



Stream Learning for Multilingual Knowledge Transfer

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

D4.6 Demonstrator for use case two

Work Package	4
Responsible Partner	Priberam
Author(s)	Afonso Mendes
Contributors	Sebastião Miranda
Reviewer	Peggy van der Kreeft
Version	1.0
Contractual Date	31 March 2024
Delivery Date	28 March 2024
Dissemination Level	Public

Version History

Version	Date	Description
0.1	14/03/2024	Initial version
0.2	18/03/2024	Internal review
0.3	19/03/2024	Final updates
1.0	25/03/2024	Publishable version

Executive Summary

This document confirms that the plain X demonstrator for the multilingual News Content Production - Use Case 2 – was made available for internal user evaluation and final users at <https://app.plaix.com>. To request access to plain X, go to [Free Trial - plain X](#).

A detailed description of the work done for the UC2 demonstrator can be found in D1.4 (Final Prototype Report). The evaluation of the demonstrator was reported in D5.3 (Final Evaluation Report).

Table of Contents

***Executive Summary*..... 3**

***1. Demonstrator for UC2 – plain X*..... 6**

***2. Integration & Orchestration* 11**

 2.1 Job Orchestration in plain X.....11

 2.2 RabbitMQ in plain X.....13

 2.3 plain X Automations API14

***3. Evaluation*..... 15**

***4. Conclusion*..... 16**

Table of Figures

FIGURE 1 PLAIN X LIBRARY SHOWING AVAILABLE VIDEOS TO WORK ON, AND PLAIN X BOARD PAGE.....	6
FIGURE 2 PLAIN X TRANSCRIPTION TASK PAGE.....	7
FIGURE 3 PLAIN X TRANSLATION TASK PAGE, SHOWING A VIDEO TRANSCRIPT IN THE ORIGINAL AND TARGET LANGUAGE	8
FIGURE 4 PLAIN X VOICE-OVER TASK PAGE, SHOWING THE GUI COMPONENT TO HIGHLIGHT TEXT SEGMENTS AND CUSTOMIZE SYNTHETIC VOICE GENERATION PARAMETERS.....	9
FIGURE 5 PLAIN X SETTINGS PAGE WHICH ALLOWS CUSTOMIZING THE WORKSPACE IN TERMS OF PREFERRED NLP ENGINES FOR EACH LANGUAGE (PAIR)	9
FIGURE 6 PLAIN X AUDIO WAVE FORM COMPONENT.	10
FIGURE 7 PLAIN X ADD TASK FORM WITH THE SELMA “LIA” SPEECH TRANSLATION MODEL SELECTED	10
FIGURE 8 EXAMPLE OF A PLAIN X JOB WORKFLOW GRAPH SHOWCASING THE DEPENDENCIES BETWEEN THE DIFFERENT ORCHESTRATED JOBS. THIS GRAPH IS RAN BY MAESTRO.....	12
FIGURE 9 PLAIN X HIGH LEVEL ORCHESTRATION ARCHITECTURE, SHOWCASING THE INTEGRATION BETWEEN PLAIN X, RABBITMQ AND MAESTRO	13
FIGURE 10 PLAIN X’S RABBITMQ INSTANCE OVERVIEW, SHOWING A FEW GLOBAL STATISTICS OF JOB QUEUE MESSAGE PROCESSING.....	14

1. Demonstrator for UC2 – plain X

This document confirms that plain X, a demonstrator for the multilingual News Content Production - Use Case 2 – is available for user evaluation (available at <https://app.plainx.com>).

plain X has mostly been developed under the SELMA project from a first prototype named “news.bridge”. This document provides a list of figures documenting some important features.

Figure 1 shows the entry point of the platform, the Library, a place where a user can find items they wish to work on (transcribe, translate, subtitle or voiceover).

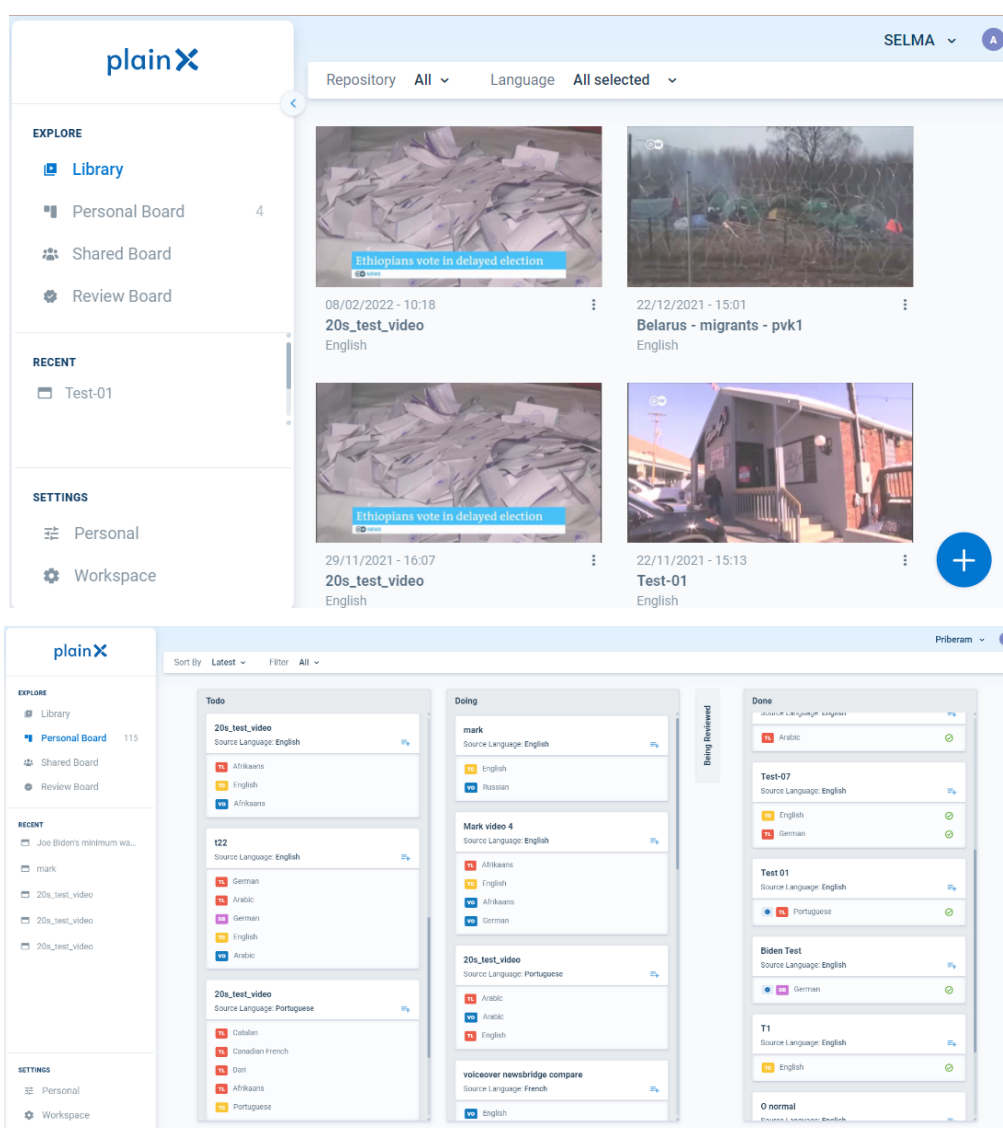


Figure 1 plain X library showing available videos to work on, and plain X board page

Figure 2 illustrates 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.

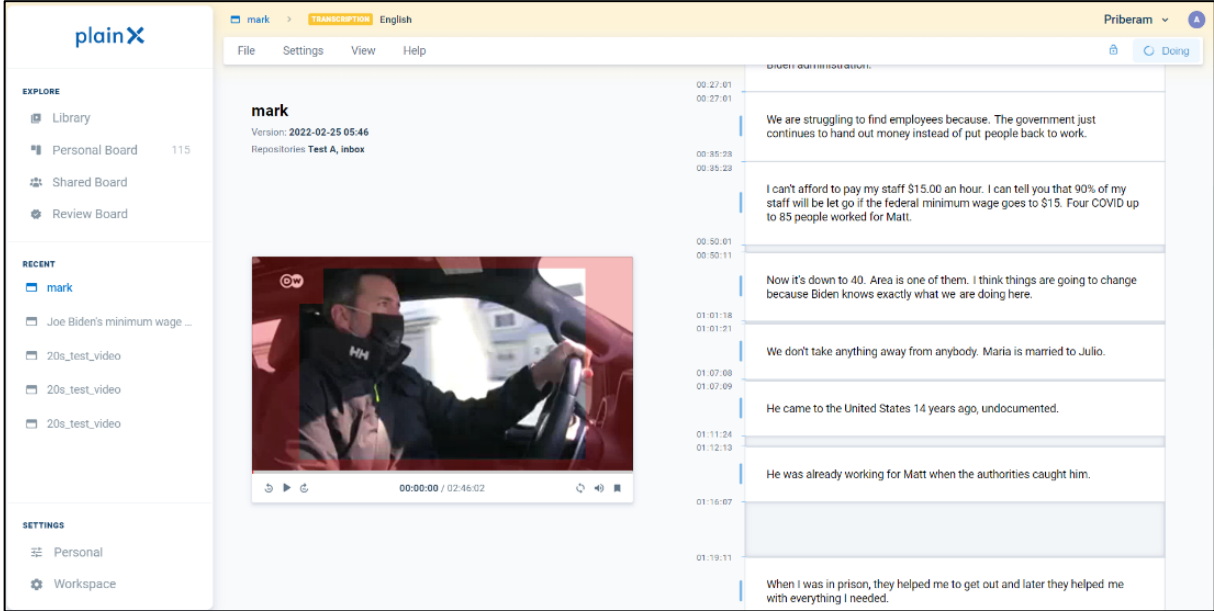


Figure 2 plain X transcription task page

In Figure 3, we show the Translation task page which starts by showing the user an automatic translation and then allows them to edit this translation. User edits are saved for future model improvement.

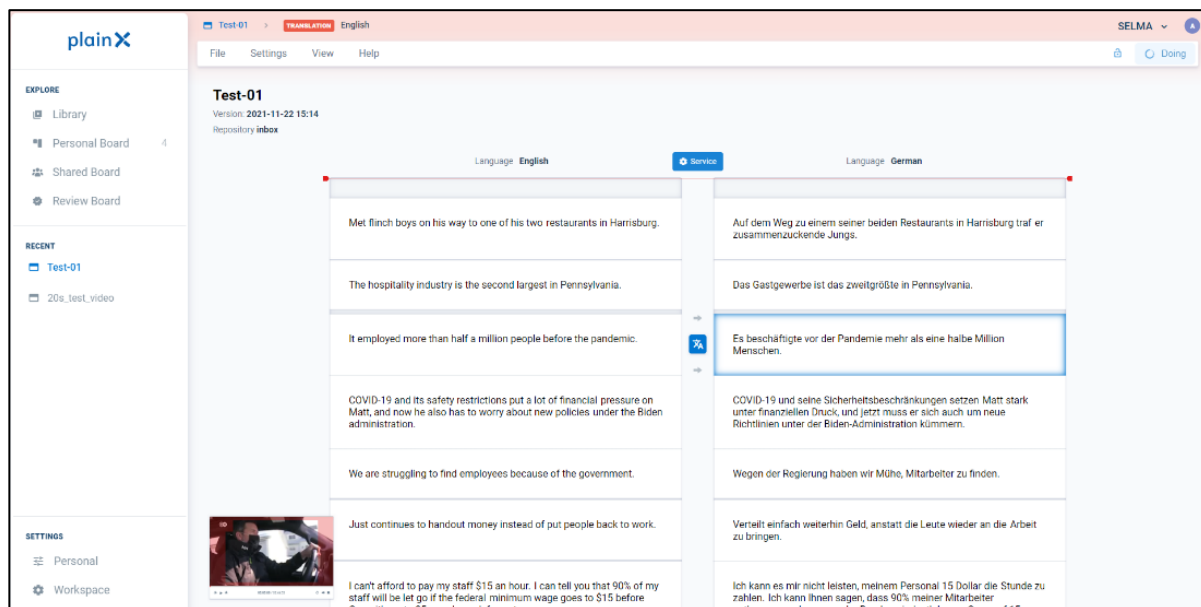


Figure 3 plain X Translation task page, showing a video transcript in the original and target language

In Figure 4, 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.

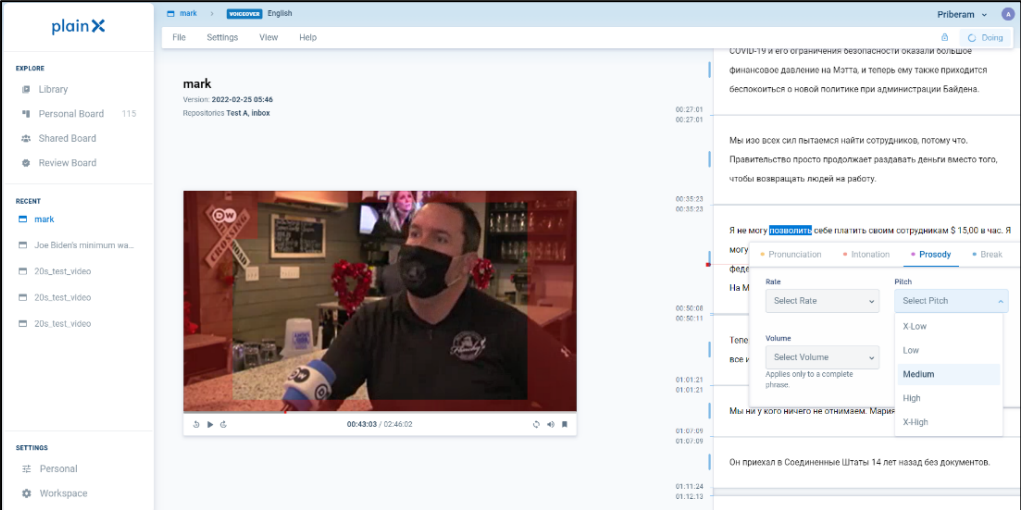


Figure 4 plain X Voice-over task page, showing the GUI component to highlight text segments and customize synthetic voice generation parameters

Figure 5 shows the user customization of the default engines that should be used for each language or language pair, as shown.

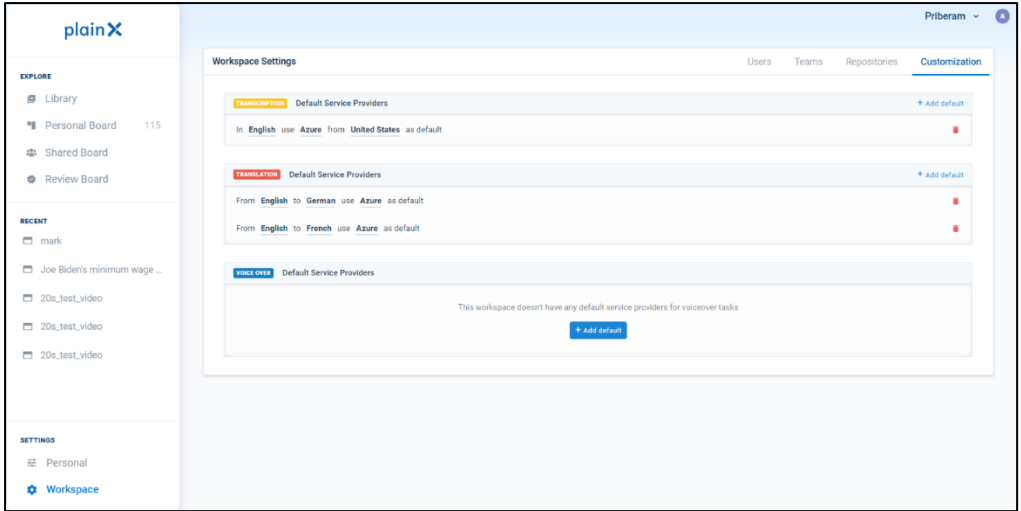


Figure 5 plain X Settings page which allows customizing the workspace in terms of preferred NLP engines for each language (pair)

Figure 6 shows an optional wave form preview component to facilitate adjusting the timecodes of the transcription, plain X

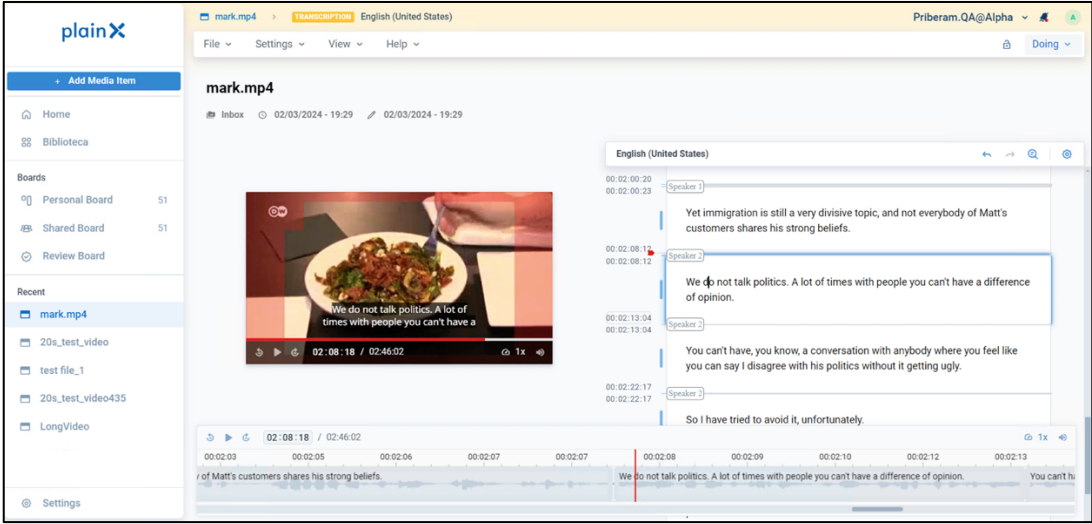


Figure 6 plain X Audio Wave form component.

Figure 7 shows the selection of one of the SELMA models, in this case the LIA speech translation from French to English.

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

- Type:** A dropdown menu with 'TRANSCRIPTION' selected.
- Language:** A text input field containing 'English'.
- Upload Text File:** A text input field with the placeholder 'Import an existing utf-8 text file' and a blue upload icon.
- Service Provider:** A dropdown menu with 'SELMA' selected.
- Service Variant *:** A dropdown menu with 'LIA (French->English)' selected.
- Task Editors:** A dropdown menu with 'Administrator' selected and a blue dot indicator.
- Current task Reviewers:** A dropdown menu with 'Select Reviewers' selected and a blue dot indicator.

 At the bottom right of the form are 'Cancel' and 'Finish' buttons.

Figure 7 plain X Add Task Form with the SELMA “LIA” Speech Translation model selected

2. Integration & Orchestration

plain X takes advantage of the worker management and scalability of DockerSpaces and the SELMA Maestro orchestrator (See WP4 deliverables).

The models/components integrated in plain X are:

- SELMA Speech Recognition
- SELMA Speech Translation
- SELMA Text to Speech
- SELMA Orchestration (Maestro, Docker Spaces)
- Open-Source Machine Translation (Meta's m2m_100)
- Open-Source Speech to Text (Whisper)
- External Service Providers (Azure, Google, DeepL, Speechmatics, ...)

All these components have been packaged in docker containers and integrated as RabbitMQ workers, which can be launched with Docker Spaces.

2.1 Job Orchestration in plain X

plain X uses the same SELMA orchestration core system as in Monitio, which leverages the Maestro component to orchestrate job graphs (See WP4 deliverables) and connects through RabbitMQ to NLP workers hosted in the cloud, in specific addresses or managed through Docker Spaces. Below is an example of such a processing job graph.

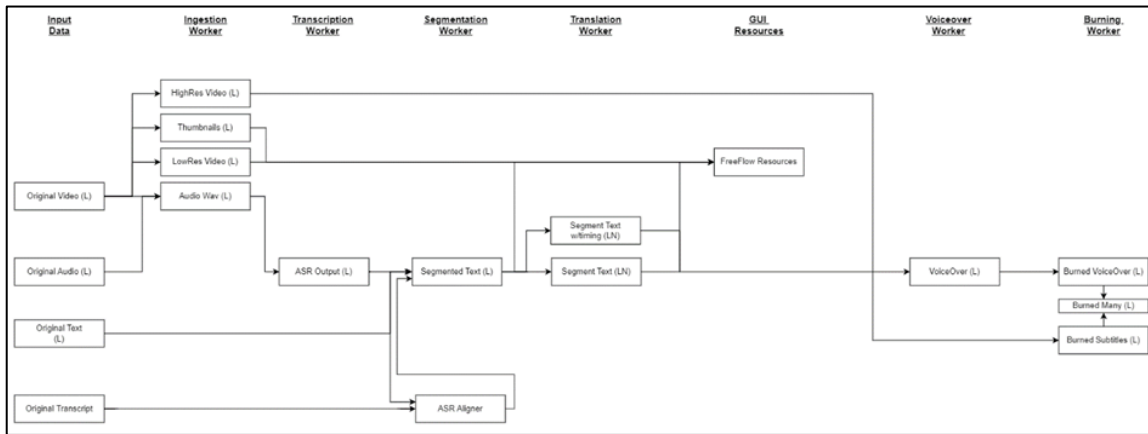


Figure 8 Example of a plain X job workflow graph showcasing the dependencies between the different orchestrated jobs. This graph is ran by Maestro.

The software architecture of the job orchestration in plain X is organized into several components. The Maestro orchestrator serves as the core orchestrator component which manages the execution of the job workflow graphs through RabbitMQ. On the plain X backend side there are service bindings which contain the data integration scripts to convert the job results into plain X specific database updates. This is represented in the following picture.

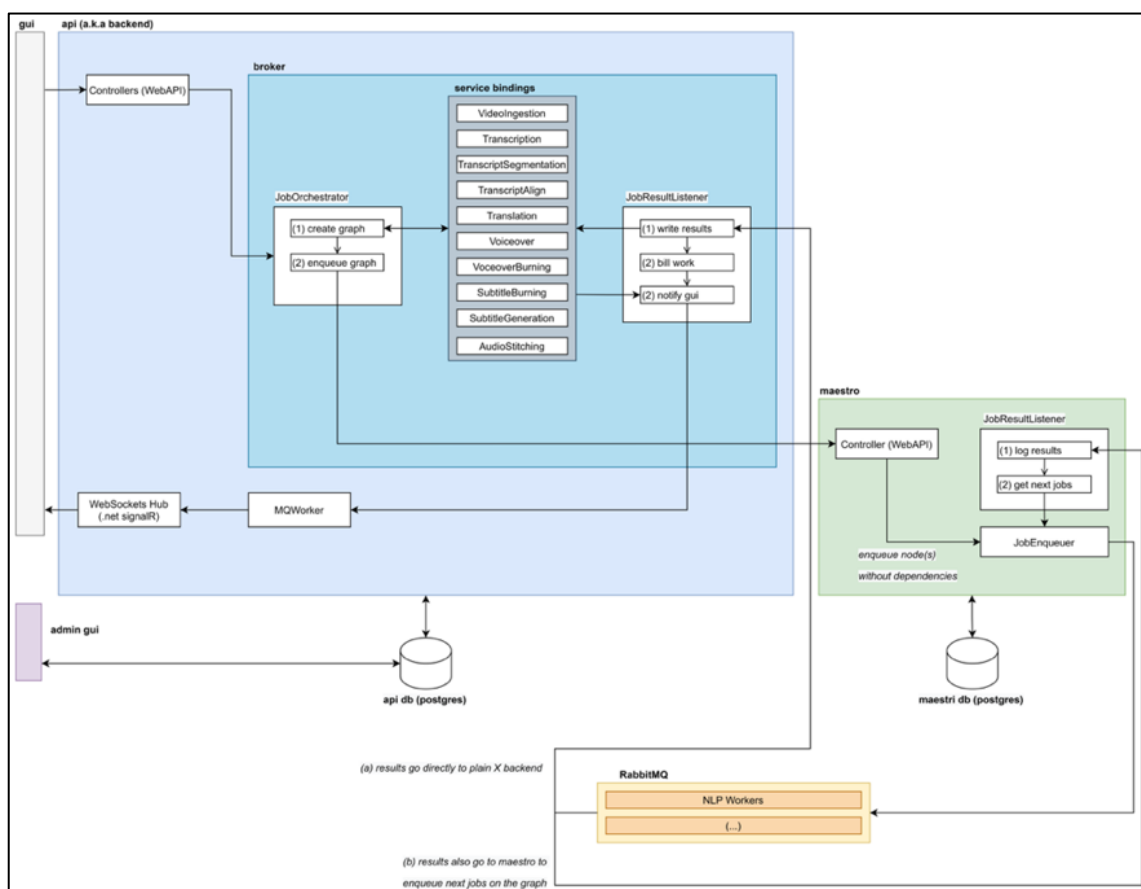


Figure 9 plain X high level orchestration architecture, showcasing the integration between plain X, RabbitMQ and Maestro

2.2 RabbitMQ in plain X

RabbitMQ is used to manage processing jobs in the plain X orchestration pipeline. plain X has less intensive requirements than Monitio since processing is only proportional to the content uploaded by users and not through scraping internet sources. Still, it uses the same orchestration system which is very beneficial in terms of development synergy between the two products.

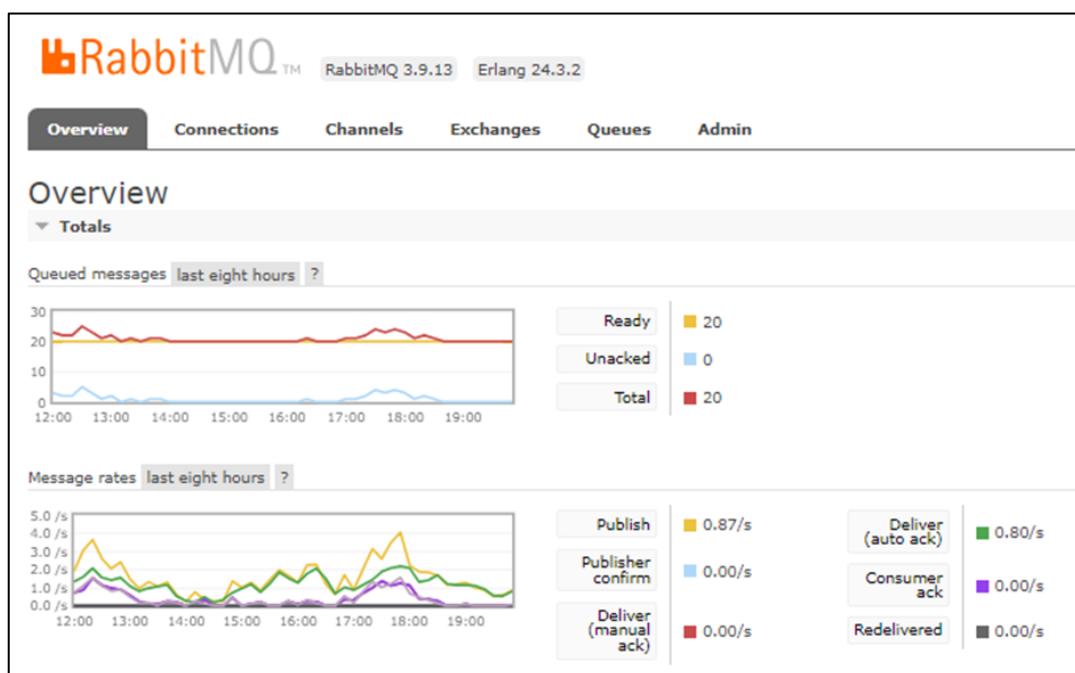


Figure 10 plain X's RabbitMQ instance overview, showing a few global statistics of job queue message processing

2.3 plain X Automations API

The plain X automations API enables the creation of transcription, translation, subtitling and voiceover jobs and the corresponding export of the final results directly through a REST API, without using the user interface. This enables integrating plain X with many other software tools. This was used within the SELMA project for example in the Podcast Creator App (See D1.4). Documentation for this API is annexed to deliverable D1.4.

3. Evaluation

The demonstrator evaluation was carried out within WP5 and reported in deliverable D5.3 (Final Evaluation Report). The demonstrator is being used at DW and the plain X API is being used by the Podcast Creator. The demonstrator has also been shown or given access to potential clients and members of the user group. Priberam has already managed to engage the first clients. During the project user feedback has continuously been integrated in the development and research cycles either to improve the underlying ML models or the UI itself.

4. Conclusion

The demonstrator has integrated the multilingual and user feedback components developed within the SELMA project. The use of the demonstrator by the testers and the user group provided excellent feedback on usability. plain X workflow and integrated models reached a new level for the content production use case.