Research and Innovation Action (RIA) H2020 – 957017


Stream Learning for Multilingual Knowledge Transfer

https://selma-project.eu/

# D4.2 Initial platform release with the primary NLP pipeline

| | |
|---|---|
| Work Package | 4 |
| Responsible Partner | IMCS |
| Author(s) | Guntis Barzdins, Sebastião Miranda |
| Contributors | |
| Reviewer | Christoph Andreas Schmidt |
| Version | V1.0 |
| Contractual Date | 31 March 2022 |
| Delivery Date | 31 March 2022 |
| Dissemination Level | Public |

# Version History

| Version | Date | Description |
|---|---|---|
| 0.1 | 05/01/2022 | Initial Table of Contents (ToC) |
| 0.2 | 09/03/2022 | Use Case 0 description added |
| 0.3 | 10/03/2022 | Executive summary added |
| 0.4 | 30/03/2022 | Internal Review version |
| 0.5 | 30/03/2022 | Finalization |
| 1.0 | 31/03/2022 | Publishable version |

# Executive Summary

The Initial SELMA platform release with the primary NLP pipeline consists of three demonstrators corresponding to the three Use Cases UC0, UC1, UC2 identified within the SELMA project. Although deeply integrated on the NLP backend, these three demonstrators have vastly different frontend GUIs and NLP pipelines tailored to the specific Use Case.

SELMA's central concept is to build a deep-learning NLP platform that trains unsupervised language models, using a continuous stream of textual and video data from media sources and make them available in a user/topic-oriented form in over 30 languages.

This document provides an overview of the initial SELMA platform release to be followed by the interim and final releases later in the project.

# Table of Contents

# Table of Figures

# 1. Introduction

SELMA Platform consists of two demonstrators tailored to the two primary use cases:

- Use Case 1 (UC1) Media Monitoring (based on the Monitio Platform)
- Use Case 2 (UC2) News Production (based on the plain X Platform)

Use cases UC1 and UC2 are being 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 with the SELMA research results which will allow the community to build stream processing software pipelines. This platform core is also currently in use in plain X (UC2) and is currently being integrated in Monitio (UC1) - by applying the SELMA platform core to these two different use cases we're also proving that it's generic enough to be useful by the Open-Source community in other NLP applications/products.

Use Case 0 was introduced in the SELMA deliverable D4.1 "Platform Architecture and API Documentation", focusing exclusively on the NLP modules developed within the SELMA WP2, WP3 and enabling their independent testing and integration into the highly scalable **SELMA NLP Service** consumed by the primary use cases UC1 and UC2.

This deliverable describes the release of the initial platform demonstrators for the UC0, UC1 and UC2. See D1.2 for more details regarding the corresponding prototypes.

# 2. Use Case 0: SELMA NLP Service Orchestration Platform

SELMA NLP Service Orchestration Platform[1], available online at https://selma-project.github.io/, provides Basic Testing and Configuration Interface (see Figure 1) for testing, deployment, scaling and monitoring of the NLP services developed within the SELMA project WP2 and WP3. The NLP worker deployment uses the TokenQueue mechanism (described in D4.1) to deliver highly scalable **SELMA NLP Service** for the primary Use Cases UC1, UC2 described in the following Sections.

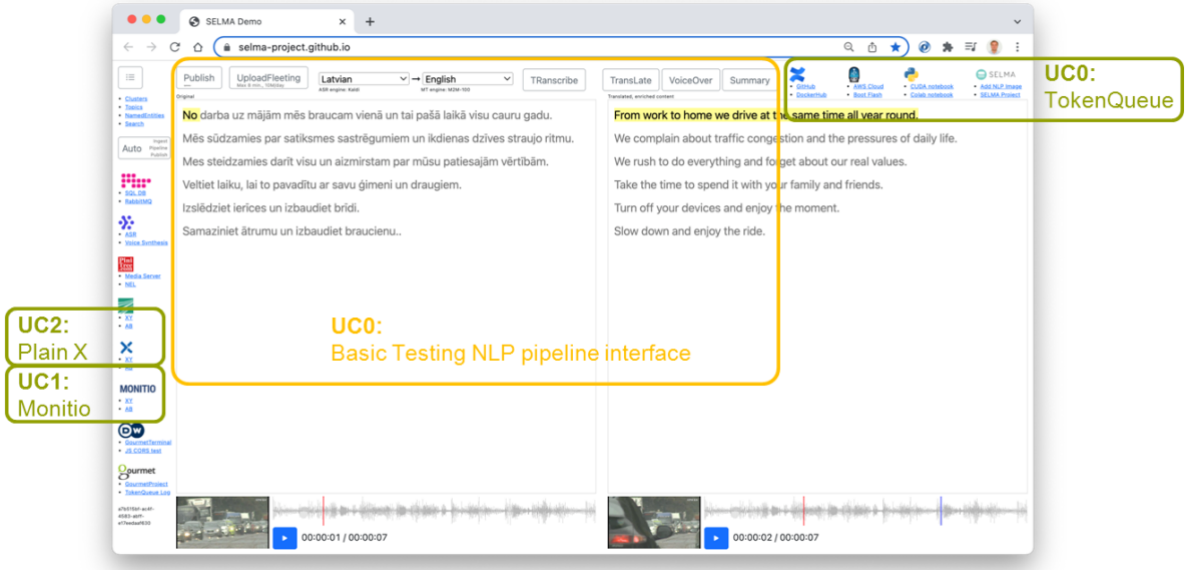
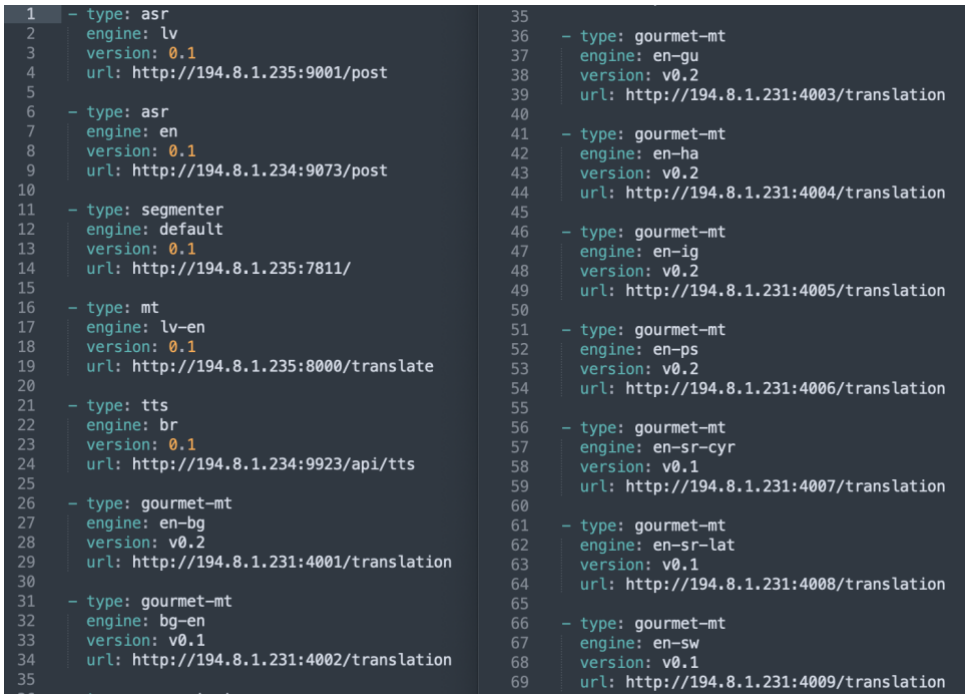



***Figure 1*** *SELMA Basic Testing and Configuration Interface (UC0)*

UC0 integrates with NLP-pipeline execution orchestrator Maestro (described in D4.1) shared with the UC1 and UC2. Maestro Orchestrator serves as a gateway between all three Use Cases UC0, UC1, UC2, allowing them to share the same NLP worker pool defined in the

[1] https://github.com/SELMA-project/selma-v8

TokenQueue. In terms of control flow integration, Maestro Orchestrator uses Token Queue as a REST API service.



```yaml
- type: asr
  engine: lv
  version: 0.1
  url: http://194.8.1.235:9001/post

- type: asr
  engine: en
  version: 0.1
  url: http://194.8.1.234:9073/post

- type: segmenter
  engine: default
  version: 0.1
  url: http://194.8.1.235:7811/

- type: mt
  engine: lv-en
  version: 0.1
  url: http://194.8.1.235:8000/translate

- type: tts
  engine: br
  version: 0.1
  url: http://194.8.1.234:9923/api/tts

- type: gourmet-mt
  engine: en-bg
  version: v0.2
  url: http://194.8.1.231:4001/translation

- type: gourmet-mt
  engine: bg-en
  version: v0.1
  url: http://194.8.1.231:4002/translation

- type: gourmet-mt
  engine: en-gu
  version: v0.2
  url: http://194.8.1.231:4003/translation

- type: gourmet-mt
  engine: en-ha
  version: v0.2
  url: http://194.8.1.231:4004/translation

- type: gourmet-mt
  engine: en-ig
  version: v0.2
  url: http://194.8.1.231:4005/translation

- type: gourmet-mt
  engine: en-ps
  version: v0.2
  url: http://194.8.1.231:4006/translation

- type: gourmet-mt
  engine: en-sr-cyr
  version: v0.1
  url: http://194.8.1.231:4007/translation

- type: gourmet-mt
  engine: en-sr-lat
  version: v0.1
  url: http://194.8.1.231:4008/translation

- type: gourmet-mt
  engine: en-sw
  version: v0.1
  url: http://194.8.1.231:4009/translation
```

***Figure 2*** *A fragment of the deployed TokenQueue worker.yaml file (UC0)*

Technically, the UC0 software release consists of three components:

- TokenQueue backend handling the scalable deployment of the SELMA NLP workers through the long-polling TokenQueue protocol (see example in Figure 3). TokenQueue is controlled by the dynamically updated list of available NLP workers (Figure 2).
- The actual NLP workers deployed on various servers within the SELMA Consortium and accessed in a uniform manner via the TokenQueue backend. The list of currently deployed workers (see Figure 2) includes both SELMA WP2, WP3 outcomes as well as other open-source NLP workers such as neural machine translation from the GoURMET H2020 project Deutsche Welle is participating in (https://gourmet-project.eu/).
- Highly scalable UC0 GUI frontend (Figure 1) served directly from the GitHub repository of the SELMA project (https://selma-project.github.io/) implementing the basic SELMA NLP pipeline for testing, monitoring and configuration purposes.

```
{"action":"acquire","id":"hkm9ag","query":{"engine":"lv","timeout":0,"type":"asr","version":"","wait":0},"unixtime":1646388027}
{"action":"acquired","id":"hkm9ag","url":"http://194.8.1.235:9001/post","unixtime":1646388027}
{"action":"release","url":"http://194.8.1.235:9001/post","unixtime":1646388030}
{"action":"acquire","id":"2of9e3","query":{"engine":"","timeout":0,"type":"segmenter","version":"","wait":0},"unixtime":1646388030}
{"action":"acquired","id":"2of9e3","url":"http://194.8.1.235:7811/","unixtime":1646388030}
{"action":"release","url":"http://194.8.1.235:7811/","unixtime":1646388030}
{"action":"acquire","id":"rp98k5","query":{"engine":"lv-en","timeout":0,"type":"mt","version":"","wait":0},"unixtime":1646388038}
{"action":"acquired","id":"rp98k5","url":"http://194.8.1.235:8000/translate","unixtime":1646388038}
{"action":"release","url":"http://194.8.1.235:8000/translate","unixtime":1646388041}
```


***Figure 3*** *A fragment of the TokenQueue protocol log-file (UC0)*

All three UC0 components are highly scalable by design. Although the actual scalability tests are scheduled in the later stages of the project, we currently estimate that the TokenQueue backend on the appropriate hardware would scale to 10M TokenQueue protocol requests per day. Frontend component is static and served directly from the SELMA project public github repository (with presumably unlimited bandwidth). The actual NLP workers are independent from each other and therefore can be scaled indefinitely by increasing their number (within the local or cloud compute resources).

# 3. Use Case 1: Monitio

Within Use-Case 1 (UC1), we're 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. The frontend is implemented in VueJs (vuejs.org), the backend is a REST API implemented in dotnet (dotnet.microsoft.com), and storage using PostgreSQL (postgresql.org). We're in the phase of migrating the platform to start using the SELMA Service Orchestration Platform (UC0) as it's NLP job orchestrator.

See D1.2 for details on the prototype.

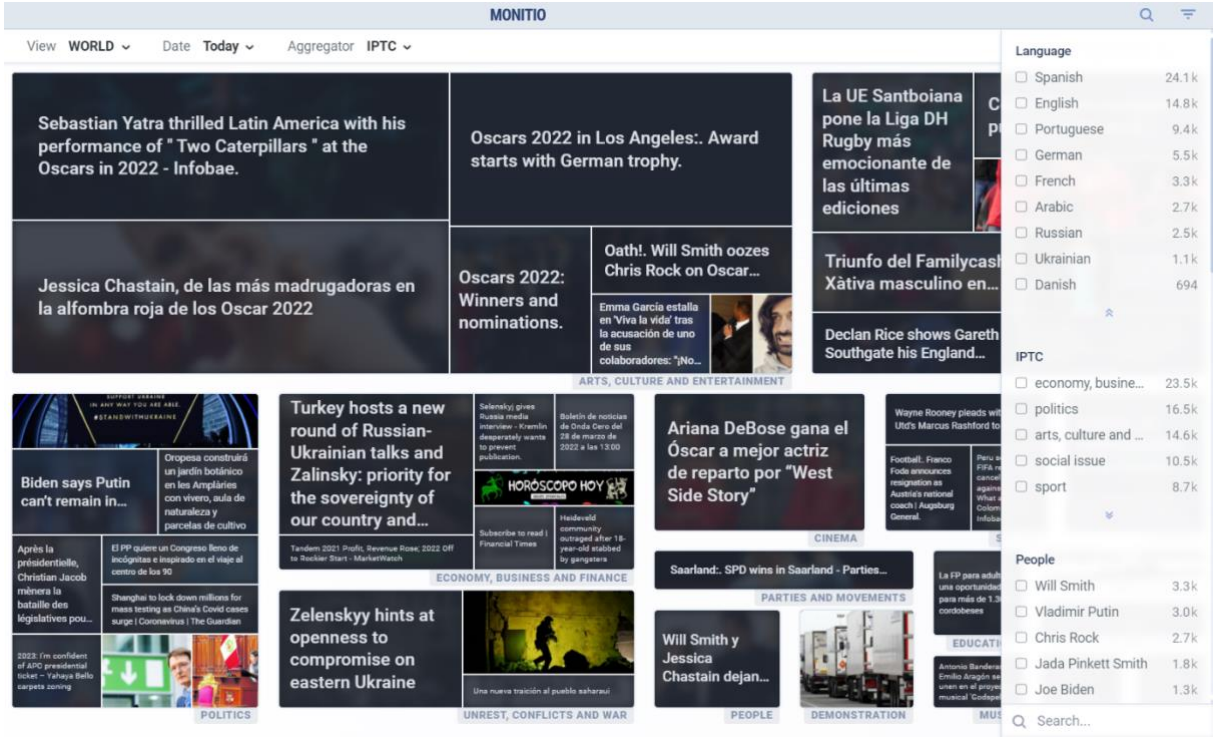



***Figure 4*** *Monitio user iinterface (UC1)*

Some of these research results are improved models using transfer and stream learning (see D2.1 for technical details), while others are changes to the platform to allow learning from User Feedback. The components released in D3.2 were integrated in UC1. We're also currently refactoring the Monito platform to use the SELMA orchestration platform as its NLP job orchestrator.

# 4. Use Case 2: plain X

Within Use-Case 2 (UC2), we're 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 frontend is implemented in VueJs (vuejs.org), the backend is a REST API implemented in dotnet (dotnet.microsoft.com), and storage using PostgreSQL (postgresql.org). The platform is already using the SELMA Service Orchestration Platform (UC0) as its NLP job orchestrator.

See D1.2 for details on the prototype.

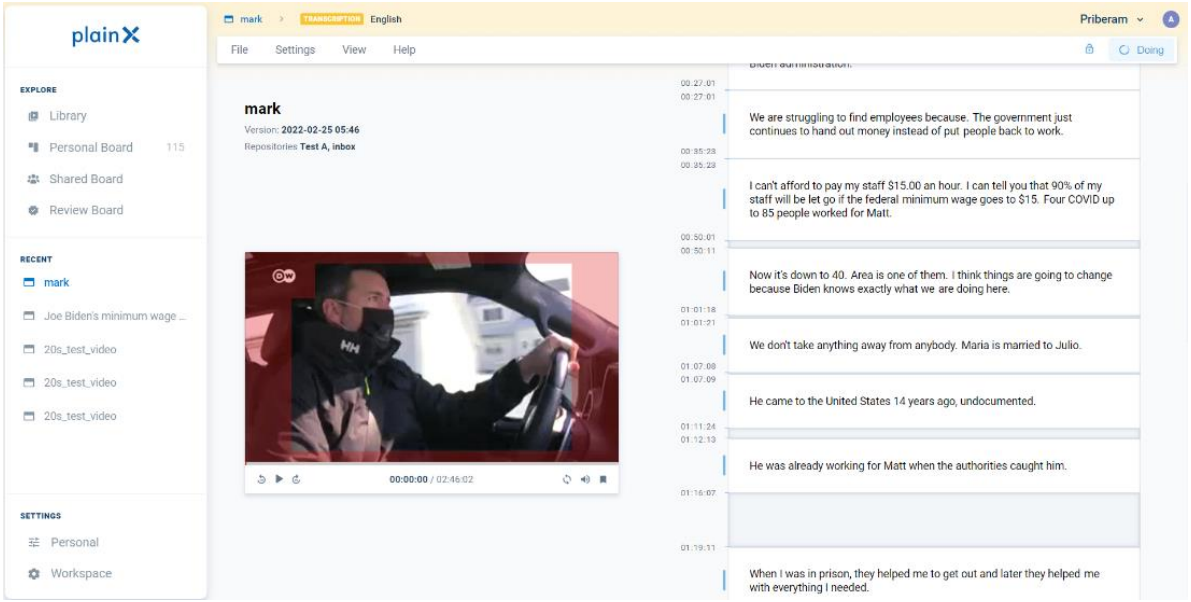



***Figure 5*** *plain X user interface (UC2)*

The components released in D2.2 were integrated in UC2. plain X is using the SELMA orchestration to schedule and execute NLP jobs (see D4.1). In the case of plain X, the SELMA orchestration allows to execute NLP jobs not only from both self-hosted APIs (e.g., the ones developed within SELMA), but also from many cloud providers (Azure, Google, etc).

# 5. Conclusion

This document presents the initial SELMA platform release with the primary NLP pipeline. To foster commercial exploitation and sustainability of the project results, the SELMA platform release consists of open-source UC0 and commercial UC1 and UC2 demonstrators integrated through the common backend technology. We'll continue integrating the SELMA platform components as they are developed and also new developments on the SELMA orchestration core in the UC1 and UC2 platforms.