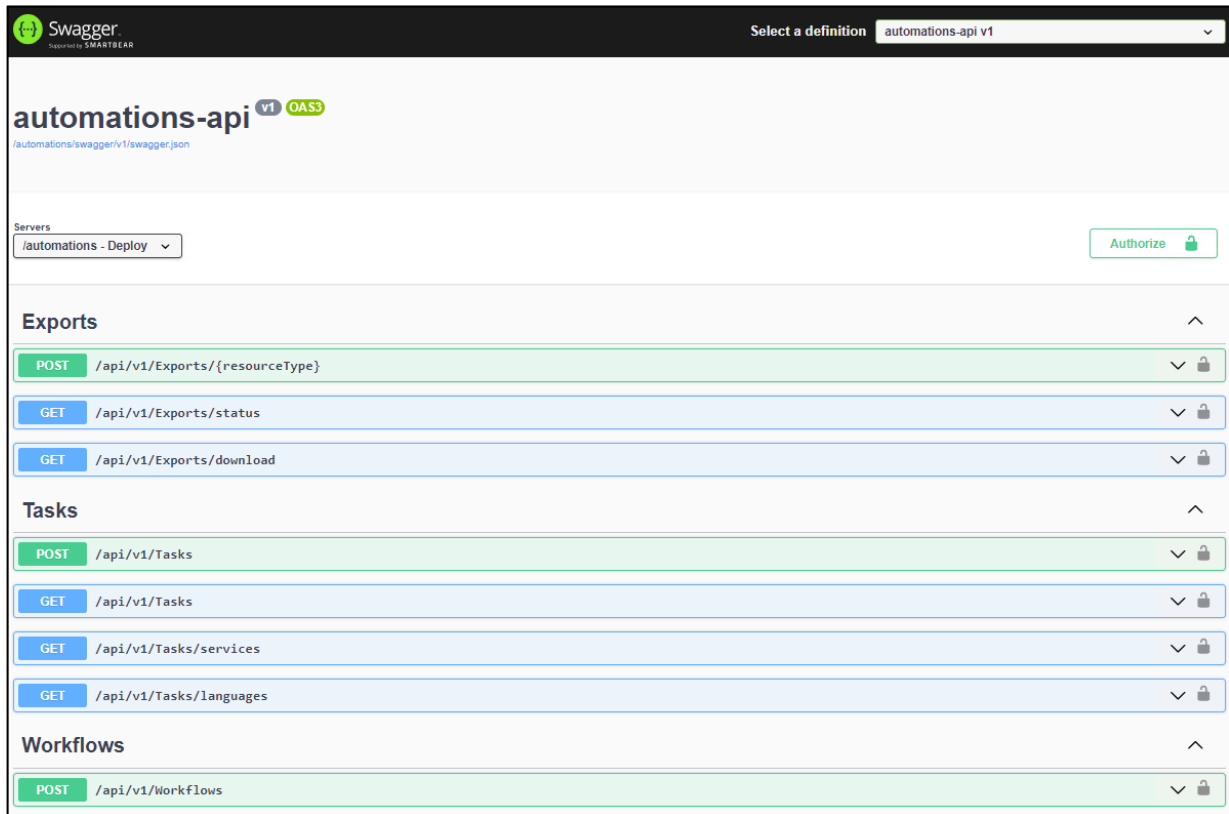


Figure 27 plain X Automations API swagger documentation.

3.4.5 Other Integrations

As described in D4.3, *plain X* can call and scale NLP services through SELMA DockerSpaces. This allows for integrating many services from different origins by just writing an API connection script. One such integration is with the output of the GoURMET European Project (<https://gourmet-project.eu/>), which provides several open-source Machine Translation Models. Another integration is with the m2m_100 translation model, which supports 100 language pairs (https://huggingface.co/docs/transformers/model_doc/m2m_100) *bs, fr, sk, pa, hi, bg, ru, mg, he, gu, hr, wo, ja, cs, lo, sd, hy, su, ff, ta, zu, ht, ms, ar, ca, th, pt, ur, hu, yo, zh, lv, ga, sr, ha, uk, yi, uz, vi, is, sv, am, ne, br, fy, bn, nl, tn, sl, no, ko, oc, ml, ns, it, ast, so, kk, gl, mk, ln, en, si, af, fa, ka, lt, az, ro, mn, gd, et, ilo, lb, km, pl, my, sq, es, cy, sw, tr, or, ba, xh, fi, mr, ig, be, kn, de, jv, id, tl, el, lg, ss, ceb, ps, da*.

The Whisper v2 and v3 models (<https://github.com/openai/whisper>) have also been integrated within a SELMA wrapper with additional modules for time code correction (vad, demucs) and

parameter tuning to reduce hallucination (logprob_threshold). It offers language support for ~100 languages: 'bs', 'fr', 'pa', 'sk', 'hi', 'bg', 'ru', 'mg', 'gu', 'hr', 'cs', 'ja', 'sd', 'lo', 'hy', 'su', 'sa', 'eu', 'ta', 'ht', 'ms', 'ar', 'jw', 'th', 'ca', 'pt', 'ur', 'hu', 'yo', 'zh', 'lv', 'sr', 'ha', 'tk', 'nn', 'uk', 'yi', 'uz', 'vi', 'bo', 'is', 'sv', 'am', 'mt', 'br', 'ne', 'bn', 'nl', 'tg', 'sl', 'no', 'ko', 'oc', 'ml', 'it', 'haw', 'so', 'kk', 'gl', 'mk', 'ln', 'en', 'si', 'af', 'fa', 'ka', 'lt', 'az', 'te', 'ro', 'mn', 'mi', 'et', 'lb', 'km', 'pl', 'my', 'sq', 'es', 'cy', 'sw', 'tr', 'ba', 'fi', 'mr', 'iw', 'be', 'la', 'kn', 'de', 'id', 'sn', 'tt', 'tl', 'el', 'bm', 'da', 'ps', 'as', 'fo'.

The m2m_100 and Whisper models are also available in UC1.

4.SELMA NLP Service Orchestration (UC0)

Use Case 0, also referred to as “SELMA NLP Service”, is a general open-source platform which serves as the core of UC1/2 and can be used by the community to implement other NLP platforms and products.

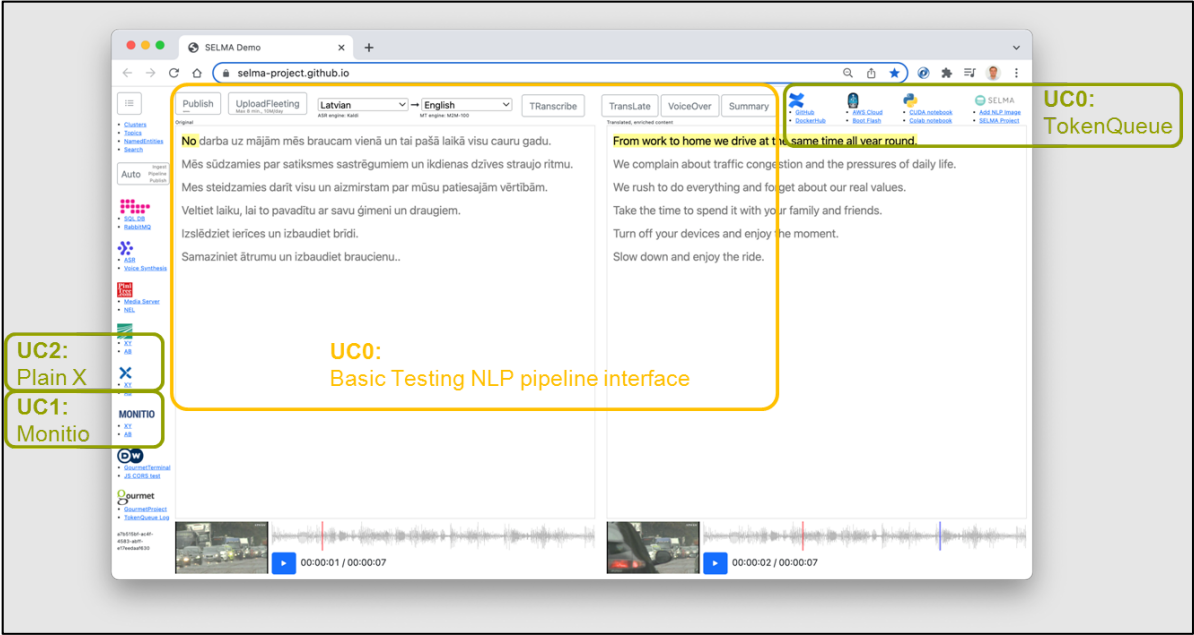


Figure 28 SELMA Basic Testing and Configuration Interface GUI (UC0)

The SELMA Basic Testing and Configuration Interface (UC0) is an open-source software <https://selma-project.github.io> (see Figure 21) for testing, deployment, scaling and monitoring of the NLP services developed within SELMA work packages WP2 and WP3. The NLP worker deployment utilizes a TokenQueue mechanism (described in D4.1) extended with the DockerSpaces technology (described in D4.3) to deliver highly scalable SELMA NLP Service Orchestration for the primary Use Cases UC1 and UC2, as described in the previous sections.

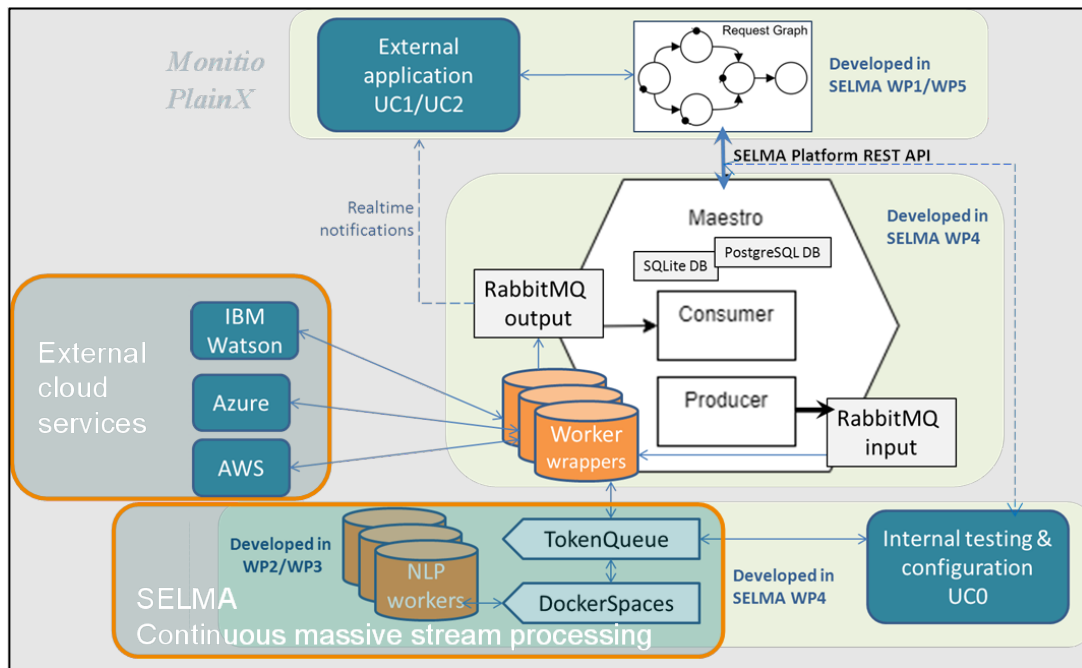


Figure 29 SELMA platform architecture

UC0 integrates with the NLP-pipeline execution orchestrator Maestro (described in D4.1) which is shared with UC1 and UC2 as shown in Figure 22. Consequently, the Maestro Orchestrator serves as a gateway between all three Use Cases UC0, UC1, UC2 allowing them to share the same scalable NLP worker pool made available via DockerSpaces continuous massive stream processing. In terms of control flow integration, Maestro Orchestrator interacts with DockerSpaces via a REST API service.

5. External Use Cases: LETA

5.1 PiniTree: Rule-based Stream Learning for NEL

SELMA partner IMCS, University of Latvia, has been involved in the Named Entity Linking (NEL) topic for several years (Barzdins, 2020; Paikens, 2016a), jointly with the Latvian national news agency LETA and the PiniTree.com startup. This has resulted in the development of the commercial PiniTree.com ontology editor. The latter integrates rule-based Stream learning of Named Entity Linking aliases as part of the entity database, against which the Named Entities are being linked. The PiniTree editor is one of the tools integrated into the SELMA Platform and besides the LETA use case, it is available for wider exploitation along with other SELMA components. PiniTree is integrated in Use Case 0 as the backend content management system accessed via the “Publish” button.

5.2 Description of the NEL Stream Learning process in the PiniTree Editor

IMCS is collaborating with the Latvian national news agency LETA which maintains a database of the nationally significant organizations and persons. The PiniTree button in LETA’s content management system allows users to link news articles to this database.

By pressing the PiniTree button, the current LETA news article opens in the PiniTree system which automatically colors the mentions of significant organizations and persons found in the LETA database (Figure 23). In most cases, the orange-colored names will be recognized correctly due to the continuous stream-learning of aliases for the entities stored in the database. However, they also need to be validated by the user so that they become valid (green-colored) mentions of an organization or person.

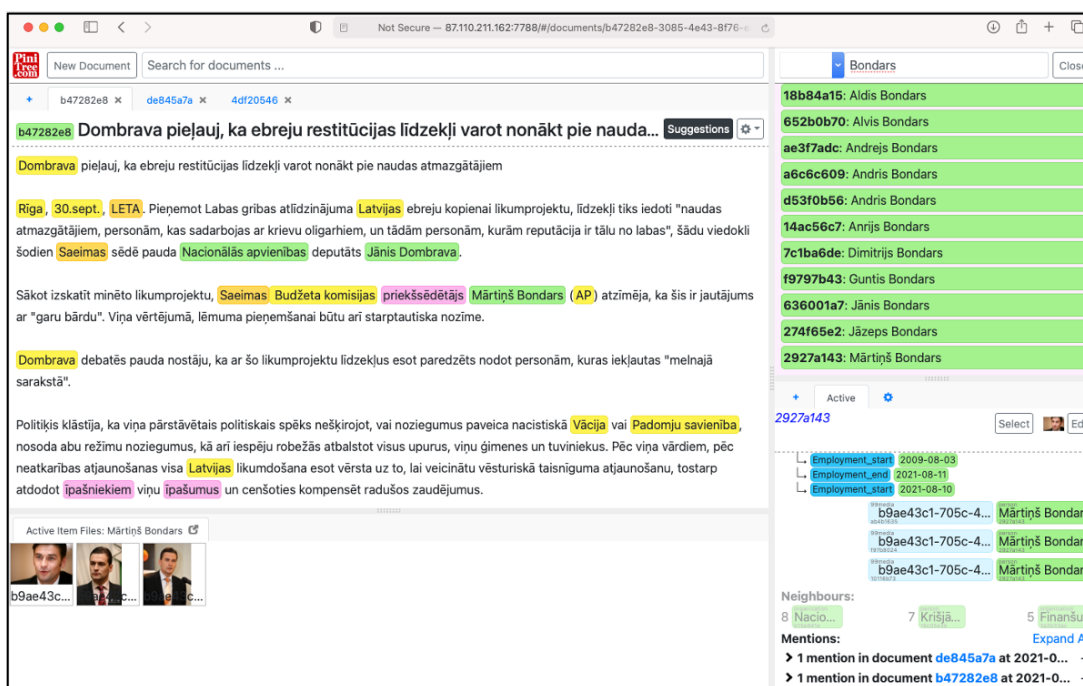


Figure 30 PiniTree ontology editor includes NEL stream learning functionality

If there are multiple persons with the same name or alias in the LETA database, they will be colored red, and the journalist must manually disambiguate the right person or organization and only then she or he will be able to validate the entity.

The article could also mention persons or organizations which are not yet included in the LETA database, they are colored in yellow by the Named Entity Recognition (NER) neural engine (Znotins, 2018) based on the massive LV-BERT large language model (Znotins, 2020). By clicking on the yellow-colored person, all the information about the person to be added is automatically filled-in inside the template on the right side of the window by the rule-based Latvian Part-Of-Speech (POS) and inflection engine (Paikens, 2016b). If everything is correct, the journalist can take action so that the new person/organization is added to the LETA database. This is how the rule-based Stream learning is implemented in the PiniTree ontology editor powering the LETA database – the newly added entity (person, organization) description includes also all possible spelling aliases for the given entity which will be automatically matched to spot that entity in all further documents automatically.

To disambiguate between people and organizations with similar names, LETA widely uses the Firmas.lv database containing facts relating persons to the organizations where they are owners or key employees. This way, the LETA database indirectly ingests up-to-date information from the Register of Enterprises of the Republic of Latvia while the PiniTree editor makes it easily accessible and allows to supplement the LETA database with new facts mentioned in the news articles.

When selecting an organization or person that is included in the Firmas.lv database in the news article, the most important facts about it are displayed automatically on the right side of the screen. By clicking on the displayed facts, one can navigate through the Firmas.lv data spider. The navigation history is displayed in the "History" field which allows the user to return easily to any of the steps visited earlier.

In the news, new facts often appear that are not part of the Firmas.lv data. PiniTree allows journalists to add such facts to the LETA database. Violet-colored phrases indicate potential new facts mentioned in the document; they are recognized in the rule-based manner using the LETA ontology (Figure 24). Clicking on a violet phrase opens a template for the new fact, where the roles mentioned in the fact must be filled in manually before adding the new fact to the LETA database by clicking the "Create" button.

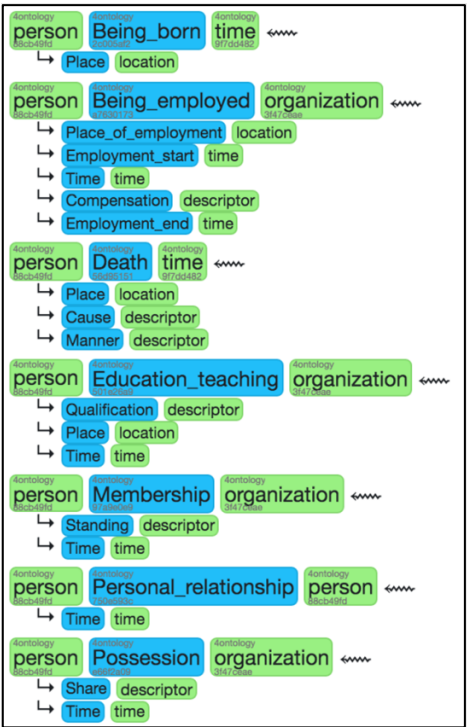


Figure 31 LETA ontology used in the PiniTree ontology editor

A new fact can also be created manually by marking the corresponding phrase in the document and pressing the "+" tab on the right side of the screen. In this case, all fields must be filled-in manually according to LETA's ontology scheme (Figure above) which focuses on the concept of a person. The ontology scheme also allows adding fine-grained sub-facts to a given fact, such as "Location", "Time" and "Qualification".

The "Edit" button can be used to edit information about organizations and persons stored in the database, such as adding a description, alternative names or a profile picture. Also, by clicking the document identification button at the upper left corner of the screen, facts or images can be added to the entire document, similarly to how they were added to a person/organization. For the newly added images to appear alongside the document, the PiniTree page must be reloaded in the web browser. It is also possible to create a visual mention for entities visible in the images by pressing the "Select" button and marking the corresponding region in the image. Regular and visual mentions appear as links under the entity at the right side of the screen.

5.3 Word Sense Disambiguation in the PiniTree Editor

Recently in collaboration with Tezaurs.lv, the Latvian online dictionary PiniTree editor has been applied to a new external Use Case – the Word Sense Disambiguation (WSD) corpus creation for the Latvian language. This new functionality is achieved by loading the entire Tezaurs.lv Latvian dictionary along with all word-inflections as aliases into the PiniTree editor – as it was described for Named Entities in Section 5.2 above. This work has just started and is showing promising results thanks to the unexpectedly high real-time performance of the PiniTree editor even with a database of full language lexicon (millions of aliases). The results of this experiment are reported in the paper (Grasmanis, et. Al, 2023) and thus provide a methodology for using the PiniTree editor for semi-automatic WSD also in the multilingual settings.

6. External Use Cases: Podcast Producer

6.1 Goal

The Podcast Creator use case is based on a workflow observed in DW's Brazilian language department. The use case's goal is to increase the workflow's efficiency by supporting the journalist in the production of daily audio news bulletins through SELMA.

6.2 Motivation

The Reuters Digital New Report¹ for 2020 highlights the rise in popularity of news podcasts over the last two years. One of the most well-known daily news podcasts is *The Daily* by the New York Times which attracts 2 million daily listeners. Based on this trend, DW's Brazilian department started its own daily news podcast in August 2020. While the monthly usage was at approximately 200.000 impressions in August 2022, it became clear how resource-intensive its production is. A full journalistic shift was required to produce the two daily 6-minute-long bulletins.

DW expects that the demand for news podcasts in other languages will rise. Consequently, it is highly desirable to be able to produce audio news content in one of DW's other 31 languages with a short ramp-up time and with minimal personnel effort.

6.3 Production Steps

The production of a single news bulletin can be subdivided into the following steps. The table shows the duration that is required for each step during the classic, manual process.

Step	What	Approx. duration
1	Research 5 stories	30 min
2	Write 5 stories	60 min

¹ <https://www.digitalnewsreport.org/>

Step	What	Approx. duration
3	Check stories by colleague	25 min
4	Recording, editing, upload into the system	70 min
5	Add metadata in CMS, create YouTube video, publish on YouTube	45 min
6	Create bit.ly links and publish on Social Media (Twitter & Facebook)	15 min
	Sum	245 min

Table 1 Production Steps in Podcast Creation

Once the Podcast creator is in use, a good benchmark will be to do these measurements again and compare them with the original production process above.

6.4 The structure of a typical news podcast

Using the example of DW Brazil's daily podcast, the structure of a typical news podcast can be generalized. The Figure below shows three sections – *Intro*, *Main Block* and *Outro*. The introduction typically contains a welcome message, followed by a selection of news headlines. The main section contains the news stories themselves, possibly separated by interstitial audio called stings (S). The outro contains a good-by message. Each section can contain music or sounds that are being played before the speech begins (M1 and M4), during speech (M2 and M5) and after the speech has ended (M3 and M6).

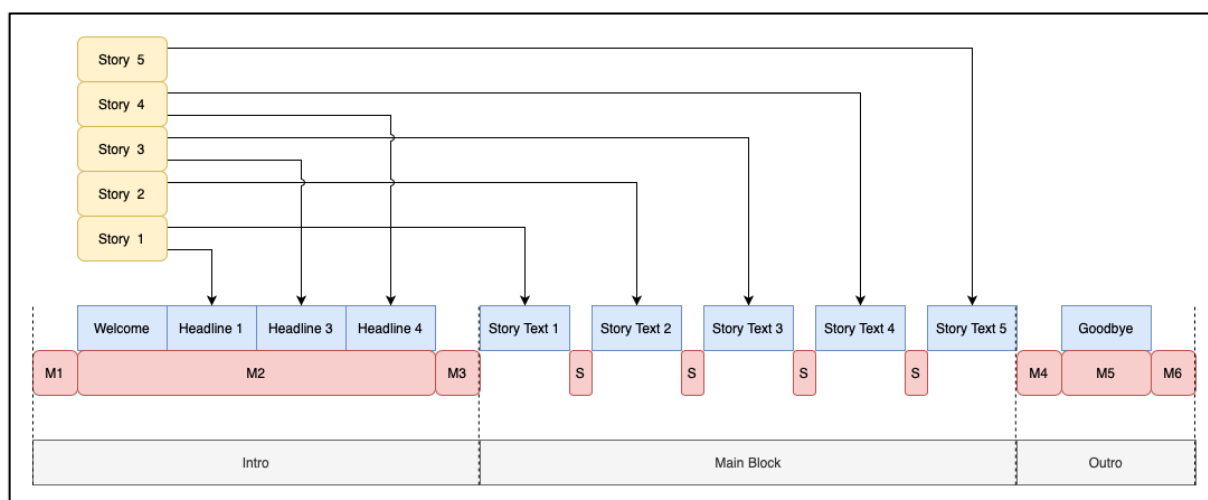


Figure 32 Generalized structure of a news bulletin

6.5 Developing a stand-alone app

The Podcast Creator use case is being explored through a stand-alone app running on macOS. The reason is twofold: first, this approach allowed us to refine the idea and the requirements while other use cases are being developed in UC0, UC1 and UC2. Second, as an external application, it allowed the consortium to design and develop APIs that expose selected functionalities of UC0, Monitio and *plain X* to external applications.

6.6 Using SELMA components to increase efficiency

We managed to use SELMA modules to support the production process in four areas:

- **Research:** cross-lingual clustering techniques as developed in SELMA and as used in MONITIO allows a journalist to identify trending stories worth investigating
- **Writing:** SELMA summarization techniques create a first draft of a story summary serving as starting point for writing the news report
- **Recording:** instead of recording speech using a microphone and recording equipment, the written text is converted into synthetic speech using a trained SELMA text-to-speech model
- **Editing:** given the rigid structure of typical audio news podcasts, the mixing of introductory music, background music and interstitial sounds can be done automatically

6.7 The Podcast Creator App in Screenshots

The Screenshot in the Figure below shows the macOS version of the Podcast Creator App developed as a use case for Deutsche Welle journalists. Two language departments within Deutsche Welle, Hindi and Urdu, are lined up to start using this application as part of a pilot project in March 2024.

The screenshot gives an impression of the app's functionality. The sidebar on the left lists the various episodes created for specific languages. The Podcast Settings (middle column) allow the user to specify the narrator's name, the engine used to create synthetic speech (Voice Provider), as well as the chosen voice. In this example, the voice 'Afsar' is selected – it was created within SELMA based on training material provided by Deutsche Welle.

Below the Settings, the Stories section lists the headlines of the stories to be included in the news podcast. The order of presentation can be changed via drag and drop. For convenience, entire DW news article can be dragged into this area, upon which headline and body text are extracted into a new story.

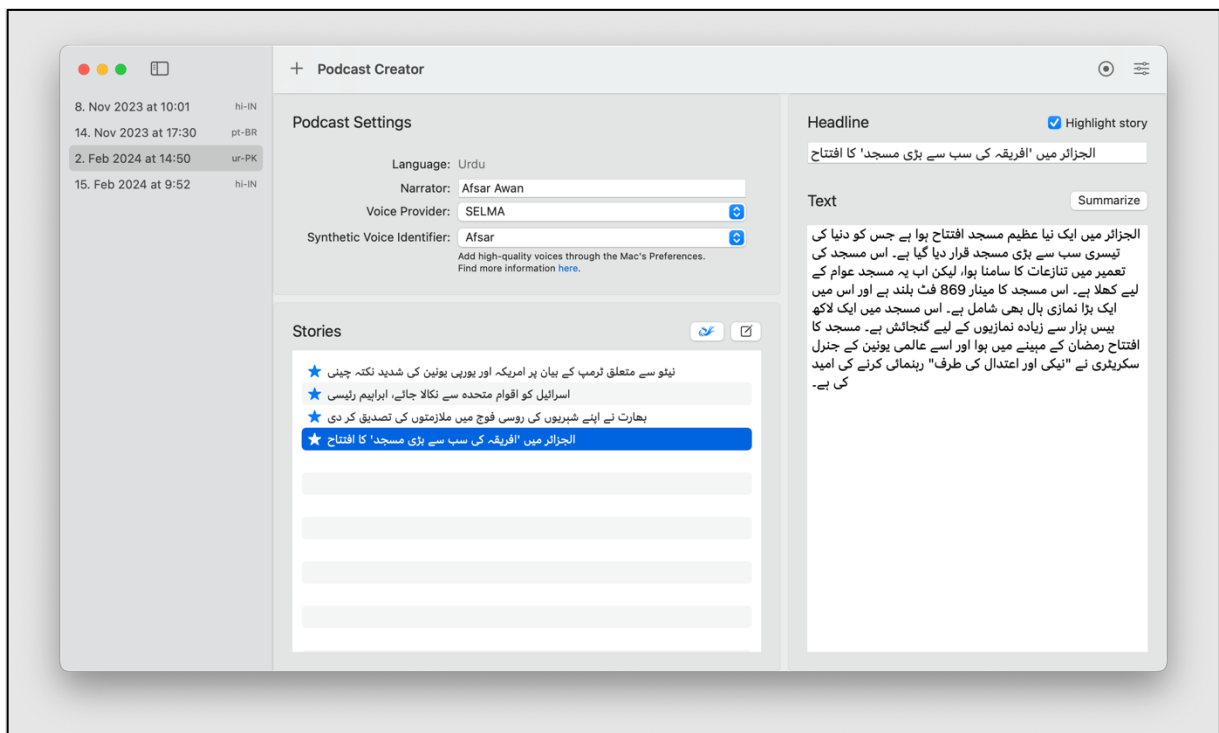


Figure 33 The main interface of the Podcast Creator app

Selecting a story headline presents the story text in the right column. Note the ‘Summarize’ button which allows the user to summarize the article text using one of several engines as shown in the screenshot in the Figure below. Amongst the selectable engines is Priberam’s summarizer developed in SELMA.

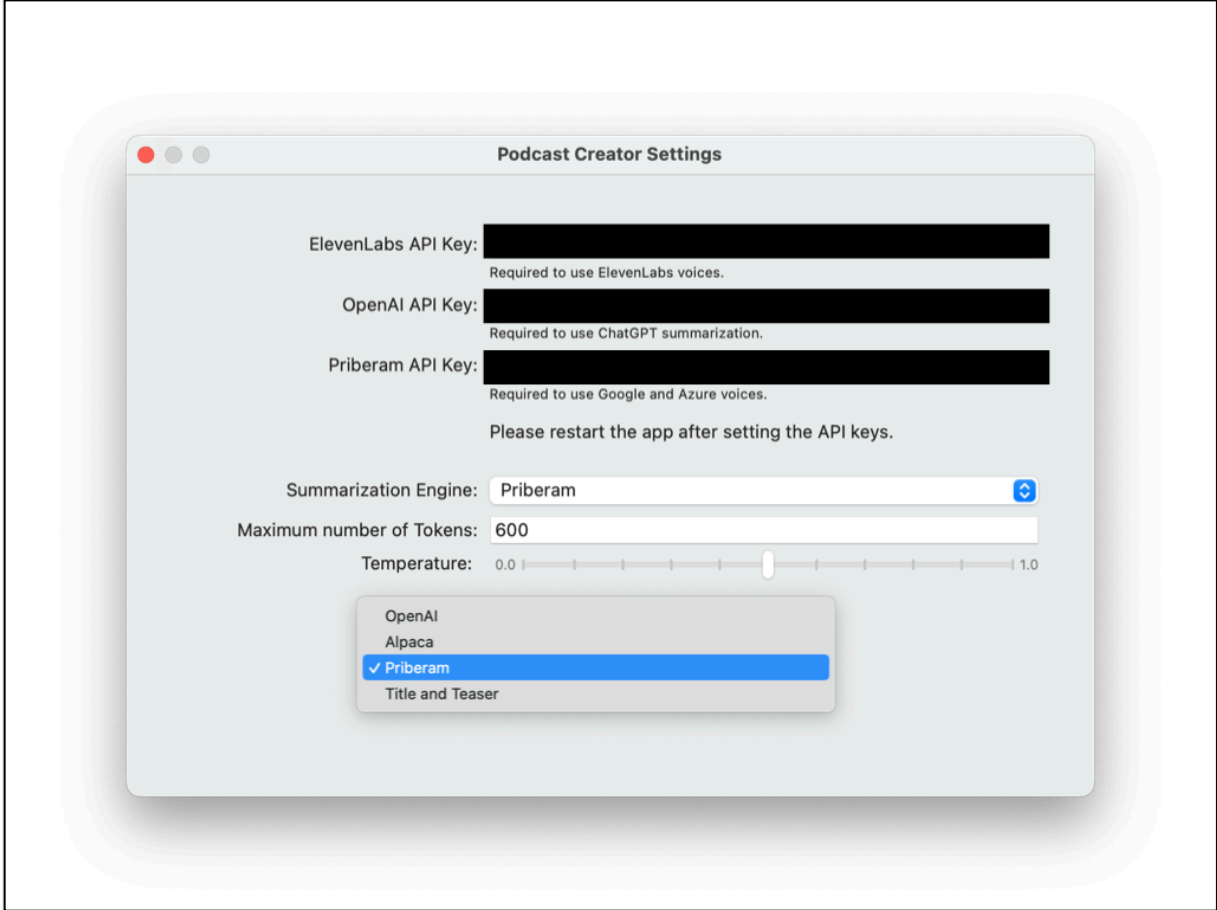


Figure 34 The settings screen allowing to select the summarization engine

The figure below shows the Structure Editor. It allows the user to specify the welcome text read out during the podcast’s introduction, with the possibility to use variables for the date and narrator. At the same time, various sounds and the background music are configurable.

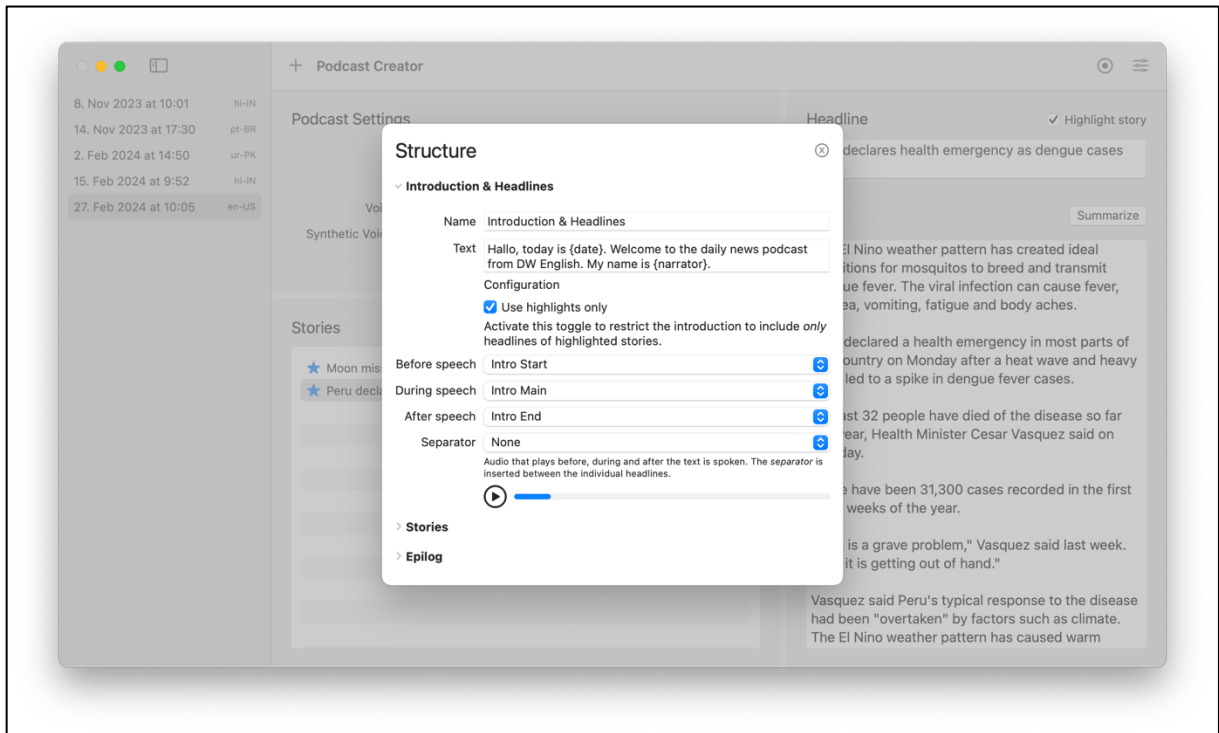


Figure 35 The Structure Editor

Finally, the figure below shows the MONITIO importer. Here, SELMA components embedded into MONITIO (UC1) propose trending stories, including their summaries, ready for import

into the application. In this example, four storylines are about to be imported from MONITIO.

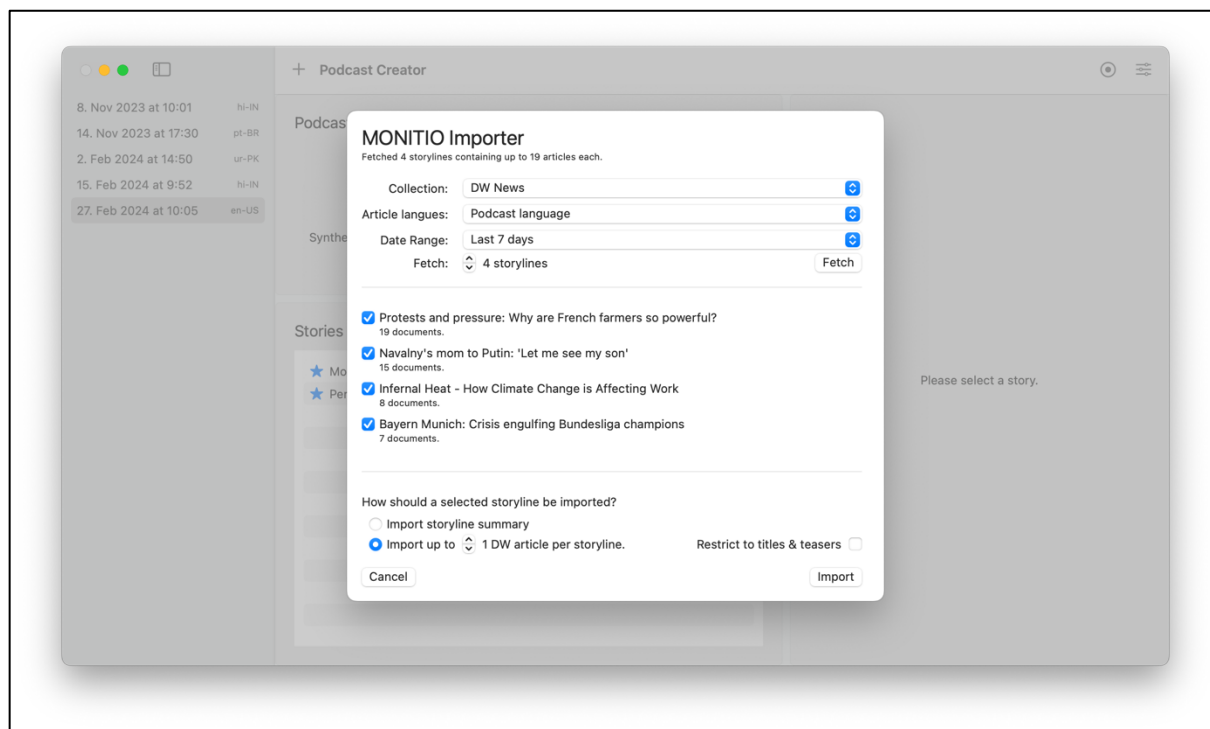


Figure 36 *The MONITIO Importer*

They are identified as trending storylines by the fact that each of them groups a large number of documents that were published over the selected period of 7 days. The bottom of the screenshot shows that, during import, each storyline cluster will be represented by a DW article contained in the given cluster.

6.8 Interaction with SELMA components

The figure below shows how the app interacts with other components developed within the context of SELMA.

The News Journalist edits her or his scripts manually within the app or drags DW articles onto the story section. An API provided through UC0 renders the text as speech which the app uses to create the final podcast.

The third branch shows the connection to MONITIO (UC1) through which trending stories and their summaries can be imported. These stories can then be used as a starting point for further

editing within the app itself before the final audio is created with the help of UC0 running on the SELMA server.

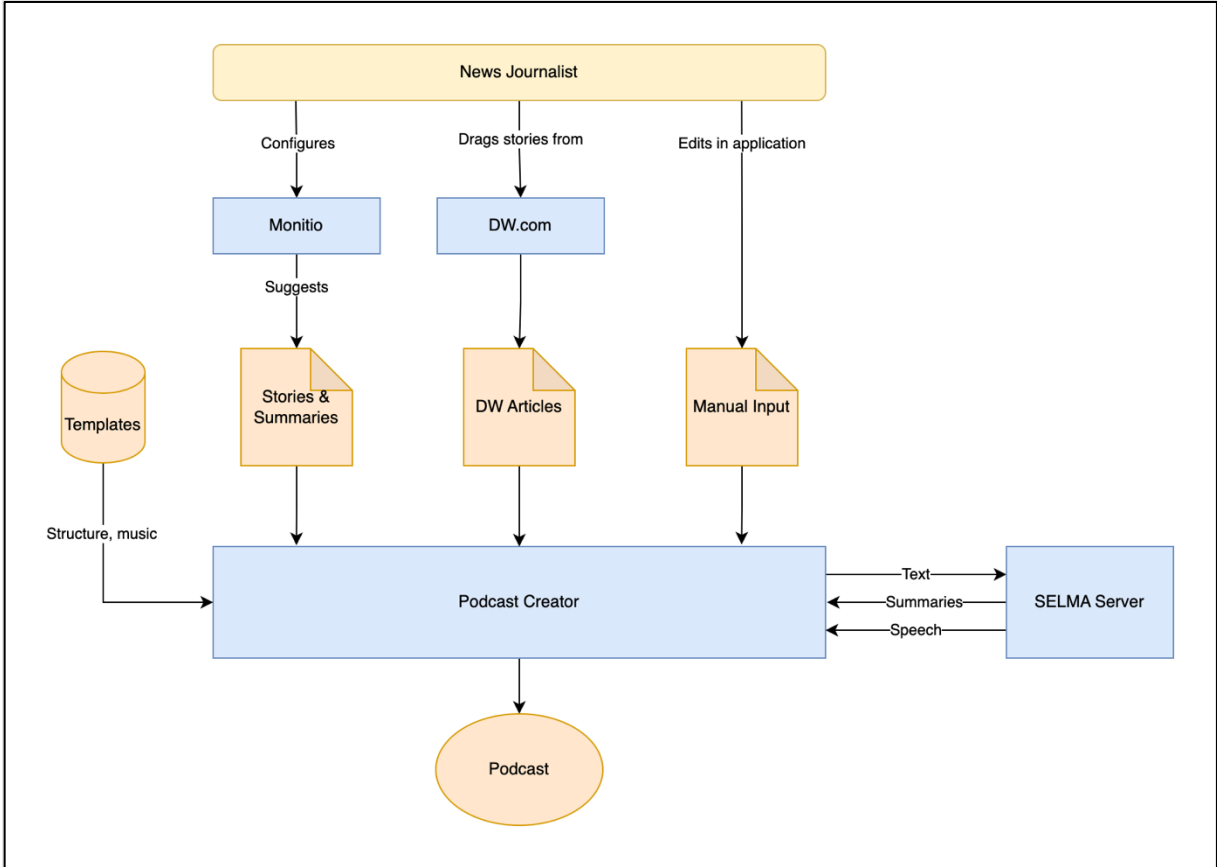


Figure 37 Interacting with SELMA components

6.9 Breaking out sub-functionalities into separate apps

During the development process of the Podcast Creator app, we developed two additional apps allowing us to test the main app’s functionality in isolation: the **DW Speaker** and the **DW Summarizer**.

Figure 31 shows a screenshot of the DW Speaker. The user can use the same Speech engines as available through the Podcast Creator app to convert text to speech. In this example, we show the SELMA voice ‘Leila’, one of the 9 Brazilian voices created in SELMA.

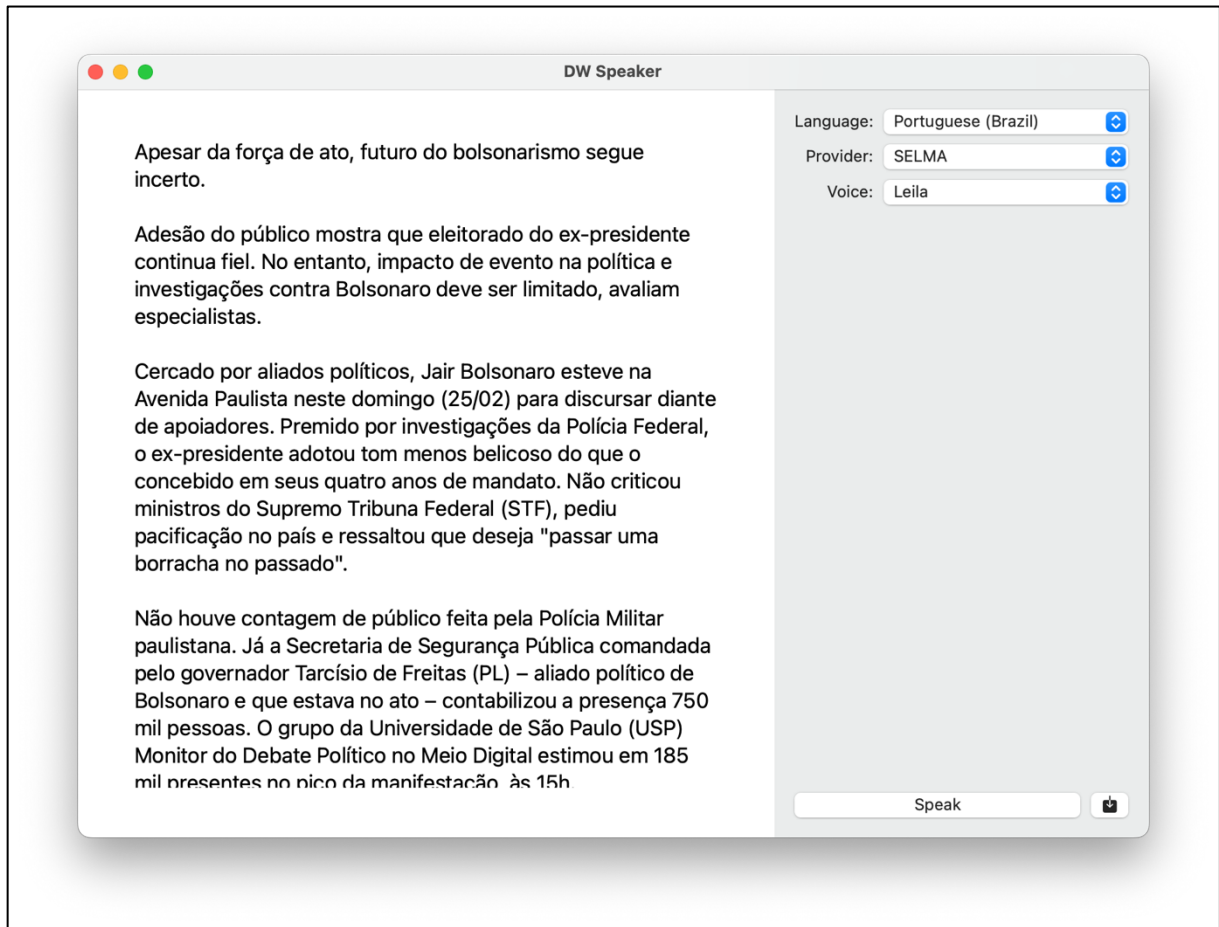


Figure 38 *The DW Speaker App*

Figure 32 shows a screenshot of the DW Summarizer which allowed us to test and compare various summarization engines, amongst them the abstractive summarisation engine developed by Priberam². In this example, a DW article written in Urdu is summarized in the style of a DW radio report using the OpenAI API.

² Diogo Pernes, Afonso Mendes, and André F. T. Martins. 2022. Improving abstractive summarization with energy-based re-ranking. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 1–17, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

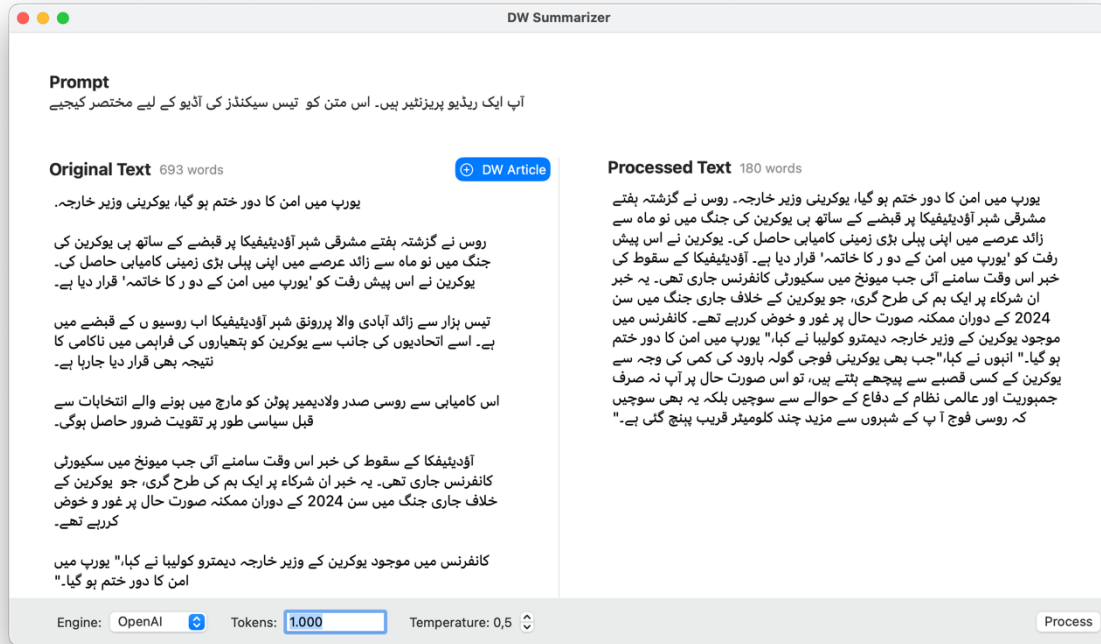


Figure 39 The DW Summarizer App

7. Research Prototypes

7.1 M-PHANTOM Transcription Prototype

As detailed in D3.7, M-PHANTOM is a new model researched in SELMA which operates together with the Whisper transcription model. It leverages the feedback collected by a user correcting an automatic transcription (text edits + the corresponding source audio segment) to improve future transcriptions. We have built an UI prototype which allows running transcriptions, collecting user feedback and curating models improved from user feedback which are compatible with *plain X*. M-PHANTOM offers language support for ~100 languages: 'bs', 'fr', 'pa', 'sk', 'hi', 'bg', 'ru', 'mg', 'gu', 'hr', 'cs', 'ja', 'sd', 'lo', 'hy', 'su', 'sa', 'eu', 'ta', 'ht', 'ms', 'ar', 'jw', 'th', 'ca', 'pt', 'ur', 'hu', 'yo', 'zh', 'lv', 'sr', 'ha', 'tk', 'nn', 'uk', 'yi', 'uz', 'vi', 'bo', 'is', 'sv', 'am', 'mt', 'br', 'ne', 'bn', 'nl', 'tg', 'sl', 'no', 'ko', 'oc', 'ml', 'it', 'haw', 'so', 'kk', 'gl', 'mk', 'ln', 'en', 'si', 'af', 'fa', 'ka', 'lt', 'az', 'te', 'ro', 'mn', 'mi', 'et', 'lb', 'km', 'pl', 'my', 'sq', 'es', 'cy', 'sw', 'tr', 'ba', 'fi', 'mr', 'iw', 'be', 'la', 'kn', 'de', 'id', 'sn', 'tt', 'tl', 'el', 'bm', 'da', 'ps', 'as', 'fo'.

Below follow screenshots of the M-PHANTOM UI in three different phases of use. First, the initial audio uploaded by the user and the corresponding automatic transcript – this is a medical transcript with several transcription errors. In this picture we highlighted the wrongly transcribed words with a red dash.

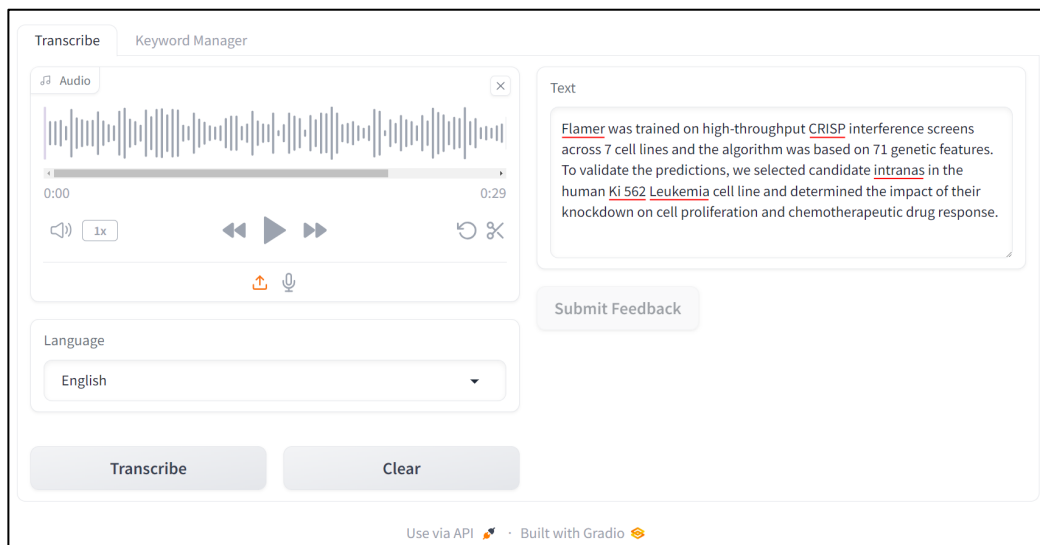


Figure 40 M-PHANTOM

Then, the user did textual corrections, and the system extracted the five keywords listed below, associated with the corresponding audio segment to inform the model.

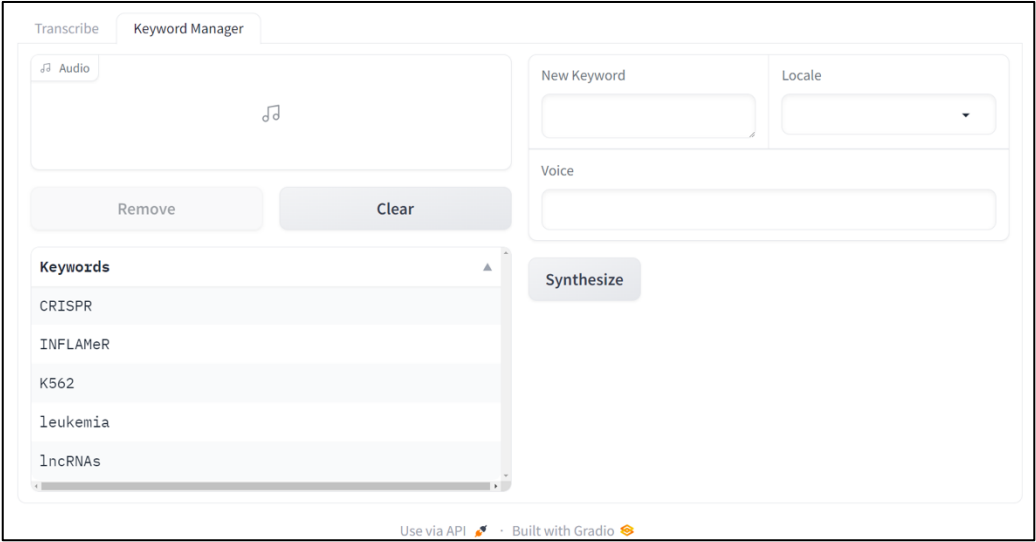


Figure 41 M-PHANTOM Keyword Manager

Finally, the user re-run the transcription + M-PHANTOM model and obtained a transcript with the errors corrected. Note that this now can be applied to any new audio. The figure shows with a green underline the corrected words.

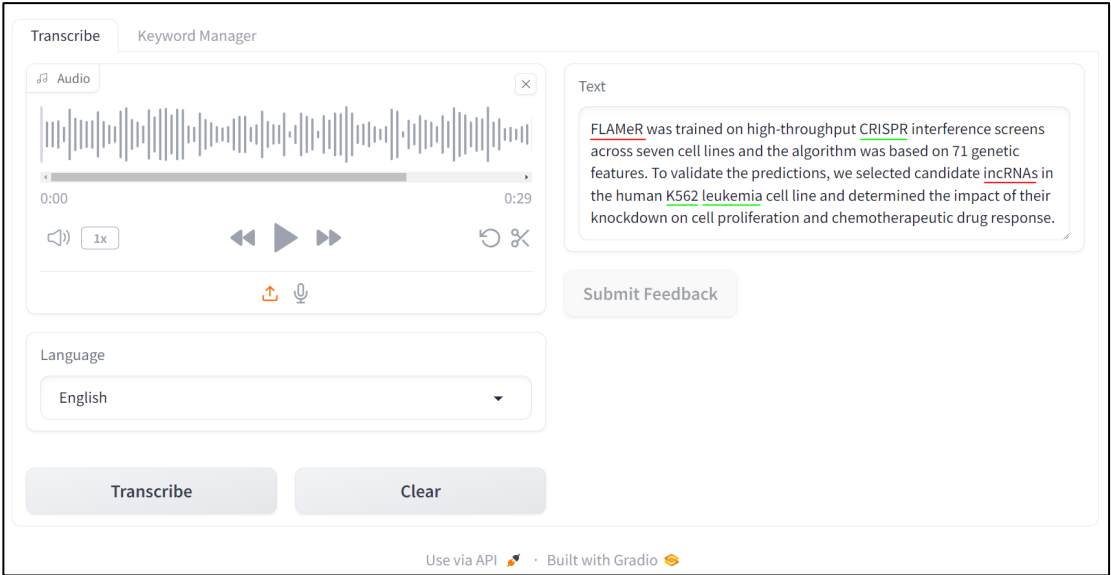
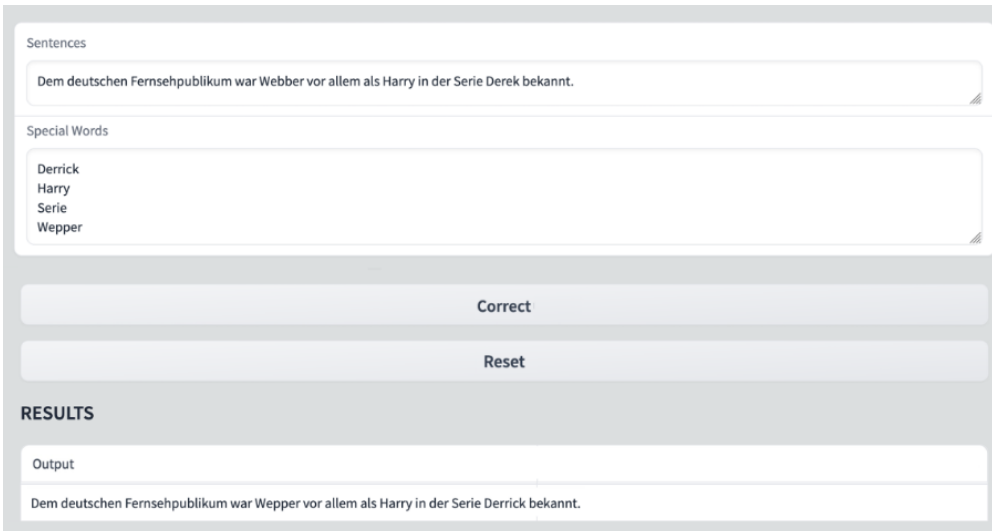


Figure 42 M-PHANTOM

7.2 Post Editing Prototype

In the domain of speech recognition, the accurate transcription of domain-specific terminology and out-of-vocabulary (OOV) words remains a considerable challenge. Contextual correction models are an alternative to shallow fusion to improve speech-to-text quality given user vocabulary. Traditional text-based mechanisms are often insufficient, as they rely on broad language models to correct common words but falter with specialized or uncommon entities. However, our prototype tool is a non-autoregressive model specifically designed to address the nuances of post-processing. This tool focuses on the precision editing of speech-to-text output to include highly specific user entities, OOV words, or unique spelling variations that conventional speech models overlook.

To refine transcription outputs, our Gradio-based application offers an intuitive interface for post-editing, integrating seamlessly with custom vocabularies. This app enables users to automatically post-edit the transcriptions without the need for retraining or complex language model integrations. The application below features a dual-text box layout, where users can input the speech-to-text generated sentences and a list of specialized entities that the model should recognize as significant for correction.



The image shows a web-based interface for text correction. It features a 'Sentences' input field containing the text 'Dem deutschen Fernsehpublikum war Webber vor allem als Harry in der Serie Derek bekannt.'. Below this is a 'Special Words' list containing 'Derrick', 'Harry', 'Serie', and 'Wepper'. There are two buttons: 'Correct' and 'Reset'. A 'RESULTS' section shows the 'Output' as 'Dem deutschen Fernsehpublikum war Wepper vor allem als Harry in der Serie Derrick bekannt.', where 'Webber' has been replaced by 'Wepper' and 'Derek' by 'Derrick'.

Figure 43 Story Segmentation Interface Showcasing Topic Change Detection in Text Streams

7.3 Story Segmentation Prototype

Dynamic text documents such as news articles, user reviews, and multi-speaker dialogues evolve over time, which are reflecting changes in topics, sentiments, and public discourse. Understanding these shifts is crucial for media analysis. However, the inherent complexity of tracking thematic changes, especially in a multi-lingual context, presents a significant challenge. SELMA's Gradio-based UI presented below is designed to meet this need by detecting topic change points in text streams. This tool is particularly adept at analyzing the output of speech-to-text transcriptions, allowing users to pinpoint the precise moments when a discussion or story shifts. By facilitating a clearer understanding of how topics evolve over time, the tool enables users to quickly adapt to new information and gain a deeper understanding of the textual landscape in multiple languages, including Spanish, English, French, Portuguese, and German.

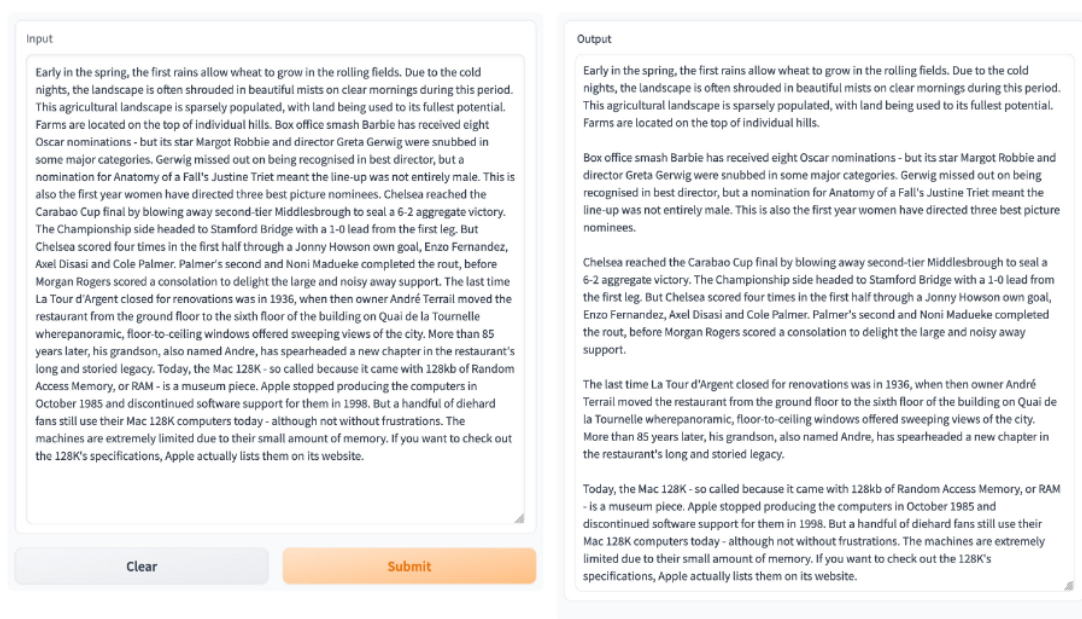


Figure 44 Story Segmentation Interface Showcasing Topic Change Detection in Text Streams

Our interface employs machine learning to process text streams and identify significant shifts in content, enabling the detection of topic change points. To accommodate multi-lingual support, we use text embeddings trained in language-agnostic representation of textual data. By transforming text into these embeddings, our tool can capture semantic nuances across different languages.

7.4 Speaker Identification Prototype

In SELMA, we focus on text-independent speaker recognition when the speaker's identity is based on how the speech is spoken, not necessarily on what is being said. Typically, such systems operate on unconstrained speech utterances, which are converted into vectors of fixed length, called speaker embeddings. Our Gradio-based prototype speaker identification service is designed specifically for recognizing speakers from the German parliament members. After uploading a sample speech from a parliament session, the pre-trained TitaNet-based embedding model, a robust neural network architecture designed for speaker identification, processes the audio to infer the speaker.

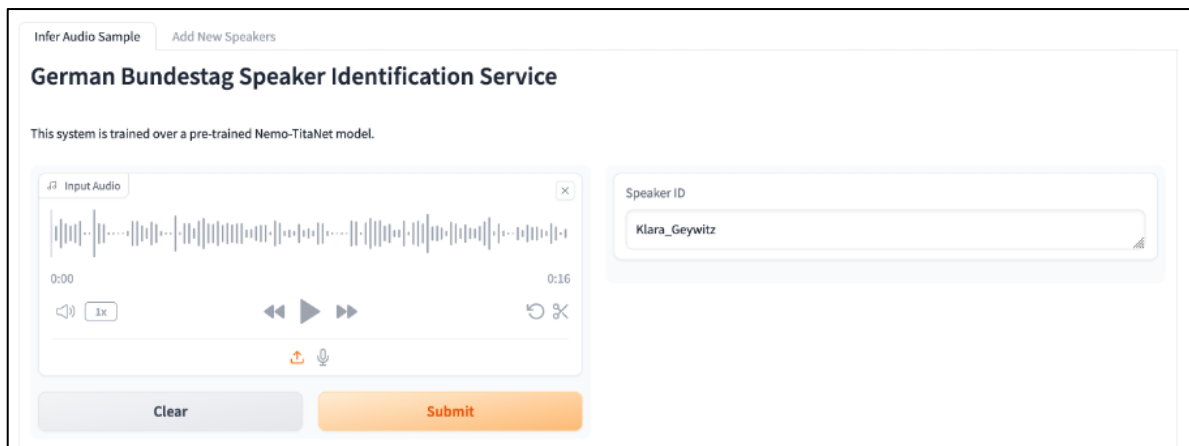


Figure 45 Interactive Dashboard of the Speaker Identification Service

In addition to identifying speakers from the existing speaker pool, the application provides a dynamic and adaptable framework that continually enhances its speaker recognition capabilities from the user feedback. Through the “Add New Speakers” tab, users can personalize and expand the model's database to recognize additional speakers, as shown in the following figure.

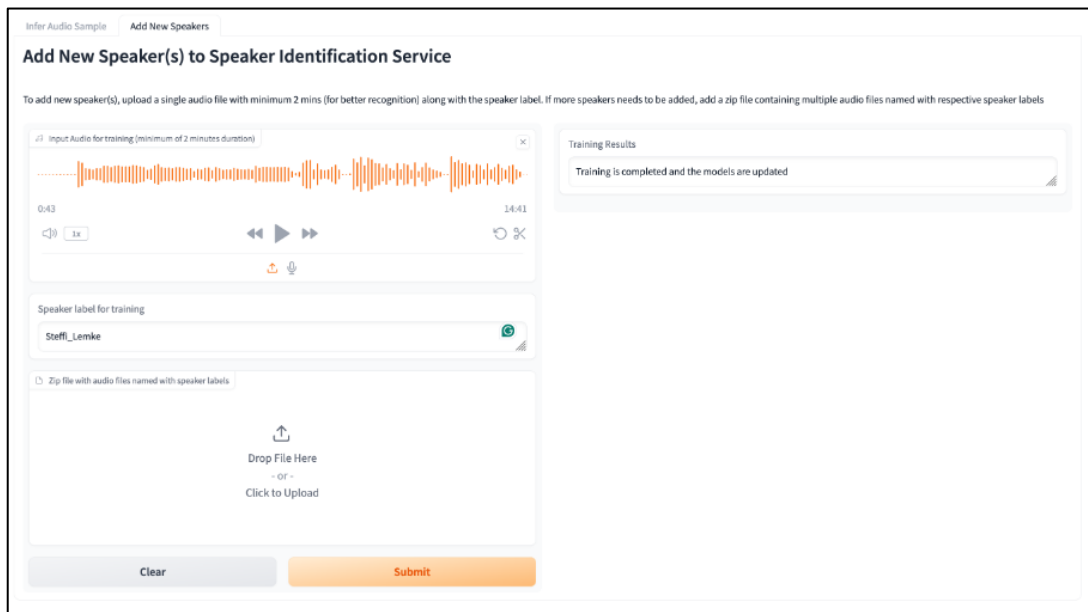


Figure 46 Enhancement Module Designed for Adding New Speaker Profiles to the System's Database

For optimal recognition, the speaker content should be around two minutes long, providing a substantial sample for the model to learn from. Accompanying the audio upload, users must input the 'Speaker label for training,' which typically includes the name or identifier of the new speaker. This label is critical as it tags the audio with the corresponding speaker's identity in the system. If there are multiple speakers to be added, users can upload an archive file containing several audio files. Each file should be named with the respective speaker labels to maintain organization and accuracy. After submission, the application processes the new audio data, integrating it into the existing model through retraining or incremental learning. This ensures the speaker recognition system updates the latest speaker profiles based on user feedback.

7.5 Diarization Prototype

Speaker diarization is the process of segmenting audio recordings by speaker labels and aims to answer the question “who spoke when?” SELMA's diarization prototype, as showcased in the Gradio app interface, serves as a sophisticated solution for the complex task of attributing speech to the correct speaker within an audio segment. This feature is indispensable for media analysis systems, elevating simple transcripts to rich, speaker-annotated dialogues.

In the prototype, users begin by entering a YouTube URL into the designated field. This is where the system will source the audio to be processed. Upon submission, the app extracts the audio from the YouTube video and begins the diarization process. This involves segmenting the entire audio stream into distinct parts, each associated with a different speaker. The results, as seen in the 'Speaker Segments' panel, are a series of time-stamped segments labeled with unique speaker identifiers (e.g., "spk_1", "spk_2"). These labels correspond to individual speakers detected in the audio, specifying their respective contributions' start and end times.

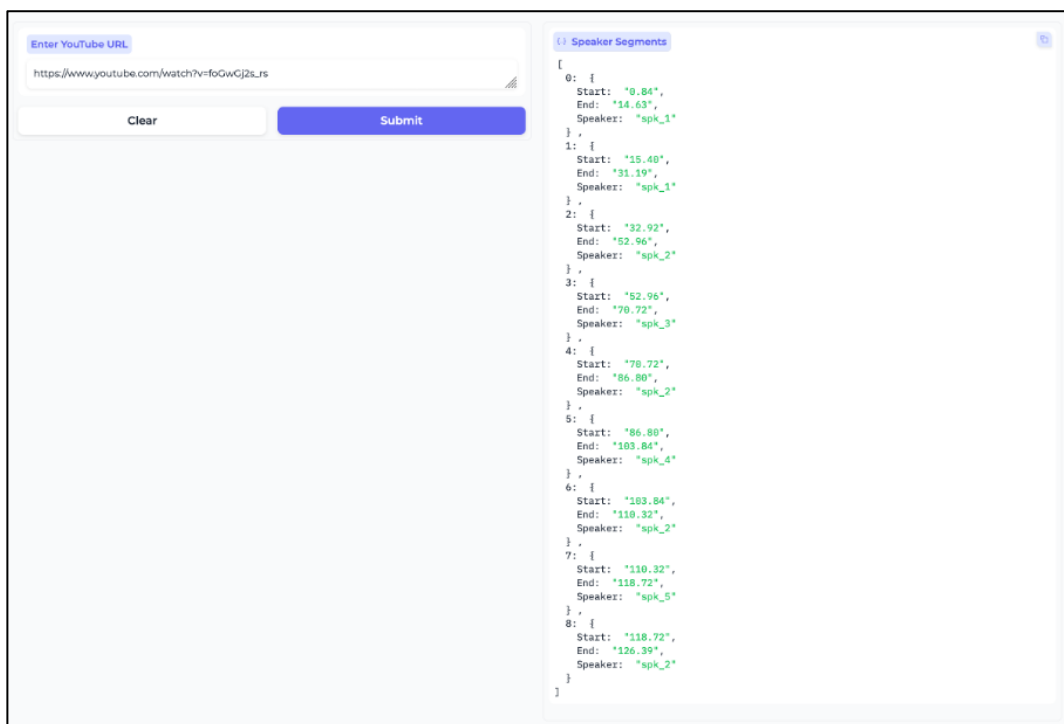


Figure 47 Interface for Speaker Diarization Displaying the Segmentation of a Sample YouTube URL

By integrating the "who spoke when" component into speech transcripts, this tool vastly enhances the utility of transcribed audio, providing a structured and detailed account of spoken content. It proves invaluable in various scenarios, from documenting legislative sessions and business meetings to analyzing broadcast media and academic research materials.

8. Requirement Implementation Status

In this section we present the list of requirements as defined in D1.1, updated in D1.2, D1.3 and finally updated here in D1.4, and the corresponding status in the UC1 and UC2 use cases as:

- Yes: Requirement Implemented
- No: Requirement Not Implemented and justified when applicable.
- Partial: Requirement Implemented with constraints, explained with a comment.

There are a few comments added below the requirements where additional references or explanations were needed.

These requirements were initially defined in the beginning of the project (D1.1) to help guide WP1 and broadly represent our direction. Over the project we have refined these requirements through agile development with testers and users in the loop, having frequent Zoom meetings and roadmap prioritization in Trello. Much work has been done which was not initially contemplated in the D1.1 requirements, such as a new Use-Case UC0 (described in D4.4), the multiple component specific prototypes reported in this document (e.g., Podcast Creator, MPHANTOM, ...) and new research directions which are described in WP2 and WP3. We have integrated SELMA's research output into two fully deployed products which are on the market, plain X and Monitio - this enables SELMA to bring the research done to real users and validate it in the market (More details in D6.6).

Further prototyping work done after the project will be guided by the needs of real users. Some next steps we have already planned are:

(1) for Monitio (UC1): Deeper entity network graph for complex domains (Pharma and Legal); Better explainability of analytics and visualizations through LLMs; Improved NER/NEL models for Pharma;

(2) for plain X (UC0): Language code switching; Improvement of automated subtitling rules; Language simplification (voiceover / L2 speakers); Translation quality estimation.

1. User & Platform Requirements

	Requirements	Use Case	MoSCoW	UC1 Status	UC2 Status
1	The System allows content to be ingested via standard interfaces as used by news organizations where available	1	Must	Yes	Yes
2	It is possible to use APIs to add content to the SELMA platform	1, 2	Should	Yes	Yes
3	The System ingests on-demand AV content in MP4 format	1,2	Must	No* (Only through streaming. Contemplated in UC2.)	Yes
4	The System monitors selected social media feeds	1	Should	Yes	-
5	The System scrapes news articles from websites	1	Must	Yes	-
Module Functionality					
6	The system processes content in the 30+ SELMA languages	1, 2	Must	Yes	Yes
7	The system transcribes from audio	1,2	Must	Yes	Yes
8	The system provides statistical analysis of ingested material	1	Must	Yes	-
9	The system provides automated translation of all SELMA languages into English as default	1	Should	Partial* (All except Dari [per])	-
10	The system provides automated translation into other SELMA languages upon request	1, 2	Should	No* (Available in a Cluster)	Yes

				Summary. Fully Contemplat ed in UC2.)	
Output					
11	The system offers the possibility to create dashboard user interfaces	1	Should	Yes	-
12	The system provides a summarization of individual media items in original language and/or English	1	Must	Yes	-
13	User selects ingested channels to monitor	1	Must	Yes	-
14	User subscribes to notifications when relevant content arrives	1	Must	Yes	-
Content					
15	The system uses clustering technology to group individual media items into over-arching high-level clusters	1	Must	Yes	-
16	The system provides a visualization which contains a list of all high-level news stories that are relevant to the user, according to the preferences they have set	1	Must	Yes	-
17	On detection of a high-level news story, the system provides a default name for this story based on the clustering technology parameters	1	Could	Yes	-

18	The system offers the user the ability to follow a specific story and subscribe to updates regarding that story	1	Could	Yes	-
19	The system allows the user to select a high-level news story and view the individual media items that are relevant to it	1	Must	Yes	-
20	For each high-level news story, the system displays a timeline. The system indicates where each individual media item fits on that timeline	1	Could	Partial* (Create view from story and Timeline in Trending Entities)	-
21	When viewing an individual media item, the user has the ability to link back to the over-arching high-level news story to which it is related (in order to view the other media items related to that cluster)	1	Must	Yes	-
22	The individual media items for a cluster continue to accrue for a selected period of time. However, the user can indicate to the system that the cluster is no longer relevant before this time has elapsed	1	Could	No	-
23	The system retains a record of clusters that it has identified along with an indication of how many	1	Must	Yes	-

	media items were identified by the system				
Entity Requirements					
24	The system uses entity identification technology to group individual media items by entities	1	Must	Yes	-
25	For each selected entity, the system displays a timeline	1	Must	Yes	-
26	The system carries out entity identification (person, organization, locations and events) using the original language and/or English translation	1	Must	Yes	-
27	The system provides a list of identified entities that are relevant to the user, according to the preferences they have set	1	Must	Yes	-
28	The system allows the user to highlight an entity identified by the system and view the individual media items that include the entity	1	Must	Yes	-
29	When viewing an individual media item, the user can link back to the entity to which it is related (in order to be able to view the other media items related to that entity)	1	Could	Yes* (The entities are shown in the Article Page. A manual copy and search is possible)	-
30	The system retains a record of entities identified along with an	1	Must	Yes	-

	indication of how many media items were identified as relating to those particular entities				
News Story Requirements					
31	The system uses preferences set by the user to detect news stories of interest to the user	1	Must	Yes	-
32	For each cluster, the system displays a timeline. The system indicates where each individual media item fits on that timeline	1	Should	Partial* (Create view from story and Timeline in Trending Entities)	-
Breaking News Alert Requirements					
33	The system provides breaking news alerts that will correspond to individual media items in accordance with the preferences set by the user	1	Must	Yes	-
34	A breaking news alert consists of a textual description of the associated media item in the original language and/or in English, along with some specified meta-data (such as date and time)	1	Must	Yes	-
35	The user selects their preferences for the type of clusters for which they wish to receive alerts (this	1	Must	Yes	-

	may be related to particular event or entity types)				
36	The user selects the manner and frequency at which they receive event alerts	1	Must	Yes	-
37	Breaking news alerts are as close to live as is technically possible	1	Must	Partial* (Alerts have been implemented through periodic Reports which do not run ASAP, but according to a periodicity)	-
General Functional Requirements					
38	The system monitors all input sources selected by the user	1	Must	Yes	-
39	The user can turn English translation on or off	1	Must	Yes	-
40	It is possible to associate a user with their team in the System	1, 2	Must	Yes	Yes
41	It is possible to indicate the role of a user in the System	1, 2	Could	No	Yes
42	A user can share a cluster or an individual media item with their team	1	Must	No* (Teams are planned for later)	-
43	Once a user has indicated that a particular news story or cluster is	1	Must	Partial*	-

	no longer relevant, individual media items relating to that entity or cluster can be removed from the user's view			(Users can create Views from Stories. These can be deleted on demand)	
44	The user can flag a particular individual media item, entity or cluster and its related media items and indicate that they wish to save them for future reference	1	Must	Yes	-
45	The user has an option in the system where they can view individual media items, entities or clusters which they chose to save (alongside all the individual media items related to these)	1	Must	Yes	-
Media Item Requirements					
46	The system provides a clear visual indicator as to the nature of an individual media item (social media, blog, website, AV etc.)	1	Must	Yes	-
47	The user can view the detail of an individual media item (when applicable)	1	Must	Yes	-
48	For an individual AV media item, the user views the video and its original transcription, a translation in the prespecified languages (where applicable) and the meta-data associated with the item	1, 2	Must	Yes	Yes

49	For individual AV media items, the system supports a player and editor with tools to 'scrub' through the video, rewind and download (UC2)	1, 2	Must	Yes	Yes
50	For individual AV media items, the system supports a player and editor with tools to select elements to 'clip'	1, 2	Could	No	No
51	It is possible for the user to 'clip' an individual AV media item by means of text selection from the transcript	1, 2	Could	No	No
52	For other media occurrences (i.e. textual), the user views the text, its translation in the prespecified language (where applicable) and any meta-data associated with the item	1, 2	Must	Yes	Yes
53	The system provides a 'confidence level indicator' which will indicate how strongly an individual media item relates to the existing or suggested clusters	1	Must	Partial* (The top 3 documents with most confidence are shown.)	-
User Preference Requirements					
54	It is possible to set up a set of default sources that will be frequently used by a particular team	1, 2	Must	Yes	Yes

55	The system contains a predefined list of sources by region	1	Must	Yes	-
56	The user can specify entities of particular interest to them	1	Must	Yes	-
57	The user can choose their region of interest in the System	1	Must	Yes	-
58	The user can prioritize countries of interest within their region of interest	1	Must	Yes	-
59	The system contains a predefined list of regions and countries	1	Must	Yes	-
60	It is possible for the user to set time parameters in the system using an indicator such as a time slider to indicate the time period in which they are interested	1	Must	Yes	-
61	In general, the system supports input of preferences in a number of ways: From predefined lists, using data being encountered in the system (the system will create new entities, events etc.), using free-format text (i.e. search boxes)	1	Must	Yes	-
62	The system is adaptable and configurable to the user's preferences	1	Should	Yes	-
Administration Requirements					
63	The user can log into the system	1, 2	Must	Yes	Yes
64	The user can log out of the system	1, 2	Must	Yes	Yes

65	The system supports a super user account	1, 2	Must	Yes	Yes
66	The system supports an administrative user (for account management)	1, 2	Must	Yes	Yes
67	The administrator can create teams in the system	1, 2	Should	Yes	Yes
68	The administrator has typical administrator rights including add new user, remove users, update user profiles as well as the ability to manage data held by the System	1, 2	Should	Yes	Yes
Search Requirements					
69	It is possible for the user to conduct a search based on an entity or entities	1	Must	Yes	-
70	It is possible for the user to search based on the type of the media item (e.g. social media, AV etc.)	1	Must	Yes	-
71	It is possible for users to search based on event types	1	Won't	No	-
72	It is possible to take a screen shot (or still frame) from an individual AV media item (rights to be considered here)	1, 2	Could	No	No
73	Entity search is able to handle a range of variable spellings for the same entity	1	Must	Yes	-

74	It is possible to train the system in relation to alternative spellings for searches. For example, it should be possible to link to alternative spellings and indicate that they relate to the same thing	1	Won't	No	-
Input Source Requirements					
75	The system informs the user if a source stops broadcasting	1	Could	No* (This information is available in the backstage only)	-
76	The system informs the user if the frequency at which a channel is broadcast, changes	1	Could	No* (This information is available in the backstage only)	-
Trend Analysis Requirements					
77	The user has the ability to set their preferences in the system with regard to the types of trend analysis they wish to see in the system	1	Must	Yes	-
78	The system offers the user various options around trend analysis including maps incorporating hotspots, graphs and timelines showing hotspots	1	Must	Yes* (There are many possible Dashboards and Reports which can	-

				be constructed. The particular case of “hotspots” was deprioritized)	
79	The system utilizes saved clusters and media items (that the system has saved along with the number of media items) to conduct trend analysis over a period longer than one week	1	Could	Yes	-
Generate Voice-Over					
80	The system generates a voice-over on request for individual AV media items	2	Must	-	Yes
81	The user chooses whether the voice-over is performed in the original or in the selected translation language	2	Must	-	Yes
82	The system provides a list of available synthetic voices for the user	2	Must	-	Yes
83	The user can choose and/or change which synthetic voice is used for the voice-over	2	Must	-	Yes
84	The user can change the synthetic voice per segment	2	Must	-	Yes

85	The user can amend the voice-over output including phonetics, pauses and pitch	2	Could	-	Yes
Edit Transcription/Translation					
86	The user can edit the transcribed text	2	Must	-	Yes
87	The user can edit the translated text	2	Must	-	Yes
88	The user saves the edited versions of the transcribed and/or translated text	2	Must	-	Yes
89	The user changes the engine and perform the transcription again for the whole text and/or by segment	2	Should	-	Yes
90	The user changes the engine and perform the translation again for the whole text and/or by segment	2	Should	-	Yes
System Learning and User Feedback					
91	The system is trained by the user. The system offers the user a selection of news stories and individual media items relevant to a cluster and the user accepts/rejects them as necessary, thus training the system	1	Must	No This requirement was deprecated after user evaluation	-
92	The system is trained by the user. The system offers the user breaking news alerts and the user accepts/rejects as necessary –	1	Must	No This requirement was deprecated after user evaluation	-

	training the system to meet the user's preferences				
93	The system learns from the user's corrections and applies them throughout the text	2	Must	-	Partial* (This functionality is available through the MPHANTO M UI (See D3.8) and the improved models are compatible with UC2 plain X)
94	The system applies corrections on different levels (current and future items) based on the preferences set by the user	1, 2	Should	Yes	Yes* (Read #93)
95	The system learns from the user's corrections for entities	1, 2	Must	Yes	Yes* (Read #93)
Diversity Detection					
96	The system identifies the binary gender associated with the author's name (if present) of an individual media item	1	Should	No* (Limited due to feedback of Ethical Committee)	-
97	The system provides the number of times each binary gender is mentioned in the media items	1	Must	Yes* (Only in internal demonstrator).	-

				Limited due to feedback of Ethical Committee)	
98	The system identifies the gender of the protagonist in each individual media item (if applicable)	1	Should	No* (Limited due to feedback of Ethical Committee)	-
99	The system provides the number of times each binary gender is mentioned in each topical cluster	1	Must	Yes* (Only in internal demonstrator. Limited due to feedback of Ethical Committee)	-
100	The system provides a visualization of the gender analysis	1	Must	No* (Limited due to feedback of Ethical Committee)	-
101	The system provides all information pertaining to the diversity data (gender, age, sexual orientation, country of citizenship, medical condition) found on the Wikidata entry of relevant public figures where applicable	1	Must	Yes* (Only in internal demonstrator. Limited due to feedback of Ethical Committee)	-

102	The system identifies the gender associated with the named entities of type Person, even if the gender information is not available in Wikidata (or the entity is not in Wikidata at all)	1	Should	No* (Limited due to feedback of Ethical Committee)	-
-----	---	---	--------	---	---

Table 2 User & Platform Requirements

2. Technical Requirements

	Technical Requirements	Use Case	MoSCoW	UC1 Status	UC2 Status
Platform - Orchestration					
P1	Orchestrates processing jobs on the data stream, automatically	1	Must	Yes	-
P2	Orchestrates processing jobs on the data stream, on user request	1,2	Must	No	Yes
P3	Allows listening for job results via push notifications (e.g, web sockets)	1,2	Should	Yes	Yes
P4	Allows listening for job state changes (errors, job progress)	1,2	Must	Yes	Yes
P5	Allows consulting the state of a job request / jobs on an item via an API (e.g., REST)	1,2	Must	Yes	Yes
P6	Accepts new “job workflow” requests, which may entail running several jobs organized in a graph of dependencies, on an item	1,2	Must	Yes	Yes
P7	Orchestrated jobs are eventually applied, meaning, a job cannot be	1,2	Must	Yes	Yes

	lost – it is either completed successfully or logs an error				
P8	Orchestration-related configuration changes happen without downtime	1,2	Could	Yes	Yes
P9	Resilience to the unavailability / downtime of specific workers. Jobs wait until the worker recovers	1,2	Should	Yes	Yes
P10	The system allows parallel jobs to execute on the same item if they can be run that way according to the orchestrated job graph	1,2	Could	Yes	Yes
Platform - Worker management					
P11	Processing workers use Docker (or equivalent) containerization for deployment	1,2	Must	Yes	Yes
P12	The system is prepared for kubectl-compatible deployment in Kubernetes clusters <i>(* This direction has changed, since the Consortium is using the alternative “docker-spaces” as described in D4.3)</i>	1,2	Should <i>(Won't*)</i>	No	No
P13	The system manages the lifetime of different worker containers <i>(Comment: Through docker-spaces as described in D4.3)</i>	1,2	Could	Yes	Yes

P14	The system scales the number of worker containers according to the corresponding task queue flux	1,2	Could	Partial* <i>(Through docker-spaces as described in D4.3)</i>	Partial* <i>(Through docker-spaces as described in D4.3)</i>
P15	Worker-related configuration changes (new workers, worker scaling, etc.) happen without downtime <i>(Comment: Through docker-spaces as described in D4.3)</i>	1,2	Could	Yes	Yes
Platform - Replication and Sharding					
P16	The system allows replication at the worker level	1,2	Must	Yes	Yes
P17	The system allows replication at the orchestration controller level <i>(Comment: Through docker-spaces as described in D4.3)</i>	1,2	Should	Yes	Yes
P18	The system allows replication at the database level <i>(Comment: As described in D4.3, docker-spaces is currently stateless, and so it doesn't need replication. The core systems of UC1 and UC2 use PostgreSQL though, which supports replication through repmgr and pgbouncer)</i>	1,2	Should	Yes	Yes
P19	The system allows sharding at the orchestration controller level	1,2	Should	Yes	Yes

	<i>(Comment: Through docker-spaces as described in D4.3)</i>				
P20	The system allows sharding at the database level	1,2	Should	Yes	Yes
Component - Online News Classification and Clustering					
C1	For each ingested news item, the system attributes a cluster	1	Must	Yes	-
C2	For each ingested news item, the system attributes an IPTC topic	1	Must	Yes	-
C3	The system clusters documents in an online fashion, e.g., without having to revisit all past decisions	1	Must	Yes	-
C4	The system clusters documents natively in all 30 SELMA languages	1	Should	Partial* (Missing 7: am,bn,bs, per,ha,sw,ps. Only limitation is the embedding model which can be replaced.)	-
C6	The system leverages user feedback on clustering decisions to improve future decisions	1	Should	Not yet	-
Component - Summarization					
S1	For each ingested news item, the system generates a summary	1	Must	Yes	-
S2	The system generates summaries natively in all 30+ SELMA languages	1	Should	Partial* (Missing 7: am,bn,bs,	-

				per,ha,sw,ps. Only limitation is the embedding model which can be replaced.)	
S3	The system generates summaries either from original text article or video transcripts	1	Should	No	-
S4	The system leverages user feedback on summarization results to improve future summaries	1	Should	No	-
Component - Machine Translation					
M1	The system translates a textual document on demand	1,2	Must	Partial* (Available in a Cluster Summary. Fully Contemplated in UC2.)	Yes
M2	The system translates a video on demand	1,2	Must	No* (Only through streaming. Contemplated in UC2.)	Yes
M3	The system translates between all 30+ SELMA languages	1,2	Should	Partial* (All except Dari [per])	Yes
Component - Automatic Transcription					
R1	The system automatically transcribes an ingested video or audio file	1,2	Must	Yes	Yes

R2	The transcription is enriched by punctuation	1,2	Must	Yes	Yes
R3	The transcription is enriched by speaker information	1,2	Should	No	Yes
R4	The transcription is enriched by named entity labeling	1,2	Must	Yes	No
R5	The system transcribes all 30 SELMA languages	1,2	Should	Yes	Yes
Component - Entity Recognition and Linking					
E1	For each ingested news item, the system detects named entities	1	Must	Yes	-
E2	For each ingested news item, the system links named entities to a knowledge base	1	Must	Yes	-
E3	The system links entities natively in all 30 SELMA languages, leveraging crosslingual representations	1	Should	Partial* (All except Dari [per])	-
E4	For each ingested news item, the system attributes a gender for each person named entity detected in the news item. <i>(* Yes, for named entities of type people that can be linked to Wikipedia/Wikidata)</i>	1	Should	Yes* (Only in internal demonstrator. Limited due to feedback of Ethical Committee)	-
Component - Story Segmentation					
G1	Each ingested long audio segment gets split into meaningful units	1,2	Could	No* (We developed a standalone SELMA	No

				Prototype instead)	
G2	Speaker clustering is used to create speaker independent units	1,2	Could	No* (We developed a standalone SELMA Prototype instead)	Yes
G3	Speaker recognition automatically identifies the original speaker in each segment	1,2	Could	No* (We developed a standalone SELMA Prototype instead)	No
G4	Topic segmentation is used to separate by spoken content	1,2	Could	No* (We developed a standalone SELMA Prototype instead)	No
Component - Voice Conversion Synthesis					
V2	The text-to-speech system will be improved to better handle foreign words	2	Must	-	Partial* (the TTS system seems to limit the loss of performance due to the presence of foreign words, but these words were not so present in the evaluation data to get

					statistically significant results)
V1	The text-to-speech system automatically produces voices in Latvian, German, and French	2	Must	-	Partial* (Changed to Urdu, Portuguese and Latvian)
V3	A speech-to-speech translation system works at least on one language pair	2	Should	-	Partial* (A research component is available but was not integrated)
V4	A speech-to-speech translation system works at least on one language pair and can generate a synthetic voice in the target language close to the natural voice in the source language	2	Could	-	Yes for French-to-English

Table 3 Technical Requirements

9. Conclusion

This document presents the current state of the prototypes which address the three main use cases, Multilingual Media Monitoring (UC1), Multilingual News Content Production (UC2) and SELMA NLP Service Orchestration (UC0). Two external use cases which integrate SELMA technology are also discussed – PiniTree and the Podcast Creator.

Within Use Case 1 (UC1), we are integrating results from the SELMA research tasks into the *Monitio* product, a Media Monitoring platform under development by Priberam, available at <https://app.monitio.com>. Within Use Case 2 (UC2), we are integrating results from SELMA into the *plain X* product, a Multilingual News Media Content Production platform under development by Priberam and Deutsche Welle, available at <https://app.plain-x.com>. The SELMA Open-Source Platform (UC0) is available at <https://selma-project.github.io/>.

The list of requirements from D1.2 is revisited and the corresponding progress reported.

Bibliography

- Barzdins, G., Gosko, D., Cerans, K., Barzdins, O. F., Znotins, A., Barzdins, P. F., Gruzitis, N., Grasmanis, M., Barzdins, J., Lavrinovics, I., Mayer, S. K., Students, I., Celms, E., Sprogis, A., Nespore-Berzkalne, G., Paikens, P. (2020b). Pini Language and PiniTree Ontology Editor: Annotation and Verbalisation for Atomised Journalism. *In: ESWC 2020 Satellite Events. LNCS, Volume 12124, pp. 32-38.*
- Peteris Paikens; Guntis Barzdins; Afonso Mendes; Daniel Ferreira; Samuel Broscheit; Mariana S. C. Almeida; Sebastiao Miranda; David Nogueira; Pedro Balage; Martins, Andre F. T. (2016a). SUMMA at TAC Knowledge Base Population Task 2016, *DOI: 10.5281/zenodo.827317*
- Znotiņš, Artūrs & Barzdins, Guntis. (2020). LVBERT: Transformer-Based Model for Latvian Language Understanding. *Baltic HLT, IOS Press, pp. 111-115, DOI 10.3233/FAIA200610.*
- Znotins A, Cirule E. (2018). NLP-PIPE: Latvian NLP Tool Pipeline. *In: Human Language Technologies - The Baltic Perspective. vol. 307. IOS Press; 2018. p. 183–189.*
- Paikens P. (2016b). Deep Neural Learning Approaches for Latvian Morphological Tagging. *In: Baltic HLT; 2016. p. 160–166.*
- João Santos, Afonso Mendes and Sebastiao Miranda, “Simplifying News Clustering Through Projection from a Shared Multilingual Space” in Text2Story at ECIR, 2022.
- Mikus Grasmanis, Pēteris Paikens, Lauma Pretkalnina, Laura Rituma, Laune Strankale, Artūrs Znotiņš, and Normunds Grūzītis. (2023). Tēzaur.lv – the experience of building a multifunctional lexical resource. *In Electronic lexicography in the 21st century (eLex 2023): Invisible Lexicography. Proceedings of the eLex 2023 conference, pages 400–418. Lexical Computing CZ s.r.o. <https://elex.link/elex2023/wp-content/uploads/89.pdf>*

Annex 1 – plain X User Guide



V – 2024.03.01

Table of Contents

What is plain X?	3
Accessing the Platform	3
Platform navigation	4
Sidebar	4
Header	4
Sub-header	4
Page breakdown	6
Home	6
Library	6
Boards	7
Personal Board	7
Shared Board	8
Review Board	9
Task pages	10
Transcription	10
Translation	12
Subtitling	15
Voice-over	17
Account	20
Settings	20
Personal	20
User Roles & Permissions	21
Workspace	23
Users	23
Consumption	26
Teams	26
Repositories	29
Customization	31
Templates	31
Workflows	32
Simple Editorial Workflow	32
Add New Media	32
Task Creation	36
Standard, recommended procedure:	39
Collaborative translation task procedure:	39
Sequential, simplest procedure:	39
Versioning	44
Text File Processing	46
Collaborative Editorial Workflow	47
Assigning Tasks and Reviews	49
Editor's View	49
Reviewer's View	52

What is plain X?

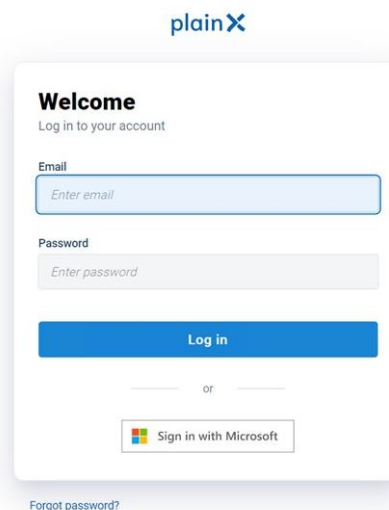
plain X is a platform that allows users to update video, audio, and text files, and automatically transcribe, translate, subtitle, and voice-over in over 100 languages. The following guide will lead you through how to use this software.

Accessing the Platform

In order to use the plain X go to <https://app.plainx.com> and login using the email and password registered with plain X.

Best Practice: Browser

As of now plain X is only fully supported on Chrome or Edge. By proceeding with using another browser your experience may not be optimal.



The screenshot shows the login interface for plain X. At the top, the plain X logo is displayed. Below it, the heading "Welcome" is followed by the instruction "Log in to your account". There are two input fields: "Email" with a placeholder "Enter email" and "Password" with a placeholder "Enter password". A blue "Log in" button is positioned below the password field. Underneath the button, the word "or" is centered between two horizontal lines. Below this, there is a "Sign in with Microsoft" button featuring the Microsoft logo. At the bottom left of the form, there is a link for "Forgot password?".

Platform navigation

Sidebar

On the left, you can see the sidebar where the Home, Library, and board pages are located. In the Recent section, you can access the media items you recently worked on. On the bottom of the sidebar, you will find the Settings menu with Personal and Workspace settings.

Header

On the header, you can see the name of your current workspace, and if you have more than one, this is where you change between them. You can also see notifications and access your user menu. There you can access the account settings, the help menu, and logout.

Sub-header

The sub-header will change according to the page you are on. On it, you will find filtering options and actions that will aid you in your work.

Details



Display Name

Nika K

Email

veronika.koshtur@priberam.pt

Timezone

Europe/Lisbon

Locale

Portugal

Language

English

Save

Change Password

New password

New password

Confirm password

Confirm password

Save

Workspaces

Name	Role
Priberam.QA	Workspace Manager

Page breakdown

Home

Home is your starting place and where you can quickly start or continue your work. On this page, you can create new tasks, add new media items, and access the recent tasks you were working on.

Library

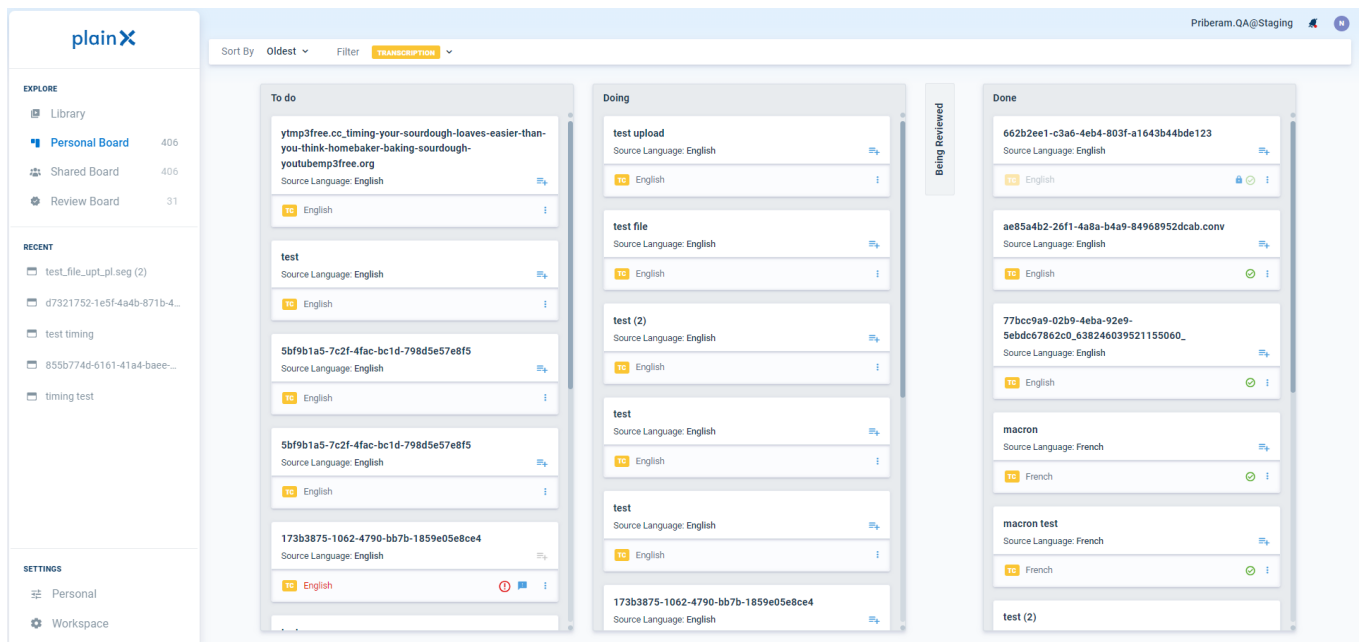
Library is the place where you can see and manage all your media items. Here you can add, share, and remove files. To make the navigation easier you are also able to filter by repository, language, and search for a specific name.

Your removed files are kept in the Recycle Bin until you delete them permanently. To toggle between the Library and the Recycle use the dropdown on the right of the sub-header.

Boards

Personal Board

You can access your **Personal Board** using the sidebar. Once you open the page you can find all tasks that have been assigned to you. To make it easier to find what are you looking for, plain X allows you to sort and filter the tasks. You can do that by using the options in the sub-header.



Shared Board

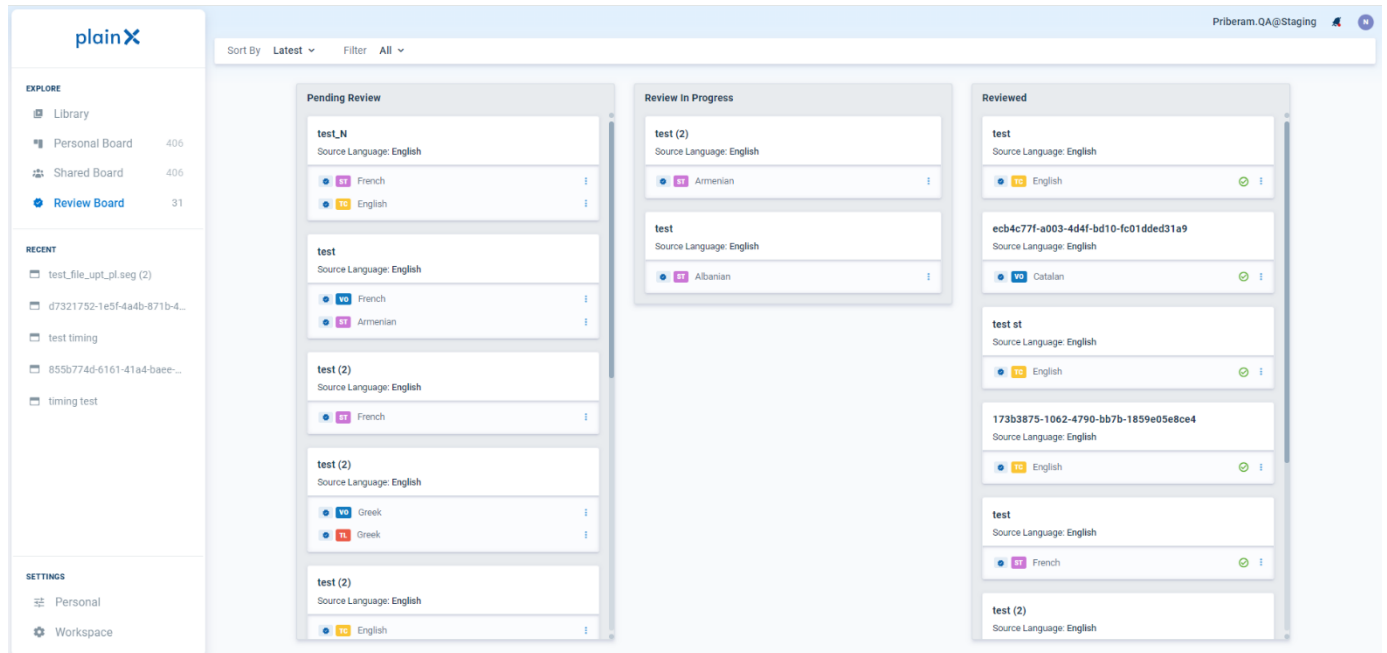
On the **Shared Board**, you can see the status of the tasks your team(s) has assigned to them. If you are not an editor of certain tasks, you won't be able to access them, only see their status. On this page, you can also order and filter the results by using the menus on the sub-header.

The screenshot displays the plainX Shared Board interface. At the top, there are navigation menus for Repository (All), Teams (All), Sort By (Latest), and Filter (All). The user is logged in as Priberam.QA. The interface is divided into several sections:

- EXPLORE:** Includes Library, Personal Board (79 items), Shared Board (79 items), and Review Board (7 items).
- RECENT:** Lists recent test cases such as fa850d6c-1f07-4599-9b8b-..., almost perfect test case, dependencies_aut, d4c825cd-ee1a-4697-9a04-..., and napoleon.
- SETTINGS:** Includes Personal and Workspace settings.
- Task Board Columns:**
 - To do:** Contains tasks like 'napoleon' (English, ST), 'dependencies_aut' (Ukrainian, ST), 'd4c825cd-ee1a-4697-9a04-2826be171fa0' (English, ST), and another instance of 'd4c825cd-ee1a-4697-9a04-2826be171fa0' (English, ST).
 - Doing:** Contains tasks like 'dependencies_aut' (Ukrainian, TC), 'fa850d6c-1f07-4599-9b8b-c1e75a241905' (Ukrainian, TC), 'almost perfect test case' (English, ST), and 'verticaltest' (Portuguese, ST).
 - Being Reviewed:** An empty column.
 - Done:** Contains completed tasks like 'napoleon' (English, TC), 'dependencies_aut' (Ukrainian, TL), 'd4c825cd-ee1a-4697-9a04-2826be171fa0' (English, TC), 'fa850d6c-1f07-4599-9b8b-c1e75a241905' (Ukrainian, TL), and 'd4c825cd-ee1a-4697-9a04-2826be171fa0' (English, TL).

Review Board

Review Board contains the tasks assigned to you for review. The tasks on this page can be ordered and filtered by using the menus in the sub-header.



Task pages

Transcription

To access the transcription page, click on a completed transcription task on the media item modal or in the recent section on the Home page. You can also select a task from the **Personal** or **Shared board**.

On the top of the page, you will see a sub menu, that allows you to interact with the task.

In the *File* menu, you can save a specific version of the task, open an already existing one, or export the done task. For the transcription task, the available export formats are .txt and .srt. You can set the .txt export options including or excluding editorial notes, comments and timecodes.

In *Settings* you can see a history of the saved versions and manage permissions for the task, choosing editors and reviewers.

View defines the structure of the page. On the view menu, you can enable compressed timeline, editorial notes, and speaker diarization options.

Under the *Help* dropdown, you will find a help menu with available keyboard shortcuts.

On the left, you will see a video/audio preview. Next to it, in a column on the right you will find transcribed text.

The language of the transcription is indicated on the left corner of the ribbon which appears right above the transcription column.

If you are not satisfied with the outcome of the transcription, you can set new parameters for it under the settings icon. There you can select a service provider and variant, enable the diarization option and upload a transcription file, select segmentation mode. Click *Reload Service* or *Re-run segmentation* accordingly after setting the desired parameters.

Remember, that all previous input and corrections will be gone after a reload. This option is only possible at the full document level, and rerunning transcription per segment is not possible at this step.

Translation

To access the translation page, click on a completed translation task on the media item modal or in the recent section on the Home page. You can also select a task from the **Personal** or **Shared board**.

On the page, you will see a sub menu, that allows you to interact with the task.

It is similar to the one on Transcription page, with an exception for the File menu.

Here you import an aligner for the translation and export the task in .txt format.

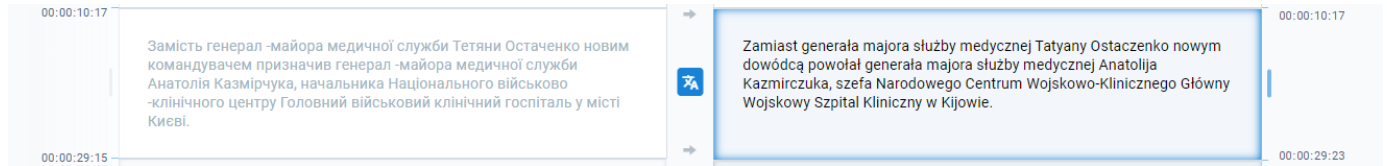
You can limit access to the task by clicking the lock icon on the right of the sub header. Currently, locking locks all related tasks. Avoid locking if there are simultaneous tasks or users.

Here you can also change the task status marking it as *Done* or *Ready for a Review* after the editing process is finished and the quality of the translation is satisfactory. Try to always make sure you set the translation as Done before proceeding to the next step (subtitling, voice-over, or export).

On the ribbon, you will find two language names indicating accordingly the language of transcription and the language of translation. You can also search for a specific word or sentence in the translation text by clicking on the search icon.

If you are not satisfied with the translation quality, you can change the service provider and re-translate the text. To change service provider, click on the settings icon on the ribbon, select the provider, and click Translate all. Remember, that during this process all previously added comments and editorial notes, as well as other previously made changes will disappear.

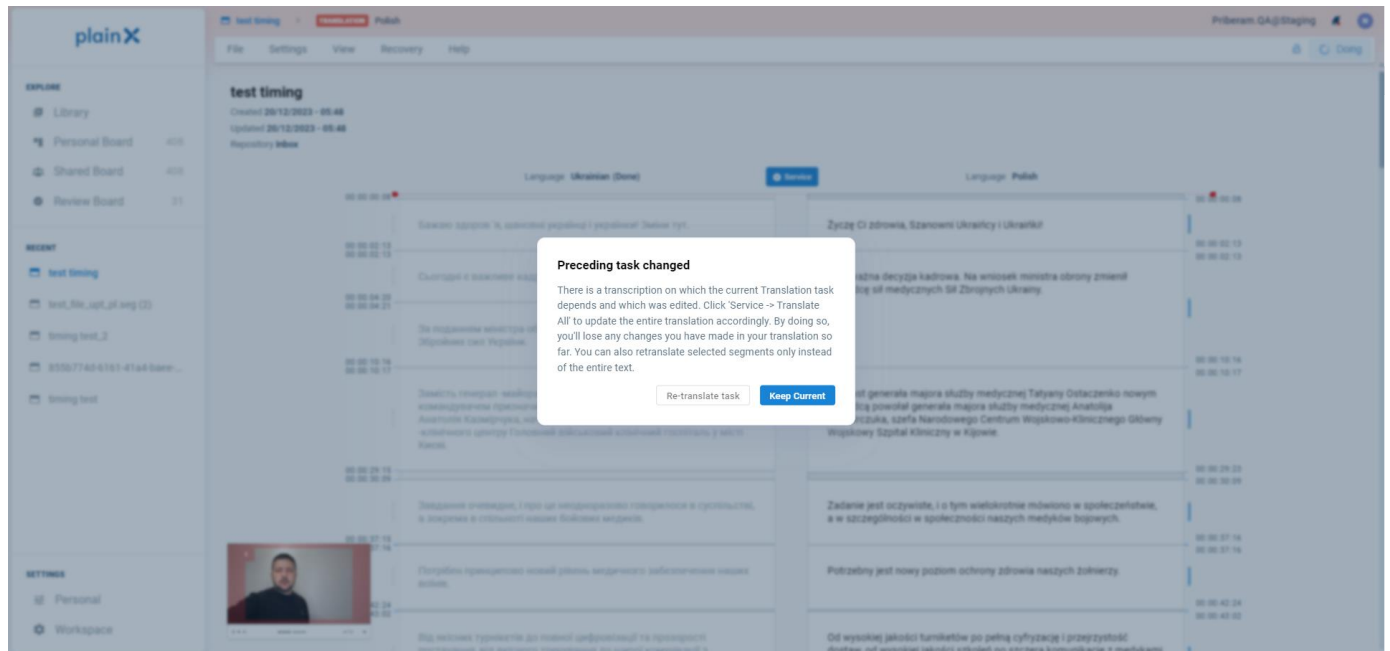
You can also rerun selected segments by clicking on the translation icon between them.



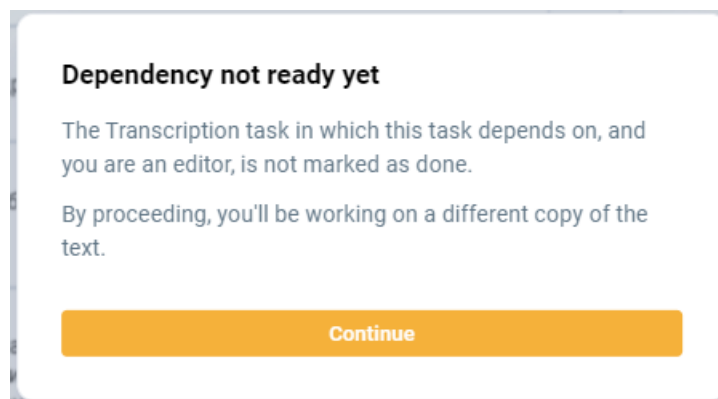
If the translation was created before the transcription was edited and set to *Done*, you will receive a message, suggesting to update the translation view to the latest version of the transcription. The source language column will then be shown as greyed out and will be no longer editable.

Preceding task changed

There is a transcription on which the current Translation task depends and which was edited. Click 'Service -> Translate All' to update the entire translation accordingly. By doing so, you'll lose any changes you have made in your translation so far. You can also retranslate selected segments only instead of the entire text.



If the transcription was not set to ✔ Done open the translation task, you will see a message warning you that you will be working on a different version of the transcription. If you are an editor in both transcript and translation, however, your changes in the transcription will be automatically applied in your translation view.



Subtitling

To access the subtitling task page, click on an uploaded subtitling task on the media item modal or in the recent section on the Home page. You can also select a task from the **Personal** or **Shared board**.

The sub header of this page is similar to the other task pages; however, the export options are different. You can export the task in two different ways: video with subtitles or subtitles only. Video export is available in .mp4 format and the subtitles-only option has several formats available, which are:

- WebVtt (.vtt)
- SubRip (.srt)
- Plain text (.txt)
- AVID (.avid)
- EBU (.stl)
- AdvSS (.ass)

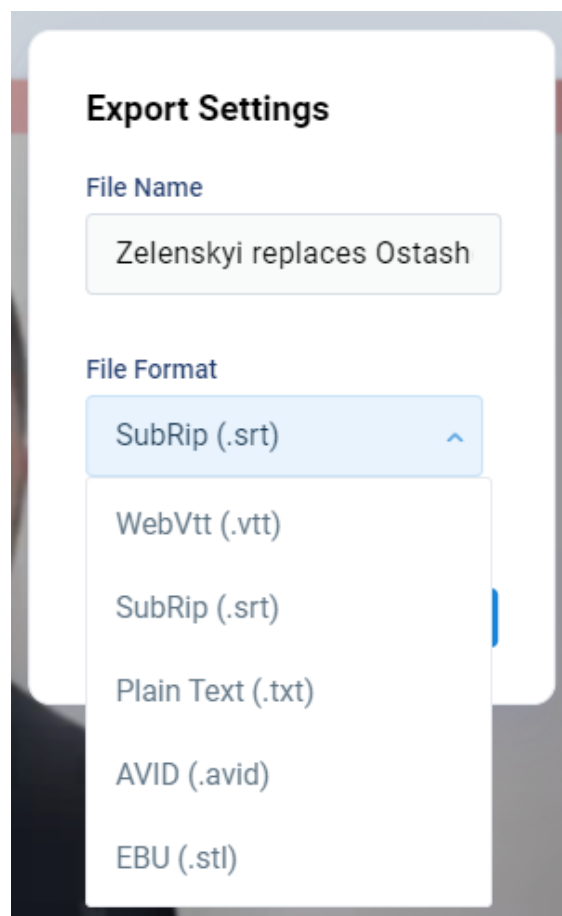
In the Subtitling view, the video will appear on the left side of the screen. When subtitles are active, a red safe area appears around the image. That frame shows which areas of the image have to stay clean. Changes to the subtitles are displayed immediately in the video player.

The subtitles will appear in a column on the right side. The text in the column is segmented as it should appear on the video. The language of the subtitles is indicated on the left corner of the ribbon which appears right above the column.

You can re-apply subtitling segmentation or change the applied template either for the full text or by a segment. To apply the changes to the whole document, click the settings icon located on the right corner of the ribbon and select wanted options. Click *Set base template* or *Reapply subtitling* accordingly.

To apply the changes to the text segment, click on the settings icon located on the right corner of the text segment.

Once the quality of the subtitles is satisfactory, you can set the task as Done and export your subtitles.



Voice-over

To access the voice-over task page, click on an uploaded voice-over task on the media item modal or in the recent section on the Home page. You can also select a task from the **Personal** or **Shared board**.

The video appears on the left side of the screen with the voice-over text on the right. You can edit this text - it is an independent version and will not overwrite the (written) translation or subtitling text.

If you are not happy with the selected voice, you can choose a new voice-over service for the video by clicking on the settings icon on the ribbon to rerun/replace the voice-over in the entire video with the newly selected voice. After changing the voice or editing the text services, you will need to press the *Rerun* button to apply the changes.

If you need to use more than one voice in your video, you can select a voice per segment. Click on the settings icon at the right-down corner of the text segment and select a new voice, then click on the re-load icon to apply it.

Enhance Voice-Over

In addition to choosing different voices, you can also change certain aspects of the selected synthetic voice, for instance, to emphasize a phrase, change the intonation, insert a break, or correct the pronunciation.

Double-click one or more word(s) to make changes and select one of the aspects to be changed. After choosing the enhancements within a segment, re-run the segment to apply the changes.

Change the pronunciation:

- With **Alias** you can have the word(s) replaced by one or more spoken words. You can also use this to correct the pronunciation, giving it the function of "pronounce as".
- **Phoneme** enables the use of phonetic spelling according to three major phonetic systems, the most important one being IPA (International Phonetic Alphabet). This should only be used if you are proficient in phonetic spelling, you need to know the principles and characters to be used in phonetic spelling for each target language specifically. Phonemes can be applied to single words or phrases. The Phoneme option may not work on all voices.
- **Reading Rule** is used to make sure numbers, times, dates, telephone numbers, e-mail addresses, URLs, etc. are correctly pronounced.

Reading rules can be applied to single words or phrases. Reading Rules may not work on all voices.

Change the Intonation:

Select a set of words and then select the tab Intonation. Enter parameters referring to position and strength.

For example, a series like (0%,+20%) (50%,+30%) (100%, -10%) means to increase the pitch 20% at the beginning (position 0%), increase to +30% in the middle (position 50%) and decrease the pitch to -10% at the end (position 100%).

Intonation can only be applied to sentences, not single words. Intonation may not work on all voices.

With Prosody we can change the Rate, Pitch, and Volume:

Rate, Pitch, and Volume can only be applied to sentences, not single words. It may not work on all voices.

Break:

To insert a break between two words, first select the words, then click on the *Break* option. Select the break strength, ranging from X-Weak to X-Strong.

After all changes to the voice-over have been inserted, press the *Rerun* button in the video to ensure all the changes are implemented in the video or audio file. Export the file in the desired format.

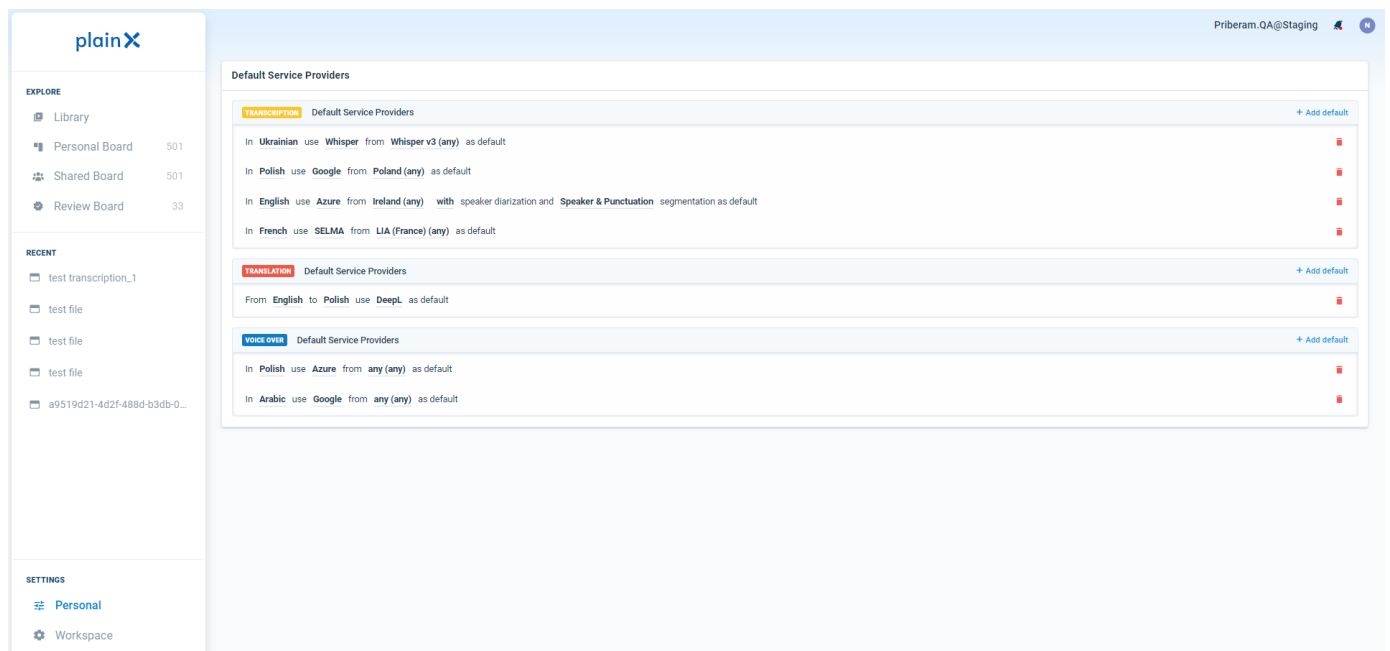
Account

To access your account page, click on the user thumbnail on the top right of the header, and select Account. There you can set a new password or change basic information such as name, time zone, location, and platform language.

Settings

Personal

On Personal you can specify default engines for every task type and language. It will help you to simplify the process of task creation and will be especially useful for often-used task/language combinations.



To add a default engine for a specific task, click on [+ Add default](#) on the right side.

Select for which language you want to set a default engine, which provider, and which service variant. With some languages, you also have access to expanded functions, such as with/without punctuation option and diarization.

User Roles & Permissions

Depending on your role or access level there are different functionalities available for you. See the table on the next page to see which functionalities are available for each role.

Role	Administrator	<u>Workspace Manager</u>	<u>Manager</u>	<u>Team Leader</u>	<u>Editor</u>	<u>Guest Editor</u>	<u>Viewer</u>
Create Tasks / Delete Tasks	✓	✓	✓	✓	✓		
View Items	✓	✓	✓	✓	✓	✓	✓
Add Media Items	✓	✓	✓	✓	✓		
Do Tasks	✓	✓	✓	✓	✓		
Create Repositories	✓	✓	✓	✓			
Do Reviews	✓	✓	✓	✓	✓		
Add other Teams to tasks	✓	✓	✓	✓			
Change task settings	✓	✓	✓	✓ (Own Teams)			
Marks as Final	✓	✓	✓	✓ (Own Teams)			
Create Teams	✓	✓	✓				
Add User & Manage Roles	✓	✓	✓				
Manage all Teams & Repositories	✓	✓	✓				
Manage Subtitle Templates	✓	✓	✓				

Workspace

Users

On the Users tab can see the list of users already registered with plain X and their roles in the platform. If you have an admin or workspace manager role, you can change the role or delete a user here. Depending on the organization settings new users may get a Deactivated status when login in to the platform. To solve this a workspace manager will have to manually activate it.

To the right, you can see the list of pending invites. You can resend an invite or delete it.

The screenshot shows the 'Workspace Settings' page in plainX. The top navigation bar includes 'Users', 'Teams', 'Repositories', 'Customization', and 'Templates'. The 'Users' tab is active, displaying a list of users with columns for Name, Email, and Role. The Role column contains dropdown menus with options like 'Deactivated', 'Editor', and 'Workspace Manager'. To the right, there is a 'Pending Invites' section with a search bar and a 'Resend' button for each invite.

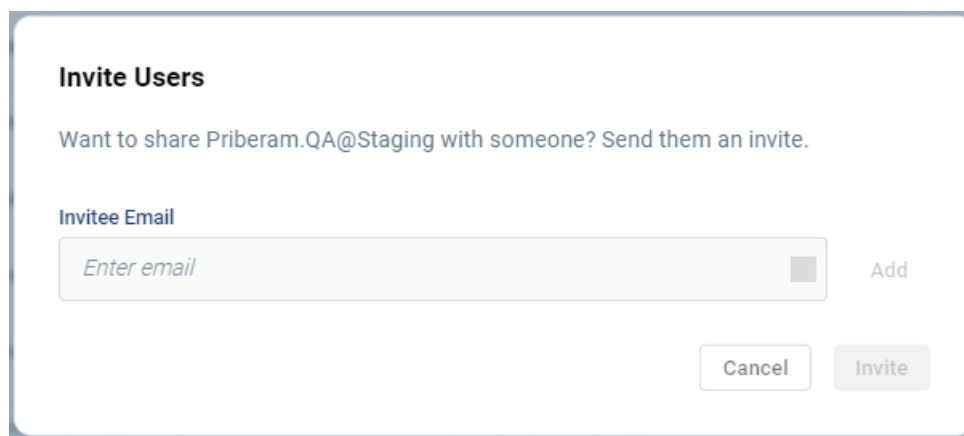
Name	Email	Role
AM		Deactivated
AM		Deactivated
AP		Editor
AA		Workspace Manager
AK		Workspace Manager
A		Workspace Manager
A		Editor
C		Workspace Manager
CL		Workspace Manager
C		Workspace Manager
CL		Workspace Manager
C		Workspace Manager
D		Workspace Manager
DR		Workspace Manager

Inviting Users

Users are invited to join workspace by workspace managers. To do that you can click the Invite Users button on the top right.

In the pop-up window, write the email address of the user you want to invite and click on **Add** . You can add multiple users at the same time, for each user you can select a role.

Once you have finished adding the email address, click on **Invite** .



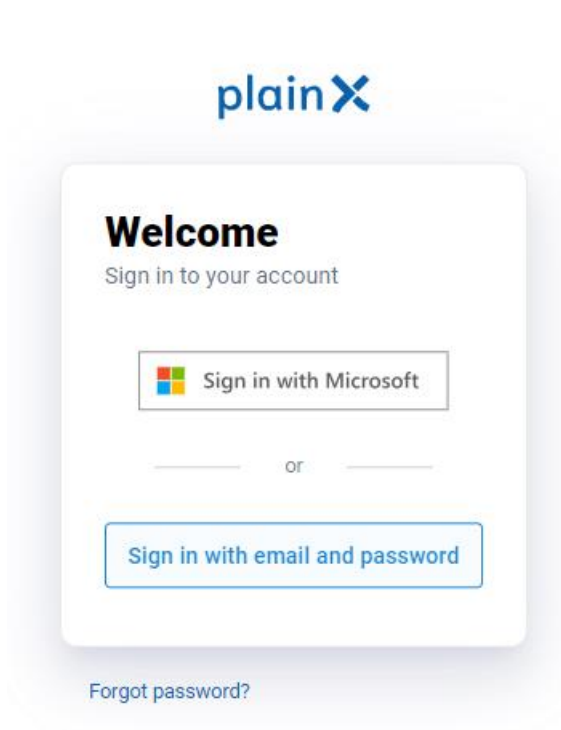
The screenshot shows a pop-up window titled "Invite Users". Below the title, it says "Want to share Priberam.QA@Staging with someone? Send them an invite." There is a text input field labeled "Invitee Email" with the placeholder text "Enter email". To the right of the input field is an "Add" button. At the bottom of the window, there are two buttons: "Cancel" and "Invite".

The user will receive an email invitation. They should follow the link in the email and the proceeding instructions to create and activate their account.

Automated Registration and Single Sign-On

plain X supports Microsoft Single Sign-On and automated registration through Microsoft Azure Active Directory. This feature requires integration with the client organization's Azure AD - please contact your workspace manager to know if you have this feature enabled. To login click the

button  .



Consumption

To access consumption page, click on Consumption tab.

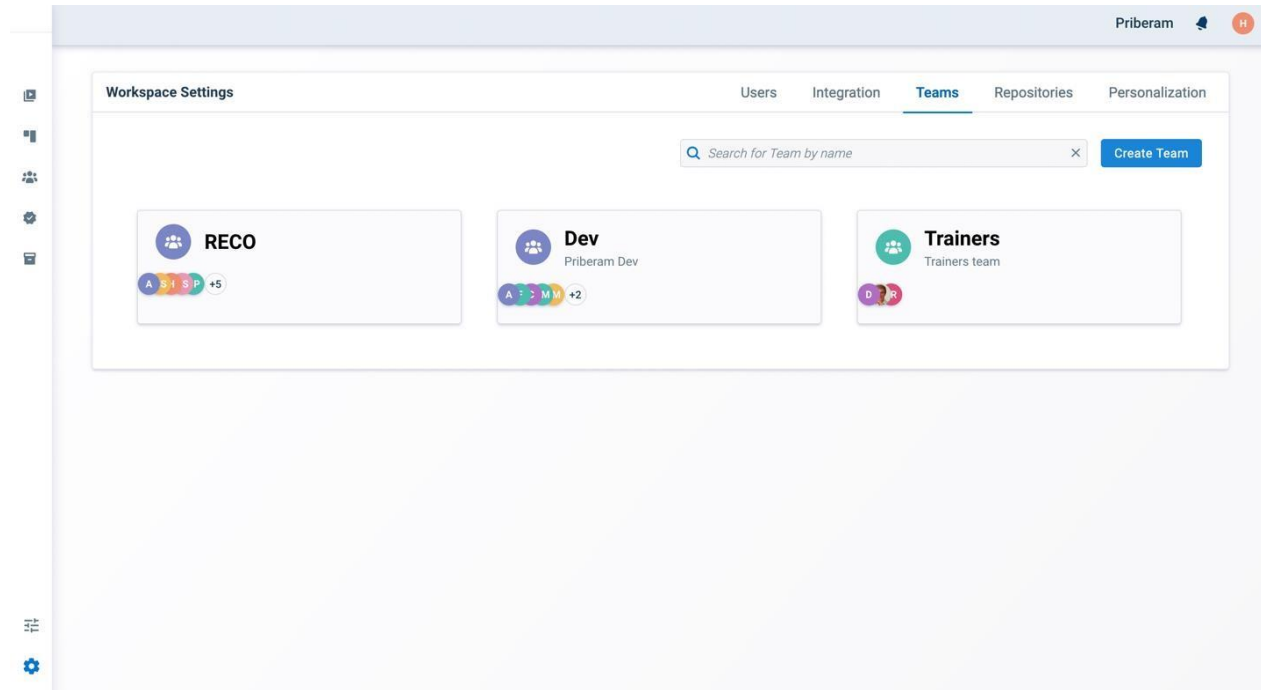
Consumption page allows you to see information about your storage, number of users and monthly transcribed minutes. You can also see statistics of using for individual users or teams.

Teams

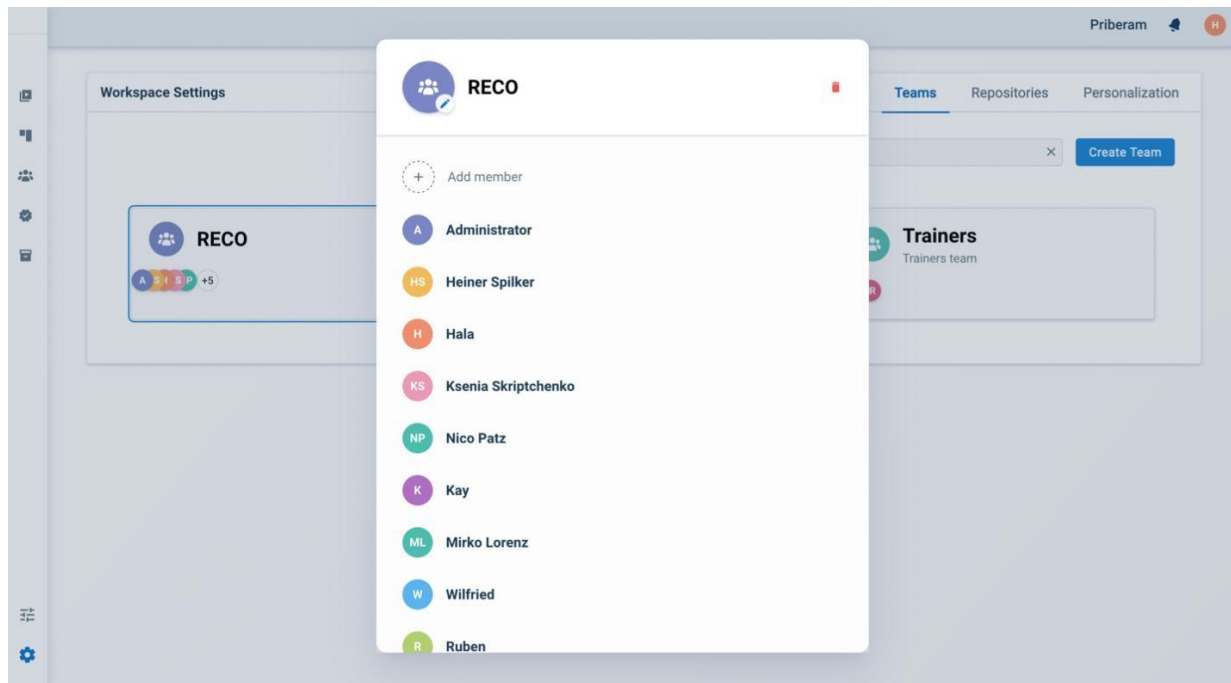
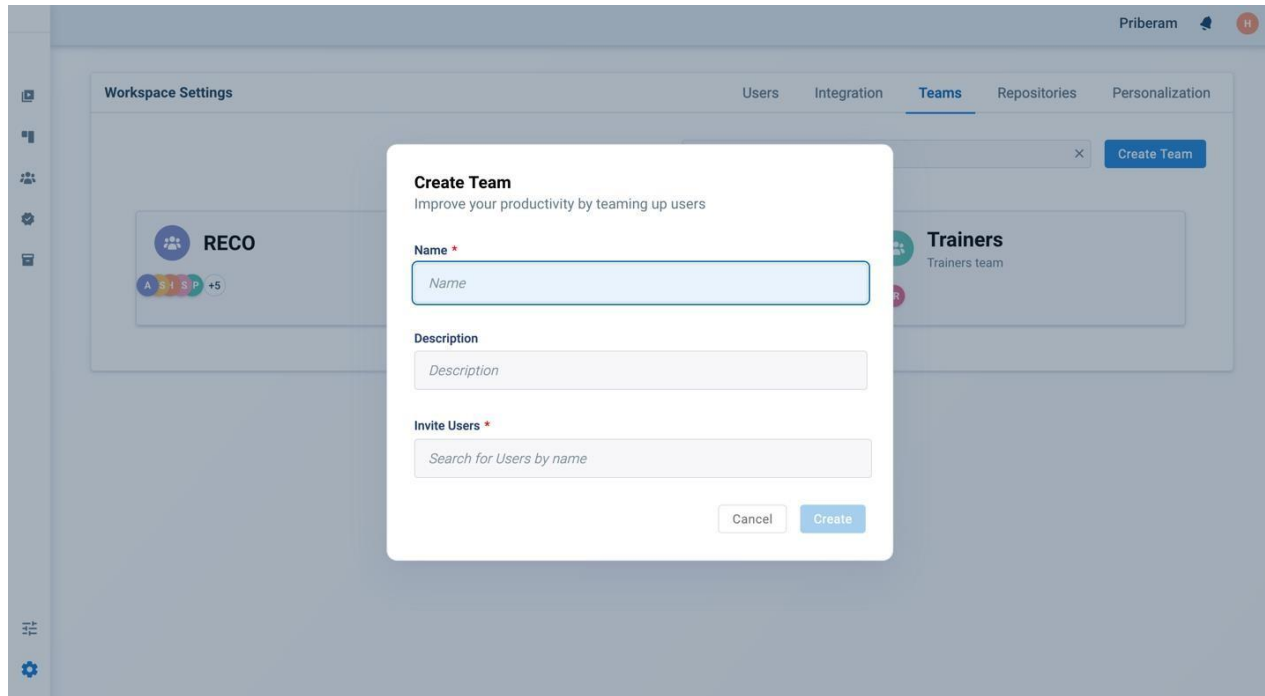
Teams tool helps to organize the work of the users who work on the tasks in the same team.

To access the Teams page, click on the Teams tab on the top right of the screen. You will see a list of current teams. You can search for a team by typing its name in the search box.

To create a new team, click on the  button at the top right of the screen.



Add a team name, and description and invite users. Click [Create](#)

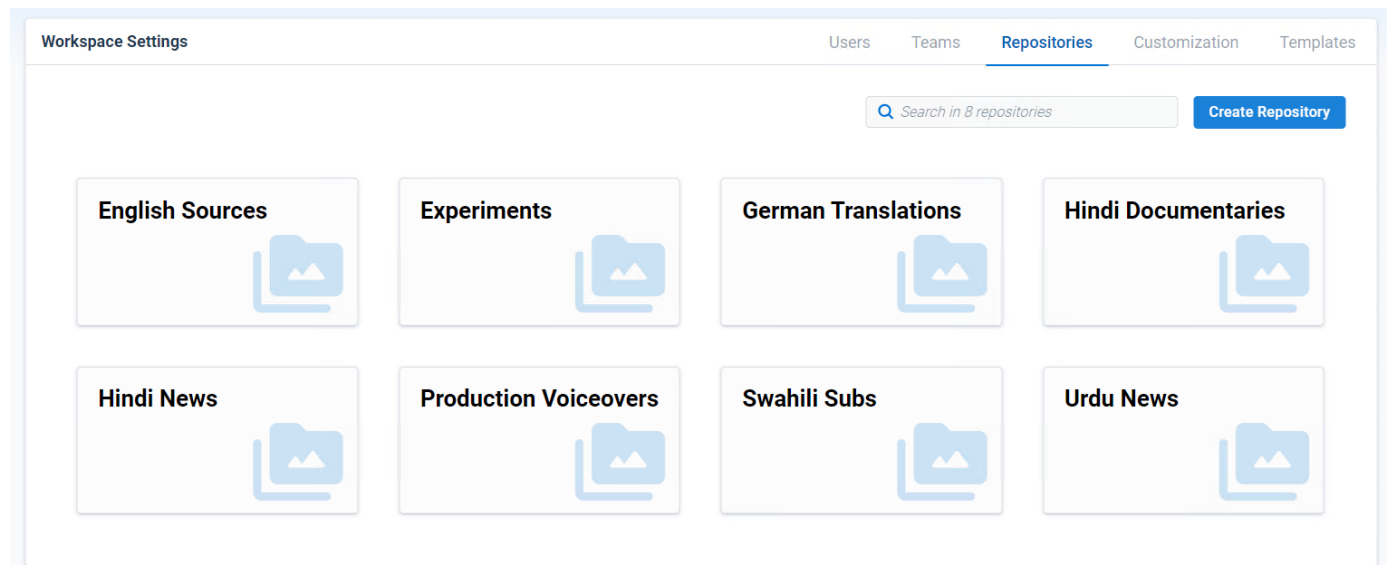


Click on any team card to see it in detail. After opening, if you have permission for it, you will be able to change the team details, add or remove users, and delete the team.

Repositories

A repository is a place where you can store the media items your team works with.

To access the repositories, click on the *Repositories* tab. There you can see all the existing repositories.



To create a new repository, click on the *Create Repository* button on the top right side.

Add the name of the repository, and a description, and select permissions in the *Collaboration Settings* selector. Select private to give access to a team or individual users or select *Workspace* to share it with all other workspace users. Click on **Create** to finalize the process.

Clicking on a repository you can see and edit which teams and users have access to it. There you can add teams by clicking *Add team* or an individual user by clicking on *Add user*. You can also see and change what rights and permissions each team and/or individual user has.

Create Repository

Create a repository to keep your media items together.

Name *


Description

Collaboration Settings

Private

Who has access to this repository?

Users

	Nika	Can edit	✕
---	------	----------	---

Cancel Create

Customization

To access this page, click on the *Customization* tab.

Customization page is accessible only to administrators and workspace managers and allows them to set the defaults for chosen language combinations for different task types. However, the defaults set by the user in **Personal** have higher priority and are respected in the first place.

To add a default engine for a specific task, click  on the right side.

Select for which language you want to set a default engine, and choose a provider, and service variant. For specific languages and engines, you can also enable speaker diarization and choose a segmentation mode for it.

Templates

Click on the Templates tab to access the page.

On the templates page, you can create personalized templates that will later can be applied to the subtitled videos. You can also edit or delete already existing templates and select the default one for each video format.

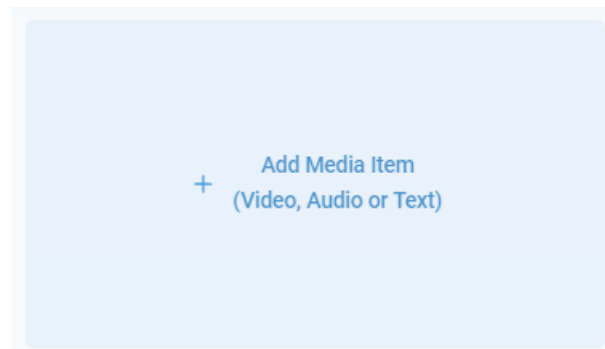
Workflows

Simple Editorial Workflow

Simple editorial workflow is the most common plain X workflow and includes the following steps: adding a media file, choosing the desired output task, editing it, and, if needed, assigning the task to be reviewed.

Add New Media

To start you will need to upload a media item to plain X. You can upload a video, audio, or text using a file on your computer, a link, or by pasting a text. For this click on the Add Media Item button and fill out all the fields in the modal. You can find the Add Media Item button on the sidenav, **Home**, or **Library page**.



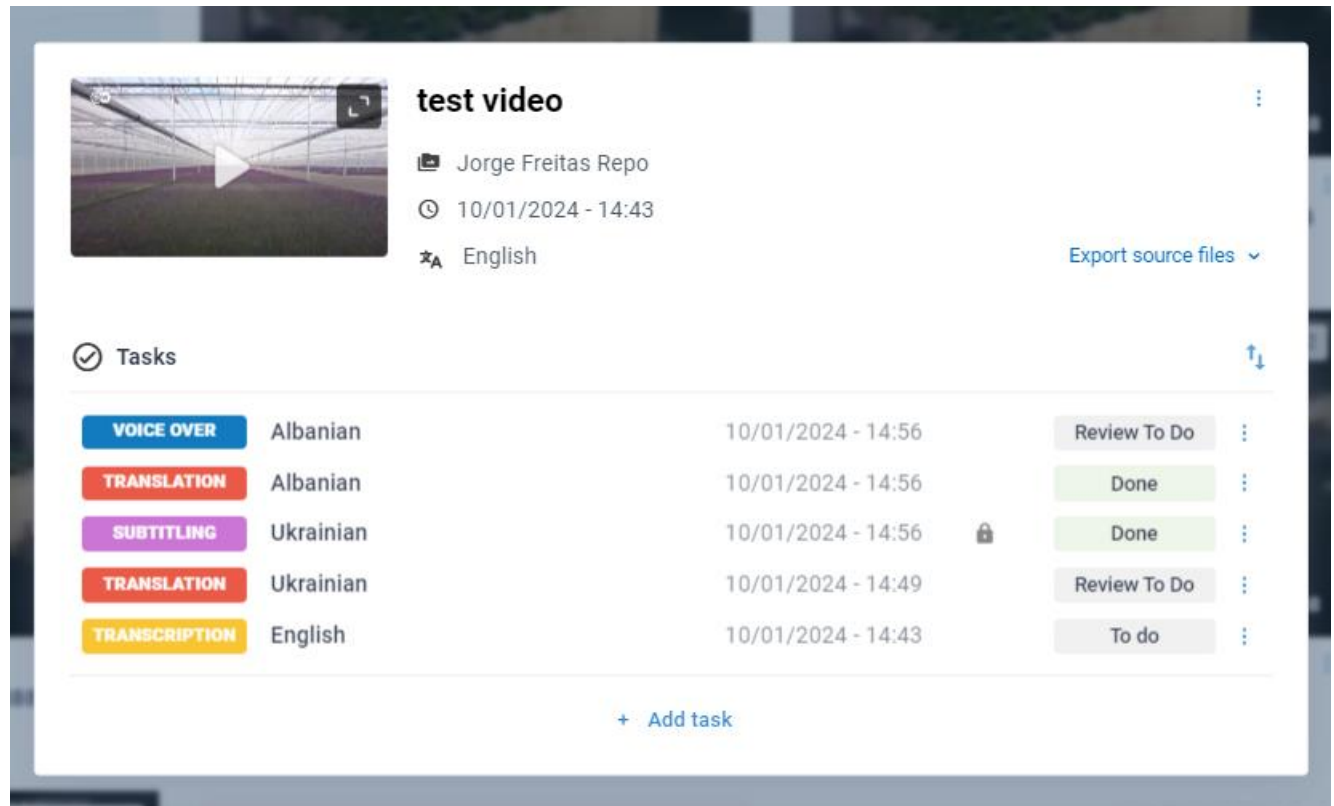
The screenshot shows a web form titled "Upload Media Item (Video, Audio or Text)". Below the title is the instruction "Choose a media item to add and setup your tasks." The form contains the following fields and options:

- File:** A text input field with the placeholder "Input a URL or select a file from your computer" and a blue upload icon button on the right.
- Title:** A text input field with the placeholder "Input a title".
- Source Language:** A dropdown menu with the text "Select Language" and a downward arrow.
- Add to:** A dropdown menu with the text "Inbox" and a downward arrow.
- What task do you want to create?:** A section with four icons and corresponding buttons: "TRANSCRIPTION" (yellow), "TRANSLATION" (red), "SUBTITLING" (purple), and "VOICE OVER" (blue).
- Buttons:** "Cancel" and "Finish" buttons at the bottom right.

Hint Silent videos or videos without a voice can be uploaded as well, with or without a transcript. This can be used to add speaker text, narrative text, or voice-over.

After clicking Finish your media item will be uploaded and processed. Since this step can take some time, depending on the size of the file, plain X allows you to continue using the platform and will notify you when the process is complete.

Hint Changing the source language is only possible if no tasks have been added to the media item yet.



test video

Jorge Freitas Repo
10/01/2024 - 14:43
English

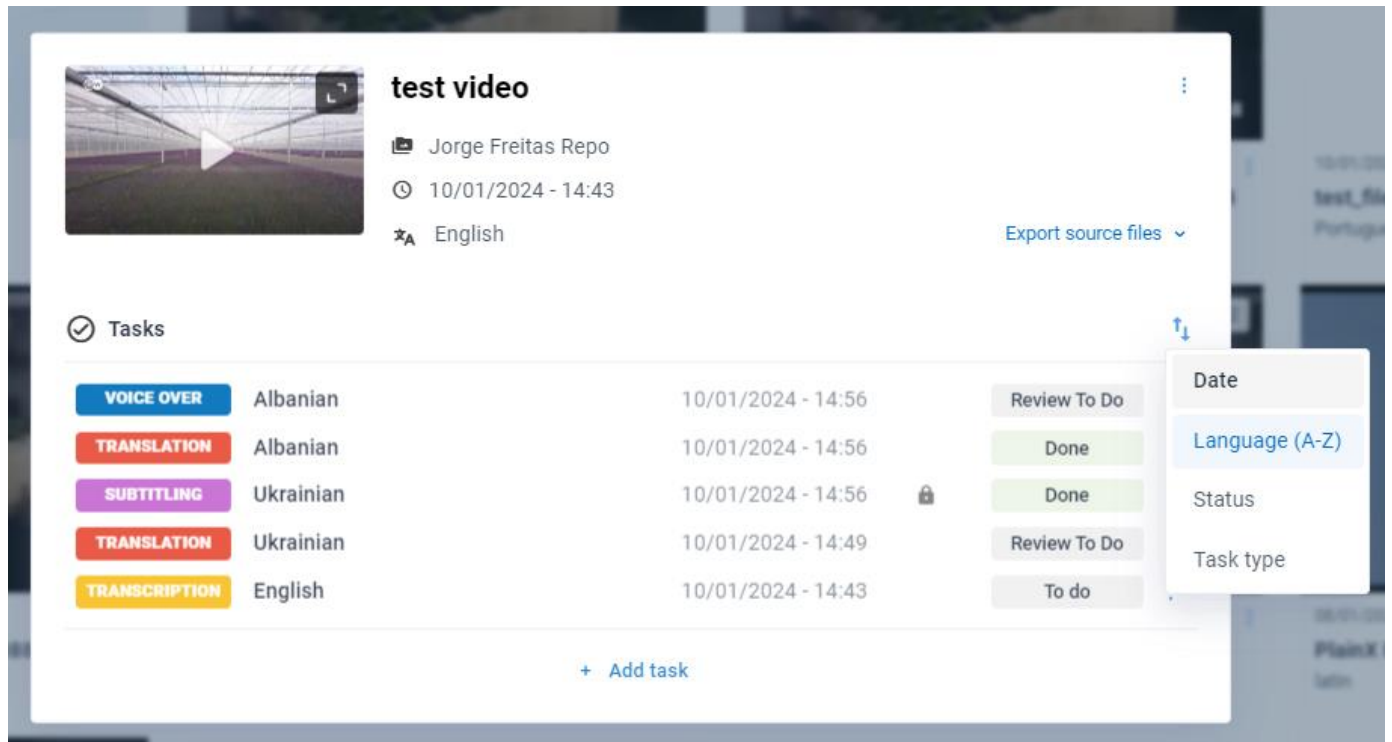
Export source files

Tasks

Task Type	Language	Date	Status
VOICE OVER	Albanian	10/01/2024 - 14:56	Review To Do
TRANSLATION	Albanian	10/01/2024 - 14:56	Done
SUBTITLING	Ukrainian	10/01/2024 - 14:56	Done
TRANSLATION	Ukrainian	10/01/2024 - 14:49	Review To Do
TRANSCRIPTION	English	10/01/2024 - 14:43	To do

+ Add task

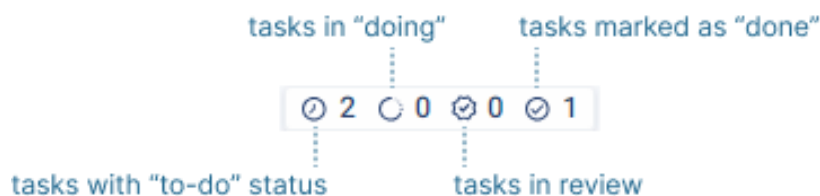
Hint If the source language is wrong after you have created tasks already, you can delete the respective tasks and run a new transcription.



From the media item card, you can also export the source files in the next available formats:

- .wav - Audio File Audio File 18/12/2023 - 02:35 [Export](#)
- .mp4 - Low Res Low Res 18/12/2023 - 02:35 [Export](#)
- .mp4 - Original File Original File 18/12/2023 - 02:35 [Export](#)

On the **Library page** at the right top corner of each media item card, you will see icons indicating if any task was associated with this item as well as the status of those tasks. Below you can find the explanation of what each of the icons indicates.



Task Creation

On the Home page select a task you want to create. Fill in the information, selecting the source and target languages for the task, its name, and repository.

There are the following task types available to create in plain X:


- Transcription
- Translation
- Subtitling
- Voiceover

There are two more ways to create a task on plain X:

1. From the Library

Select the item you want to add a task to and click the *Add task* button.


2. From Personal Boards

Click icon  to add a task. In the open modal select a task type.

What task do you want to create?

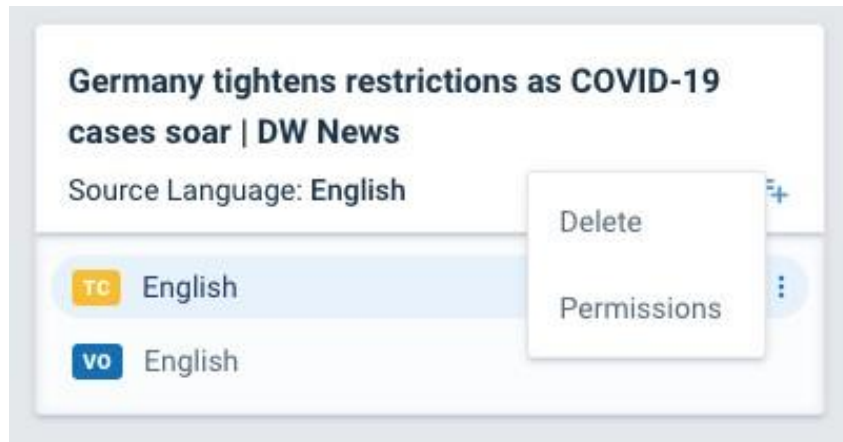


Note: You can create any task at this point. However, some tasks might have dependencies. For example, if you wish to select a Subtitling task with the target language Spanish for a video item with the source language German, a Transcription in German and a Translation to Spanish will be created as well (if those haven't existed yet).

After selecting a task type click 

On the Personal board on the right from the task type and the target task language, you can see three dots that correspond to more options icons. By clicking on those dots, **depending on your role**, you may be able to:

- delete tasks
- edit permissions: this will allow you to add other persons with editing or reviewer
- rights to your personal tasks.



Best Practice: Creating Tasks

We recommend using initially the standard, goal-oriented workflow, in which each the task is created and executed separately, in sequence.

Below, we will describe the two most commonly used sequences of steps to create a task.

Standard, recommended procedure:

This procedure enables working only in the end-goal step (e.g. subtitling in the source language or target language) and disregarding the rest. Do this only if you are sure you do not need the other steps as an output.

1. Create a combined task for the end goal, for example subtitling in a target language
2. Go immediately into the end-goal task, e.g. subtitling in Spanish - edit and mark as done a dependent (previous) steps, such as transcription and translation, are left untouched.

Collaborative translation task procedure:


1. Create transcription - edit - mark as done => This can be used as a basis for all subsequent tasks.
2. Create source language subtitling if needed - Edit - Mark as Done
3. Per target language:
 1. Create a combined task (e.g., Translation and Subtitling in one step => select Subtitling immediately)
 2. Edit the translation as needed - Mark as Done
 3. Go into the Subtitling file - note & accept the warning - follow the suggestions to work from the edited translation by selecting Upgrade.
4. Create a Voice-Over task if needed - Edit - Mark as Done.

Sequential, simplest procedure:


1. Create transcription task - Edit - Mark as Done
2. Create source language subtitling task if needed - Edit - Mark as Done
3. Create target language translation task - Edit - Mark as Done
4. Create target language subtitling task - Edit - Mark as Done

5. Create target language voice-over task if needed - Edit - Mark as Done

Steps 3, 4, and 5 can be done in parallel for other target languages. Once you are familiar with this workflow, you can use alternatives that suit your work better:

For Translation, Subtitling, or Voice over tasks select the target language. You can also choose Service Provider and Service Variant (where applicable) in the Advanced Configuration section which can be accessed by clicking the 

In many cases, the Service Provider and Service Variant fields are already filled in with a recommended service for the requested language or language pair. Those pairs are set at the **Customization** page we described earlier. If you prefer to use your pair, you can always change it at this step or on the **Personal** page.

For Translation, Subtitling, and Voice-over tasks there is a possibility to select dependencies manually. To do so click on Dependencies selector and select Manually.  button will appear. Click it to proceed to the next step.

For the **Transcription** task, it is also possible to add a manuscript at this step. A manuscript is a text file that aligns the plain X-generated transcription to the transcription in your text file. You can upload the manuscript while creating a task, or after, accessing Service menu on the transcription task page.


If you want to upload a manuscript, you can drag and drop or upload a txt file from your computer. **The file needs to be in the format txt UTF-8** and contains only transcribed/subtitled text to be processed properly.

Text file

Use a text file with the transcript. It must be a text file using UTF-8 encoding.



You can also add the *Editors* and *Reviewers* at this stage. The review process is covered in more detail later in this guide.

Click on  to finish the process and create the task.

Once the task is created, it will appear on the **Personal Board** and the **Shared Board**.

Both boards have similar structures and are divided into *To Do*, *Doing*, *Being Reviewed*, and *Done* columns. *Being Reviewed* column is collapsable and to expand it you need to click on it.

Newly created tasks, and tasks that are being processed, appear in the *To do* column. The progress of each automated task is indicated by the circular progress bar next to it. Once the item is ready, the progress bar disappears. You can then click on the item to open it and start editing.

- Once you open a task, it is moved into the *Doing* column.
- Once a task has been finished and in review, it is moved to the *Being Reviewed* column.
- Once the item (with or without review) is marked as completed, the task appears in the *Done* column.

The screenshot displays the plainX interface with a Kanban board. The top navigation bar includes the plainX logo, the user name 'Priberam', and a notification icon. Below the navigation bar, there are dropdown menus for 'Sort By Latest' and 'Group By Medialtem'. The left sidebar is divided into 'EXPLORE' and 'RECENT' sections. The 'EXPLORE' section lists 'Library', 'Personal Board', 'Shared Board', 'Review Board', and 'All Done Tasks'. The 'RECENT' section lists 'Test_engines_HA', '90 secs news_test VO_p...', 'Test_URDU_RTL', and '90 secs news_2021-04-...'. The 'SETTINGS' section at the bottom left includes 'Personal' and 'Workspace'. The main workspace contains a Kanban board with four columns: 'Todo', 'Doing', 'Being Reviewed', and 'Done'. The 'Todo' column has three cards: 'Darum ist Spanien bei COVID-19 Impfungen so erfolgreich? HA' (Source: German) with sub-items for French and German; 'Test_engines_HA' (Source: English) with sub-items for English and French; and 'Test_URDU_RTL' (Source: German) with sub-items for German and Urdu. The 'Doing' column has one card 'Test_URDU_RTL' (Source: German) with a sub-item for Arabic. The 'Being Reviewed' and 'Done' columns are empty.

For every type of task, the editing workflow stays the same and usually includes the steps described below:

To review and correct the text:

1. Check that the video and transcript are synchronized.

The platform will automatically create time codes and synchronize them with the text. However, you may need to check them. You can do this by pressing the play ► button in the player (or using Space to play and stop the video). A red marker will appear on the transcript and will indicate where in the video the text is situated.

2. Correct the synchronization by changing the position of the timecodes on the left of the script. You can drag the blue marker to move the timecode earlier or later. Drag the top or bottom line of the segment to extend or shorten the segment duration.
3. You can also change the numbers in the timecode. The format is hh:mm:ss: frame. You may need to create some space to enlarge another segment, do this by moving the timing of a neighboring segment up or down.
4. You can scroll up or down. If you click on any segment, the video jumps directly to that position.
5. Check the text for errors. You can edit the text directly in the corresponding box by clicking on it. Pay special attention to named entities and punctuation.
6. To merge 2 segments, you can press on the symbol appearing between two segments
7. To split 2 segments, press Control + Enter after the word at which you want to split the segment

8. To delete a segment, you can press on the delete icon at the bottom right of a segment
9. To add a comment, press the comment icon on the bottom right corner of the text segment
10. To add an editorial note click the note icon. Editorial notes are the “hints” left by the editor for reviewers. Make sure editorial notes are visible are checked in the View menu.

Speaker diarization

Some transcription services in plain X support speaker diarization. When creating a transcription task, you can now select an expected number of speakers appearing in a file. You can also choose a segmentation mode from two available options: Speaker & Grammar, or just Grammar. You can manually change speaker names after the task is created.

Versioning

This function is only available to workspace managers.

For each task you are working on, you can save a version of it at any time. This is a snapshot of the task at a given time and allows you to revert to a previous version if you need it.

To save a version of a task, select the top left *File - Save*.

Save As

Save current version of the task

Name

Version 6

Summary

Input a Summary

Cancel

Save As

To work in a previous version, open that version by selecting *File - Open*.

The screenshot shows the plainX application interface. On the left, there is a sidebar with 'EXPLORE' (Library, Personal Board, Shared Board, Review Board) and 'RECENT' (test transcription_1, test file, etc.) sections. The main area displays a video player with a version history dropdown menu. The dropdown menu lists the following versions:

- 2023-12-21 10:07 (Nika - 21/12/2023 at 03:07)
- 2023-12-21 10:02 (Nika - 21/12/2023 at 03:02)
- 2023-12-21 10:00 (Nika - 21/12/2023 at 03:00)
- Version 3 (Google transcription, Nika - 21/12/2023 at 03:02)
- 2023-12-21 09:49 (Nika - 21/12/2023 at 02:49)

The video player shows a man speaking, and the right sidebar displays the transcription text in Ukrainian. The transcription text includes:

- an editorial note looks like this
- Бажаю здоров'я, шановні українці і українки! Зміни!
- Сьогодні є важливе кадрове рішення. За поданням міністра оборони змінив Командувача Медичних Сил Збройних Сил України.
- Замість генерал-майора медичної служби Тетяни Остащенко новим командувачем призначив генерал-майора медичної служби Анатолія Казмірчука, начальника Національного військово-клінічного центру головний військовий клінічний госпіталь у місті Києві.
- Завдання очевидне. І про це неодноразово говорилося в суспільстві, а зокрема в спільноті наших бойових медиків.
- Потрібен принципово новий рівень медичного забезпечення наших воїнів.
- Від якісних турнікетів до повної цифровізації та прозорості постачання, від якісного тренування до ширшої комунікації з бойовими медиками у тих підрозділах, де медицина розгорнута дійсно правильно та дійсно ефективно.
- Досвід ефективності конкретних підрозділів має бути поширений на всі сили оборони.

The saved version appears at the bottom right of the screen and it is saved as default with the date and time (UTC/GMT) it was created.

When opening the saved version if the user wants to edit it, the platform will ask the user to mark the opened version as current.

Text File Processing

You can also transcribe, translate, and get a synthetic voice to read the text/translation of text files by using a workaround.

Upload any short video as explained above and make sure you select the source language of your targeted text as the target language and upload the text file you want to get processed as a manuscript to the video.

Now process the video using the normal procedure.

Export Files

There are two ways to export a file from plain X.

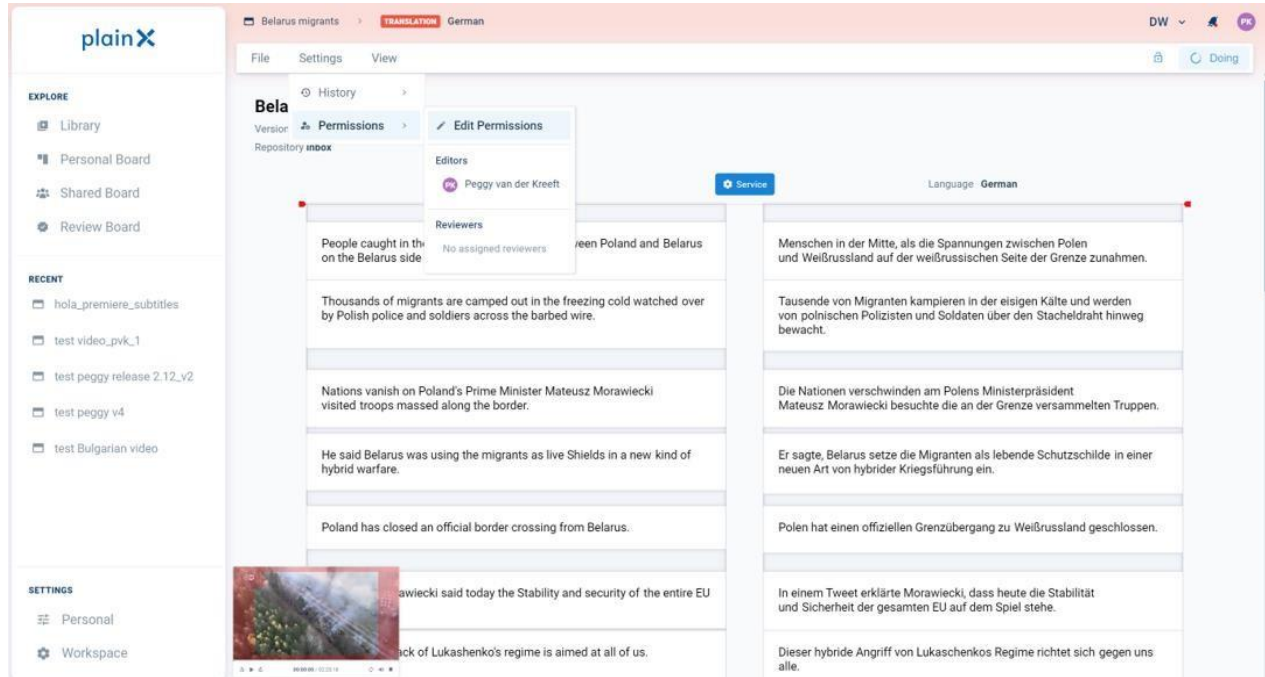
After tasks have been set as *Done* the output becomes available as exportable files. You can directly export a file from the task page or the Media Item Modal. At the Media Item Modal, you also have an option to export the source file.

Collaborative Editorial Workflow

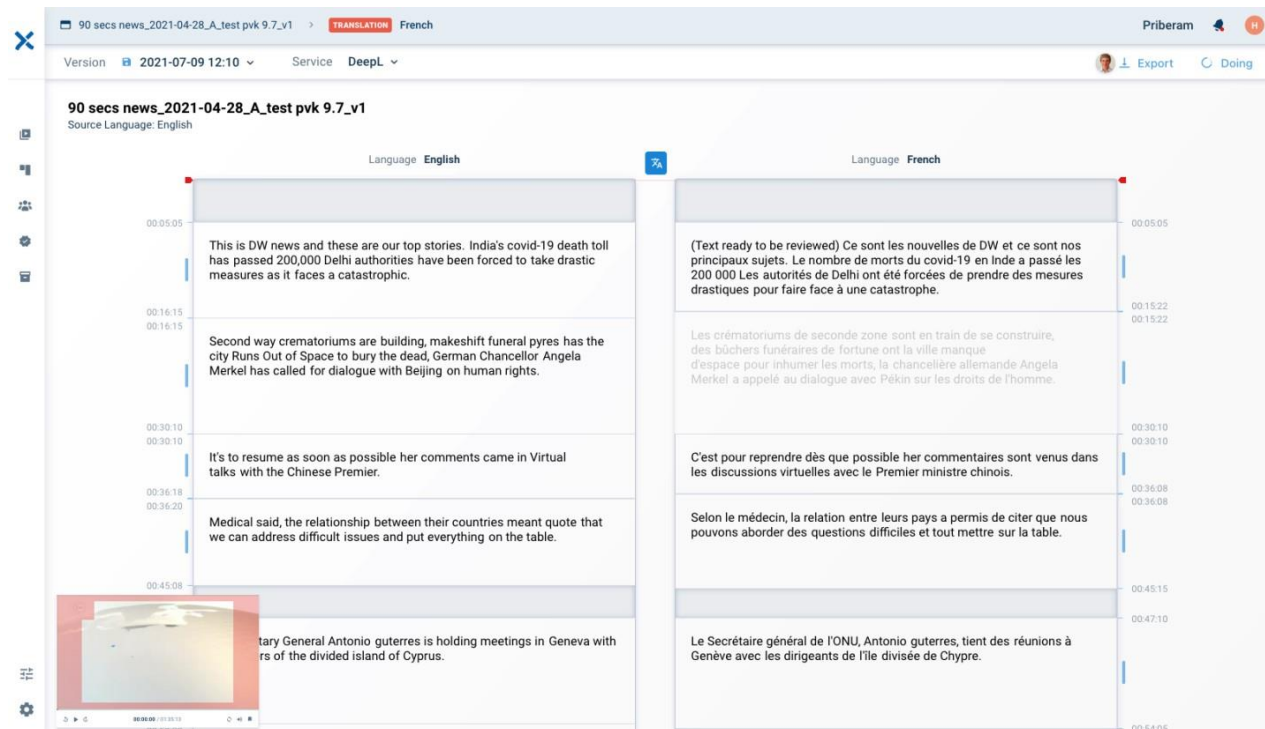
The plain X platform allows collaborative work between team members. This means that two or more users can access the same task at the same time. Collaboration can only happen between users in a shared team and repository.

Collaborative work should be done per task in the same document.

To allow collaborative work, add the collaborator as an editor to a task. If a user is in a task, it is visible in the board's view. A circular user icon will appear next to the task name. Inside a task, the user icon will show at the top right of the window.



When another user is editing a segment, this segment will appear greyed out for you. You will not be able to edit it until the other user stops being active in the segment.



Assigning Tasks and Reviews

The review workflow allows you to assign tasks to other users or teams of users as well as select a reviewer to review the task once it is done.

You can assign people or teams to tasks/reviews at several stages:

- when creating the task
- while editing, by selecting *Settings – Permissions*
- when passing a task for review

Editor's View

Once a task is created, editors assigned to the task can start working on it. The task you are assigned should appear in your **Personal board**.

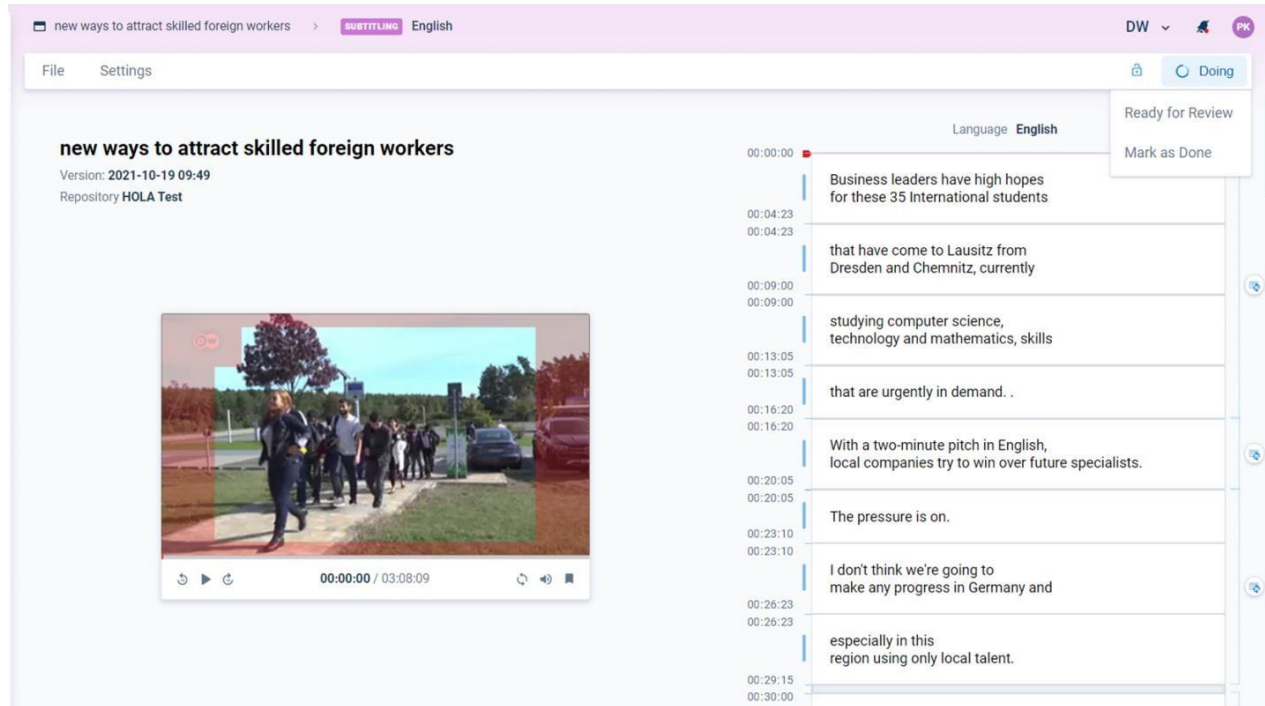
Once the task has been completed, you can:

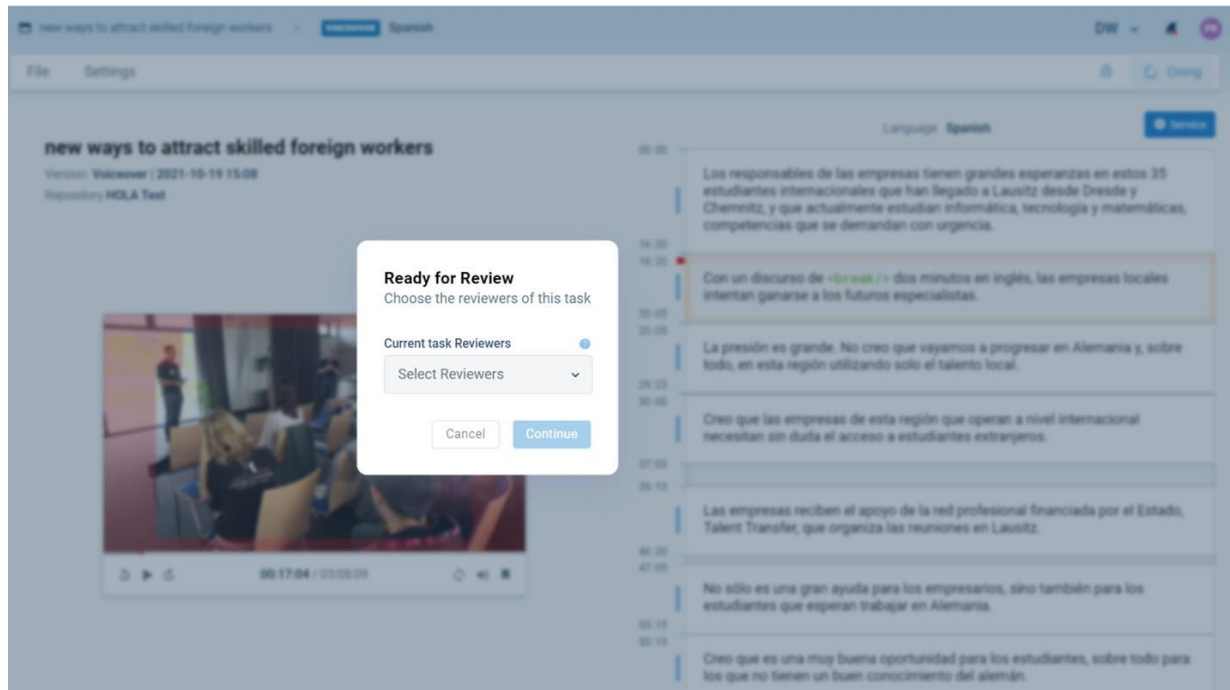
- Mark the task as *Done*
- Mark as *Doing*. This will send the task back to the board under *Doing*
- Mark as *Ready for a Review*. Clicking that button will open the window where you can choose a Reviewer

Once you have sent the task for review, you no longer have access to the item as an editor.

It will appear on your **Personal Board** under *Being Reviewed*. Click on the vertical tab *Being Reviewed* to open that view.

You can mark the task as done immediately and skip the review if it's not needed. You will get a warning when skipping the review.



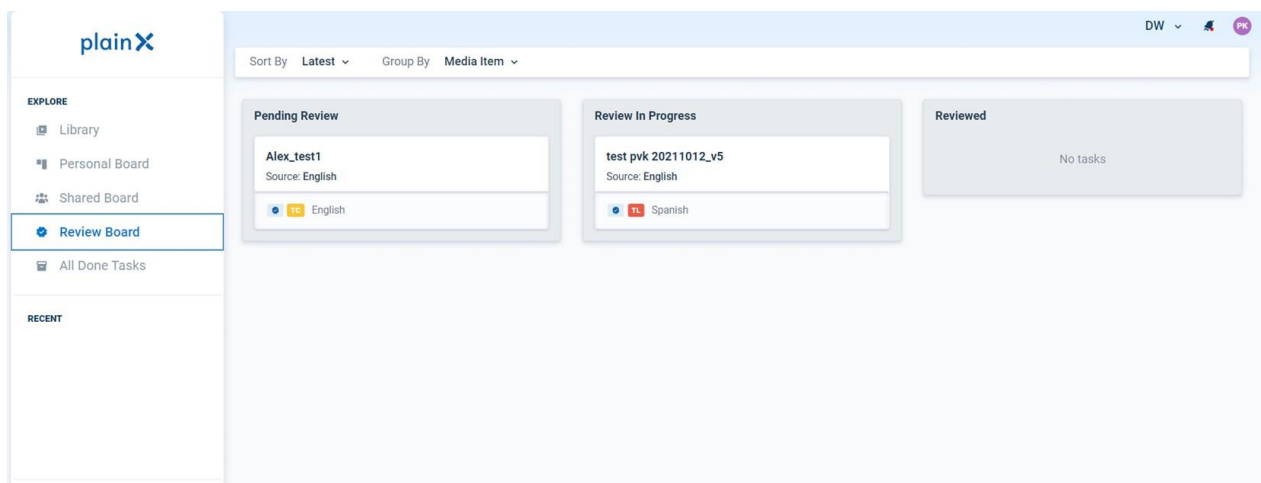


The screenshot shows a video player interface for a video titled "new ways to attract skilled foreign workers". The video is in Spanish and has a duration of 03:08:09. A dialog box titled "Ready for Review" is overlaid on the video, asking the user to "Choose the reviewers of this task". The dialog box includes a "Current task Reviewers" section with a "Select Reviewers" dropdown menu, a "Cancel" button, and a "Continue" button. The video player shows a scene of people in a meeting room. The video content on the right side of the player includes subtitles in Spanish, such as "Los responsables de las empresas tienen grandes esperanzas en estos 35 estudiantes internacionales que han llegado a Lausitz desde Dresde y Chemnitz, y que actualmente estudian informática, tecnología y matemáticas, competencias que se demandan con urgencia."

Reviewer's View

Once a task has been assigned to a reviewer, it will appear on the Review Board for the reviewer.

It will also show in the Being Reviewed section in the **Shared Board** for the team members to keep track of the status of all tasks within the team. Click on the vertical tab **Being Reviewed** to expand the column.



In the *Review* view, the reviewer can edit the text. Changes are immediately applied to the text and edits are not directly traceable, although changed segments are highlighted in blue for the editor.

In addition, comments (one of the options bottom right of each segment) can be added to alert the editor to some of the changes.

Once the review is done, the reviewer can click on the *Review Doing* button on the top right of the view. The reviewer can then:

- Mark the review as *Done*
- or
- send the review back to the editor

If the task is re-assigned to the editor, then the editor can make the appropriate changes and set the task as *Done* or send it back for another review.

Annex 2 – plain X Automations API

[SELMA] plain X Automations API v1.0

1. Introduction

The plain X automations API enables the creation of transcription, translation, subtitling and voiceover jobs and the corresponding export of the final results. These actions can be performed in a step-by-step fashion through the `Tasks` and `Exports` routes or they can be performed in a synchronous unified call through the `workflows` route.

Unless there's a reason to do so, it's recommended to use the `workflows` routes.

There's an interactive Swagger documentation here: <https://app.plainx.com/automations/swagger/index.html>

2. Synchronous Operation - `workflows` (Recommended)

The `workflows` route allows performing a processing task in an end-to-end synchronous fashion and get the result immediately as the result of the HTTP call.

Note: Some of the fields in the workflow configuration are optional and not specified in the examples below (e.g., `itemTitle`, `fileName`, etc) - Please consult the Swagger documentation to see the complete API models.

2.1 Transcription Workflow Example

- Description: Upload an mp4 video, request a transcription and download the resulting text file.

```
1 curl --max-time 14400 --request POST \  
2   --url https://app.plainx.com/automations/api/v1.0/workflows \  
3   --header 'Authorization: Bearer <ApiKey>' \  
4   --header 'Content-Type: multipart/form-data' \  
5   --form 'workflowConfiguration={"Upload": {"SourceLanguage":"en", "RepositoryId": "<repositoryId>"},"Tasks": {"W  
6   --form 'file=@<filePath>' > out.txt
```

2.2 Translation Workflow Example

- Description: Upload an mp4 video, request a transcription and a translation and download the resulting text file.

```
1 curl --max-time 14400 --request POST \  
2   --url https://app.plainx.com/automations/api/v1.0/workflows \  
3   --header 'Authorization: Bearer <ApiKey>' \  
4   --header 'Content-Type: multipart/form-data' \  
5   --form 'workflowConfiguration={"Upload": {"SourceLanguage":"en", "RepositoryId": "<repositoryId>"},"Tasks": {"W  
6   --form 'file=@<filePath>' > out.txt
```

2.3 Translation Workflow with Text Input Example

- Description: Upload an text, request a translation and download the resulting text file.

```
1 curl --max-time 14400 --request POST \  
2   --url https://app.plainx.com/automations/api/v1.0/workflows \  
3   --header 'Authorization: Bearer <ApiKey>' \  
4   --header 'Content-Type: multipart/form-data' \  
5   --form 'workflowConfiguration={"Upload": {"SourceLanguage":"en", "RepositoryId": "<repositoryId>"},"Tasks": {"W  
6   --form 'text=This is an example text.' > out.txt
```

2.4 Subtitling SRT Export Workflow Example

- Description: Upload an mp4 video, request a transcription, a translation and a subtitling, download the resulting srt file.

```
1 curl --max-time 14400 --request POST \  
2 --url https://app.plainx.com/automations/api/v1.0/workflows \  
3 --header 'Authorization: Bearer <ApiKey>' \  
4 --header 'Content-Type: multipart/form-data' \  
5 --form 'workflowConfiguration={"Upload": {"SourceLanguage":"en", "RepositoryId": "<repositoryId>"},"Tasks": {"W  
6 --form 'file=@<filePath>' > out.srt
```

2.5 Subtitling Video Export Workflow Example

- Description: Upload an mp4 video, request a transcription and a subtitling, download the mp4 file with the video with subtitles.

```
1 curl --max-time 14400 --request POST \  
2 --url https://app.plainx.com/automations/api/v1.0/workflows \  
3 --header 'Authorization: Bearer <ApiKey>' \  
4 --header 'Content-Type: multipart/form-data' \  
5 --form 'workflowConfiguration={"Upload": {"SourceLanguage":"en", "RepositoryId": "<repositoryId>"},"Tasks": {"W  
6 --form 'file=@<filePath>' > out.mp4
```

2.6 Voiceover Audio Export Workflow Example

- Description: Upload an mp4 video, request a transcription and a voiceover, download the wav file with the voiceover audio.

```
1 curl --max-time 14400 --request POST \  
2 --url https://app.plainx.com/automations/api/v1.0/workflows \  
3 --header 'Authorization: Bearer <ApiKey>' \  
4 --header 'Content-Type: multipart/form-data' \  
5 --form 'workflowConfiguration={"Upload": {"SourceLanguage":"en", "RepositoryId": "<repositoryId>"},"Tasks": {"W  
6 --form 'file=@<filePath>' > out.wav
```

2.7 Voiceover Video Export Workflow Example

- Description: Upload an mp4 video, request a transcription and a voiceover, download the mp4 file with the voiceover video.

```
1 curl --max-time 14400 --request POST \  
2 --url https://app.plainx.com/automations/api/v1.0/workflows \  
3 --header 'Authorization: Bearer <ApiKey>' \  
4 --header 'Content-Type: multipart/form-data' \  
5 --form 'workflowConfiguration={"Upload": {"SourceLanguage":"en", "RepositoryId": "<repositoryId>"},"Tasks": {"W  
6 --form 'file=@<filePath>' > out.mp4
```

3. Step-by-Step Asynchronous Operation - `Tasks` and `Exports`

The `Tasks` routes allow querying for available services, upload media items, create plain X tasks and check the results of queued tasks.

The `Exports` routes allow creating an export task, check the status and finally get a URL to fetch the exported resource. These routes offer more fine grained control over the process than the `Workflows` routes - For example, through `Tasks` you can add tasks to existing media items instead of creating a new media item for each tasks.

Note: Since these routes are more intricate to use, the examples are provided in python.

3.1 Transcription Task and Export Example

- Description: Upload an mp4 video, request a transcription and download the text file in the format of segmented blocks with timecodes.
- Python Script

```
1 import requests
2 import time
3 import json
4 import sys
5
6 # Example
7 # python .\transcription.py .\20s_test_video.mp4
8 # Need to have credentials.json file with:
9
10 #{
11 #   "apiKey": "",
12 #   "repositoryId": "",
13 #   "workspaceUserId": ""
14 #   "baseUrl": ""
15 #}
16 #
17 # NOTE: apiKey should be already the base64 version of the json with {"OrchestrationIdentityId":"","Orchestrati
18
19 video_path = sys.argv[1]
20 taskName = "Transcription"
21
22 with open('credentials.json', 'r') as json_file:
23     data = json.load(json_file)
24     apiKey = data['apiKey']
25     repositoryId = data['repositoryId']
26     workspaceUserId = data['workspaceUserId']
27     BASE_URL = data['baseUrl']
28
29 headers = {
30     "Authorization": "Bearer " + apiKey
31 }
32
33 # Step 1: Create tasks via POST /api/v1/Tasks
34 task_data = {
35     "upload": {
36         "itemTitle": "Test Title",
37         "fileType": "Video",
38         "itemDescription": "Test Description",
39         "sourceLanguage": "en",
40         "filename": "file.mp4",
41         "repositoryId": repositoryId,
42         "sourceUrl": None,
43         "text": None,
44     },
45     "tasks": {
46         "workspaceUserIds": [workspaceUserId],
47         "taskName": {
48             "serviceProvider": "Azure",
49             "serviceVariant": "US"
50         }
51     }
52 }
53
54 form_data = {
```

```

55     "taskWorkflowConfiguration": json.dumps(task_data)
56 }
57
58 sent_files = {
59     'file': ('file.mp4', open(video_path, 'rb'), 'application/octet-stream')
60 }
61
62 response = requests.post(f"{BASE_URL}/api/v1/Tasks",
63                          data=form_data, headers=headers, files=sent_files)
64 try:
65     response_data = response.json()
66     media_item_id = response_data["mediaItemId"]
67 except:
68     print("/api/v1/Tasks: Error reading response")
69     print(response)
70     print(response.text)
71     exit(1)
72
73 print("media_item_id: " + media_item_id)
74
75 # Step 2: Check tasks status in a loop
76 while True:
77     response = requests.get(f"{BASE_URL}/api/v1/Tasks",
78                             params={"mediaItemId": media_item_id}, headers=headers)
79     task_info = response.json()
80     if task_info["status"] == "Done":
81         break
82     print ("waiting for Task to be Done...")
83     time.sleep(5)
84
85 # Step 3: Create an export via GET /api/v1/Exports/{resourceType}
86 for task in task_info["tasks"]:
87     if task["type"] == taskName:
88         taskId = task["id"]
89
90 export_data = {
91     "taskId": taskId,
92     "resourceType": "TimedSegmentBlock" # or "Txt"
93 }
94
95 response = requests.post(f"{BASE_URL}/api/v1/Exports/{export_data['resourceType']}", params={
96     "taskId": export_data["taskId"]}, headers=headers)
97 export_response_data = response.json()
98 workflow_id = export_response_data["workflowId"]
99
100 # Step 4: Check export status in a loop
101 while True:
102     response = requests.get(f"{BASE_URL}/api/v1/Exports/status",
103                             params={"workflowId": workflow_id}, headers=headers)
104     if (response.status_code != 202):
105         try:
106             export_status = response.json()["status"]
107             media_resource_id = response.json()["mediaResourceId"]
108         except:
109             print("/api/v1/Exports/status: Error reading response")
110             print(response)
111             print(response.text)
112             exit(1)

```



```

113
114     if export_status == "Ready":
115         break
116     print ("waiting for Export to be Done...")
117     time.sleep(5)
118
119 # Step 5: Get export result via GET /api/v1/Exports/download
120 response = requests.get(f"{BASE_URL}/api/v1/Exports/download",
121                         params={"mediaResourceId": media_resource_id}, headers=headers)
122
123 try:
124     export_result = response.json()
125 except:
126     print("/api/v1/Exports/download: Error reading response")
127     print(response)
128     print(response.text)
129     exit(1)
130
131
132 output_path = "output.seg.txt"
133 print ("writing output to " + output_path + " ...")
134
135 with open(output_path, "w", encoding='utf-8') as json_file:
136     json_file.write(export_result["blob"])
137

```

3.2 Translation Task and Export Example

- Description: Upload an mp4 video, request a transcription and a translation, download the text file in the format of segmented blocks with timecodes.
- Python Script:

```

1  import requests
2  import time
3  import json
4  import sys
5
6  # Example
7  # python .\translation.py .\20s_test_video.mp4
8  # Need to have credentials.json file with:
9
10 # {
11 #   "apiKey": "",
12 #   "repositoryId": "",
13 #   "workspaceUserId": ""
14 #   "baseUrl": ""
15 # }
16 #
17 # NOTE: apiKey should be already the base64 version of the json with {"OrchestrationIdentityId":"","Orchestrati
18
19 video_path = sys.argv[1]
20 taskName = "Translation"
21
22 with open('credentials.json', 'r') as json_file:
23     data = json.load(json_file)
24     apiKey = data['apiKey']
25     repositoryId = data['repositoryId']

```

```

26 workspaceUserId = data['workspaceUserId']
27 BASE_URL = data['baseUrl']
28
29 headers = {
30     "Authorization": "Bearer " + apiKey
31 }
32
33 # Step 1: Create tasks via POST /api/v1/Tasks
34 task_data = {
35     "upload": {
36         "itemTitle": "Test Title",
37         "fileType": "Video",
38         "itemDescription": "Test Description",
39         "sourceLanguage": "en",
40         "filename": "file.mp4",
41         "repositoryId": repositoryId,
42         "sourceUrl": None,
43         "text": None,
44     },
45     "tasks": {
46         "workspaceUserIds": [workspaceUserId],
47         taskName: {
48             "ServiceProvider": "Azure",
49             "SourceLanguage": "en",
50             "TargetLanguage": "pt-PT"
51         }
52     }
53 }
54
55 form_data = {
56     "taskWorkflowConfiguration": json.dumps(task_data)
57 }
58
59 sent_files = {
60     'file': ('file.mp4', open(video_path, 'rb'), 'application/octet-stream')
61 }
62
63 response = requests.post(f"{BASE_URL}/api/v1/Tasks",
64                          data=form_data, headers=headers, files=sent_files)
65 try:
66     response_data = response.json()
67     media_item_id = response_data["mediaItemId"]
68 except:
69     print("/api/v1/Tasks: Error reading response")
70     print(response)
71     print(response.text)
72     exit(1)
73
74 print("media_item_id: " + media_item_id)
75
76 # Step 2: Check tasks status in a loop
77 while True:
78     response = requests.get(f"{BASE_URL}/api/v1/Tasks",
79                             params={"mediaItemId": media_item_id}, headers=headers)
80     task_info = response.json()
81     if task_info["status"] == "Done":
82         break
83     print ("waiting for Task to be Done...")

```

```

84     time.sleep(5)
85
86 # Step 3: Create an export via GET /api/v1/Exports/{resourceType}
87 for task in task_info["tasks"]:
88     if task["type"] == taskName:
89         taskId = task["id"]
90
91 export_data = {
92     "taskId": taskId,
93     "resourceType": "TimedSegmentBlock" # or "Txt"
94 }
95
96 response = requests.post(f"{BASE_URL}/api/v1/Exports/{export_data['resourceType']}", params={
97     "taskId": export_data["taskId"], headers=headers)
98 export_response_data = response.json()
99 workflow_id = export_response_data["workflowId"]
100
101 # Step 4: Check export status in a loop
102 while True:
103     response = requests.get(f"{BASE_URL}/api/v1/Exports/status",
104                             params={"workflowId": workflow_id}, headers=headers)
105     if (response.status_code != 202):
106         try:
107             export_status = response.json()["status"]
108             media_resource_id = response.json()["mediaResourceId"]
109         except:
110             print("/api/v1/Exports/status: Error reading response")
111             print(response)
112             print(response.text)
113             exit(1)
114
115         if export_status == "Ready":
116             break
117     print ("waiting for Export to be Done...")
118     time.sleep(5)
119
120 # Step 5: Get export result via GET /api/v1/Exports/download
121 response = requests.get(f"{BASE_URL}/api/v1/Exports/download",
122                         params={"mediaResourceId": media_resource_id}, headers=headers)
123
124 try:
125     export_result = response.json()
126 except:
127     print("/api/v1/Exports/download: Error reading response")
128     print(response)
129     print(response.text)
130     exit(1)
131
132
133 output_path = "output.seg.txt"
134 print ("writing output to " + output_path + " ...")
135
136 with open(output_path, "w", encoding='utf-8') as json_file:
137     json_file.write(export_result["blob"])
138

```

3.3 Voiceover Task and Export Example

- Description: Upload an mp4 video, request a transcription, a translation and a voiceover, download the wav file with the voiceover audio.
- Python Script:

```
1  import os
2  import requests
3  import time
4  import json
5  import sys
6
7  # Example
8  # python .\voiceover.py .\file.mp4
9  # Need to have credentials.json file with:
10
11  #{
12  #   "apiKey": "",
13  #   "repositoryId": "",
14  #   "workspaceUserId": ""
15  #   "baseUrl": ""
16  #}
17  #
18  # NOTE: apiKey should be already the base64 version of the json with {"OrchestrationIdentityId":"","Orchestrati
19
20  input_path = sys.argv[1]
21  taskName = "Voiceover"
22
23  with open('credentials.json', 'r') as json_file:
24      data = json.load(json_file)
25      apiKey = data['apiKey']
26      repositoryId = data['repositoryId']
27      workspaceUserId = data['workspaceUserId']
28      BASE_URL = data['baseUrl']
29
30  headers = {
31      "Authorization": "Bearer " + apiKey
32  }
33
34  text_extensions = [".txt"]
35  audio_extensions = [".wav", ".mp3", ".flac", ".ogg", ".m4a", ".wma"]
36  video_extensions = [".mp4", ".ogm", ".wmv", ".mpg", ".webm", ".ogv", ".mpeg", ".mov", ".m4v", ".avi", ".mxf"]
37
38  for video_extension in video_extensions:
39      if input_path.endswith(video_extension):
40          fileType = "Video"
41          extension = video_extension
42
43  for audio_extension in audio_extensions:
44      if input_path.endswith(audio_extension):
45          fileType = "Audio"
46          extension = audio_extension
47
48
49  for text_extension in text_extensions:
50      if input_path.endswith(text_extension):
51          fileType = "Text"
52          extension = text_extension
53
54  # Step 1: Create tasks via POST /api/v1/Tasks
```

```

55 task_data = {
56     "upload": {
57         "itemTitle": "A Test Title",
58         "fileType": fileType,
59         "itemDescription": "Test Description",
60         "sourceLanguage": "en",
61         "filename": "file" + extension,
62         "repositoryId": repositoryId,
63         "sourceUrl": None,
64         "text": None,
65     },
66     "tasks": {
67         "workspaceUserIds": [workspaceUserId],
68         taskName: {
69             "SourceLanguage": "es",
70             "Voice": "Alvaro",
71             "ServiceVariant": "ES",
72             "ServiceProvider": "Azure",
73             "Gender": "Male"
74         }
75     }
76 }
77
78 form_data = {
79     "taskWorkflowConfiguration": json.dumps(task_data)
80 }
81
82 sent_files = {
83     'file': ('file' + extension, open(input_path, 'rb'), 'application/octet-stream')
84 }
85
86 response = requests.post(f"{BASE_URL}/api/v1/Tasks",
87                          data=form_data, headers=headers, files=sent_files)
88
89 try:
90     response_data = response.json()
91     media_item_id = response_data["mediaItemId"]
92 except:
93     print("Error reading response")
94     print(response)
95     print(response.text)
96     exit(1)
97
98 # Step 2: Check tasks status in a loop
99 while True:
100     response = requests.get(f"{BASE_URL}/api/v1/Tasks",
101                             params={"mediaItemId": media_item_id}, headers=headers)
102     task_info = response.json()
103     if response.status_code == 500:
104         raise Exception(json.loads(response.content)["detail"])#there was an error uploading the input media)
105     if task_info["status"] == "Done":
106         break
107     print ("waiting for Task to be Done...")
108     time.sleep(5)
109
110 # Step 3: Create an export via GET /api/v1/Exports/{resourceType}
111 for task in task_info["tasks"]:
112     if task["type"] == taskName:

```

```

113     taskId = task["id"]
114
115 export_data = {
116     "taskId": taskId,
117     "resourceType": "TranscriptVoiceoverAudio"
118     #"resourceType": "VoiceOverVideo" --> If you want the burned video instead
119 }
120
121 response = requests.post(f"{BASE_URL}/api/v1/Exports/{export_data['resourceType']}", params={
122     "taskId": export_data["taskId"],
123     "VideoAudioVolume": 0.7,
124     "VoiceOverAudioVolume": 1}, headers=headers)
125 export_response_data = response.json()
126 workflow_id = export_response_data["workflowId"]
127
128 # Step 4: Check export status in a loop
129 while True:
130     response = requests.get(f"{BASE_URL}/api/v1/Exports/status",
131                             params={"workflowId": workflow_id}, headers=headers)
132     if (response.status_code != 202):
133         try:
134             export_status = response.json()["status"]
135             media_resource_id = response.json()["mediaResourceId"]
136         except:
137             print("Error reading response")
138             print(response)
139             print(response.text)
140             exit(1)
141
142         if export_status == "Ready":
143             break
144     print ("waiting for Export to be Done...")
145     time.sleep(5)
146
147 # Step 5: Get export result via GET /api/v1/Exports/download
148 response = requests.get(f"{BASE_URL}/api/v1/Exports/download",
149                         params={"mediaResourceId": media_resource_id}, headers=headers)
150
151 try:
152     export_result = response.json()
153 except:
154     print("Error reading response")
155     print(response)
156     print(response.text)
157     exit(1)
158
159
160 #output_path = "output.mp4" --> if you want the burned video instead
161 output_path = "output.wav"
162 print ("writing output to " + output_path + " ...")
163
164 responseVideoDownload = requests.get(export_result["url"])
165
166 with open(output_path, "wb") as video_file:
167     video_file.write(responseVideoDownload.content)
168

```