D7.2. DESCRIPTION OF THE INTEGRATION TOOLKIT GUIDELINES AND PLAN

Grant Agreement:	833635
Project Acronym:	ROXANNE
Project Title:	Real time network, text, and speaker analytics for combating organised crime
Call ID: **Call name:**	H2020-SU-SEC-2018-2019-2020, Technologies to enhance the fight against crime and terrorism
Revision:	V1.0
Date:	30 May 2020
Due date:	30 June 2020
Deliverable lead:	AIRBUS
Work package:	WP7
Type of action:	RIA

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

Disclaimer

The information, documentation and figures available in this deliverable are written by the “ROXANNE - Real time network, text, and speaker analytics for combating organised crime” project’s consortium under EC grant agreement 833635 and do not necessarily reflect the views of the European Commission.

The European Commission is not liable for any use that may be made of the information contained herein.

Copyright notice

Project co-funded by the European Commission within the H2020 Programme (2014-2020)		
Nature of deliverable:	R	
Dissemination Level		
PU	Public	☐
CO	Confidential, only for members of the consortium (including the Commission Services)	☒
EU-RES	Classified Information: RESTREINT UE (Commission Decision 2015/444/EC)	☐
* R: Document, report (excluding the periodic and final reports) DEM: Demonstrator, pilot, prototype, plan designs DEC: Websites, patents filing, press & media actions, videos, etc. OTHER: Software, technical diagram, etc.		

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 833635.

Revision history

Revision	Edition date	Author	Modified Sections / Pages	Comments
V0.1	16 March 2020	AIRBUS	All	Editing all the sections
V0.2	25 May 2020	AIRBUS	All	Editing all the sections
V0.3	25 May 2020	PHO		structure of speech technologies added; diagram
V0.4	10 June 2020	AEGIS	All	Review and updates to Section 8
V0.5	10 June 2020	Airbus	3	add section 3 - integration guidelines
V0.6	18 June 2020	Airbus	All	describe services and their interfaces
V0.7	19 June 2020	LUH	4	add conversation information in output of speech/text processing chain
V0.8	19 June 2020	SAIL	All	Review, updated descriptions of SAIL components
V0.9	24 June 2020	LUH	4.9 7	add conversation information into the output of voice processing chain add description for network analysis
	24 June 2020	ITML	7	add section for Network Analysis
	30 June 2020	LUH	4.9 7	add topic information into the output of voice processing chain add examples in Section 7.1
	02 July 02 July	IDIAP IDIAP	5.3 All	added topic information and interface example Corrected several typos, added some elements related to speech + some comments/questions
v1.0	03 July	AIRBUS	All	Final minor corrections.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

Executive summary

ROXANNE, a novel platform combining advances of speech, language and video technologies and criminal network analysis for supporting investigators in their daily work especially on large criminal cases, has as its main goal to speed up the investigative processes, as well as to reduce the cost and burden to the society caused by organized crime activities. ROXANNE focuses on typical investigation processes where a significant amount of information is collected from telecommunication sources (e.g. wiretaps, interview recordings or audio provided by social media, complemented by video and geographical information). Usually two particular but separated approaches are employed in practice by LEAs: (i) identification of individuals from media (audio, video) by means of speaker identification, and (ii) analysis of relations among individuals and the whole structure of criminal network derived from various kinds of police data (i.e. criminal network analysis). The main objective of ROXANNE is to develop novel statistical methods and a platform to interface these two approaches (i.e. speaker identification and criminal network analysis technologies) to substantially increase their performance which will naturally lead to better investigation and identification capabilities of LEAs. ROXANNE also includes multilingual automatic speech recognition, natural language processing, video analysis and relation analysis in extraction of information from the available data.

To achieve these goals, to ingest and process huge volumes of heterogeneous data, and to derive useful information out of them, a highly available and resilient platform is needed. This document is describing the ROXANNE Platform Architecture, the detailed interfaces of each components and the integration guidelines to follow to integrate a component in the ROXANNE platform.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

Table of contents

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

1. Introduction

1.1. Purpose of the document

This document is the second deliverable of WP7 : D7.2 Description of the integration toolkit, guidelines and plan (M9, leader: AIRBUS); it describes the way to integrate components into the ROXANNE platform.

1.2. Document structure

Section 2 presents the high-level architecture used in ROXANNE project to ingest data, orchestrate its processing and then visualise results efficiently.

Section 3 describes the guidelines for each technical partners to build and deliver their components for an integration in ROXANNE platform.

Section 4 describes processing of audio data.

Section 5 describes processing of text data.

Section 6 describes processing of video data.

Section 7 describes network analysis.

Section 8 describes interfaces to be used by GUI components to retrieve data and processing results.

1.3. Technical definitions

This paragraph contains some technical definitions that will be used in this deliverable and all along the project.

API – a set of protocols, routines and tools for building software and applications. The purpose of an API is to express the functionalities of a software in terms of the interface (e.g., operations, inputs, outputs, underlying types, etc.) allowing for changes in implementation without compromising the interface itself. API can be a simple specification of remote calls exposed to consumers (e.g., SOAP and REST services), a library (e.g., specifications for routines, classes, variables, etc.), a set of classes with associated list of class methods (e.g., documentation of all the kinds of objects one can derive from the class definitions, and their associated possible behaviours).

Note: API is associated with both frameworks and libraries. Certain behaviour can be implemented by a library or built into a framework, in either case, described by an API.

Framework – a universal and reusable software environment that provides a specific functionality supposed to be part of a larger software application. The purpose of a framework is to ease the development of software applications, products and solutions. Frameworks provide a general solution that can be made more specific by the end-user by adding some user-written code.

Platform (specifically, the ROXANNE Platform) – The software which will be developed as an outcome of the ROXANNE project, which is an extensible analytics platform to support LEAs with advanced investigation capabilities, combining speech, language and video technologies with criminal network analysis.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

Component – Each one of the interacting parts of the ROXANNE platform which is responsible for the processing of different data types, e.g., Speech and speaker analytics component, as well as visualising outputs in a user-friendly way.

Microservice – Way to develop each ROXANNE component in an independent and loosely coupled process

Technology/ies – The element of a component which is responsible for carrying out a specific function, e.g., Speaker identification.

Containers – A standard unit of software that packages up code and all its dependencies and libraries so the application runs quickly and securely by taking advantage of Operation-system-level virtualization. All ROXANNE components are packaged in containers.

Orchestrator – Tool used to orchestrate containers: deploy, start, stop, scale containers inside the ROXANNE platform.

2. ROXANNE High Level Architecture

Detailed information about the architecture of the ROXANNE platform was given in the deliverable D7.1 Technical Specifications and Detailed Architecture Report. We would like to remind the reader of the ROXANNE high-level architecture, before explaining the integration and processing of individual components in detail.

The high-level architecture of ROXANNE can be summarized as follows

- During data ingestion, 3 processes are running at the same time
  - Pushing raw data in Ceph object storage
  - Pushing corresponding metadata in a Kafka topic: filename, URI to raw data on object storage, date…
  - Consuming Kafka's queue (topic) for storing metadata in the Case Management System
- Nifi is used for data processing orchestration:
  - By listening to Kafka input topics
  - By calling REST microservices to process / enrich data
  - By finally pushing processed data in Elasticsearch indices, which is periodically retrieved for visualization purposes
- All components are running as containers in a Kubernetes cluster
- Stateful components are running as statefulset kubernetes components (always up, resilient data, stable network identities):
  - Apache Nifi
  - Apache Kafka
  - Elasticsearch master / data nodes
- Processing Lightweight Microservices are running as Kubernetes workload
  - Auto scaling
  - Ephemeral
  - Resources sharing (CPU, RAM, GPU, Data volumes, Object storage)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

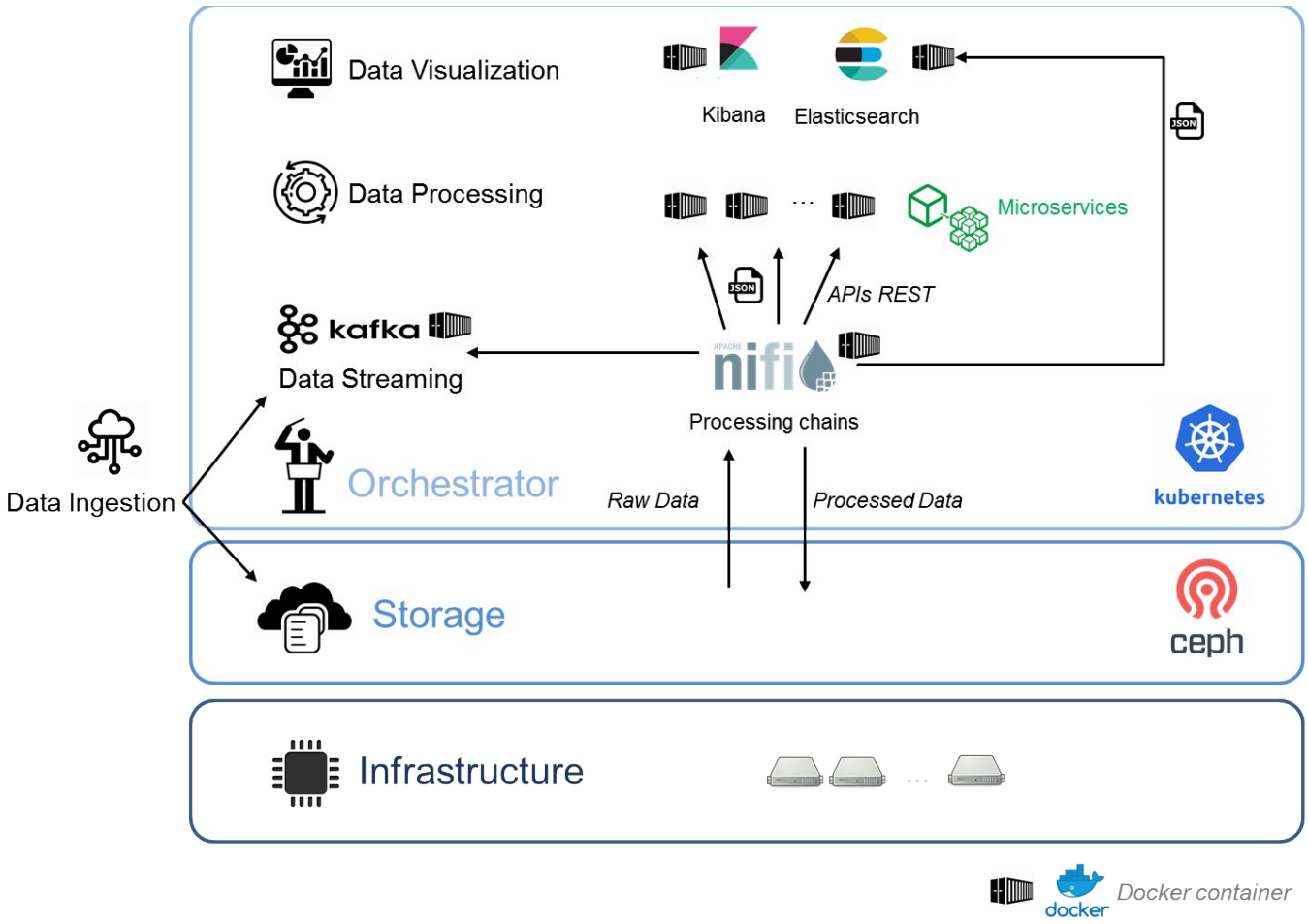


Figure 1 - ROXANNE High Level Architecture

3. Integration Guidelines

This section describes the way for ROXANNE technical partners to provide their components to Airbus for integration in the ROXANNE platform.

3.1 Containerized microservices

Each component should:

- Propose a JSON REST interface as described in the following sections,
- Be packaged in a Docker container,
- Be self sufficient (all needed external libraries already installed in the container)

3.2 Documentation

Each component should be delivered with following documentation:

- Detailed description of JSON interface with input / output examples
- Detailed description of Docker run command used to launch the container

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

- When available, description of hardware resources (RAM, CPUs, GPUs, disk space) needed by the container to process data
- Dockerfile used to build the container so that Airbus could rebuild it from scratch if needed
- Detailed description of Docker build command used to build the container
- Any other needed documentation (about licensing mechanism, for instance)

3.1 Way of delivery

Ideally, technical partners should use the GitLab platform, managed by LUH, to deliver their components: https://git.l3s.uni-hannover.de/

This GitLab platform could be used :

- As a code repository to push code, Dockerfile and documentation
- As a Docker registry to push Docker image

If it is not possible to use this GitLab platform, technical partners could also provide Airbus:

- A way to download a zip file containing all code, documentation and all other needed files
- An access to their own Docker registry so that Airbus could pull the docker image
- Alternatively, a way to download Docker image as a tar file, result of Docker save command

3.1 Best practices

The components of the ROXANNE platform are microservices running in a Kubernetes cluster and should be developed with this in mind.

- A container should be as light as possible:
  - Prefer alpine images as base images rather than big images with a lot of libraries installed
  - Use Docker external volumes for huge data (models) that could be shared between component instances
- Several instances of the same component could be started by the cluster to process huge amount of data. Each instance should support multi-threading and avoid conflicts with all other running instances (do not write in the same file at the same time, no lock on the same file at same time...)
- Each request is self sufficient. Service only needs JSON input to perform the request. No dependency to any previous result.
- Processing time should be reasonable

4. Audio Processing

The aim of the ROXANNE platform is to process different types of files among which audio files are probably the most important. The audio content may come from several sources, i.e. telephone conversations, broadcast news, video files.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

4.1 Audio Processing Chain

The audio processing chain in the ROXANNE platform extends from uploading of the input audio file to composition of the speaker similarity matrix. An overview of the audio processing chain is presented in Figure 2.

This is the first version of audio processing chain which will probably be updated in the future according to end users feedback and new needs.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

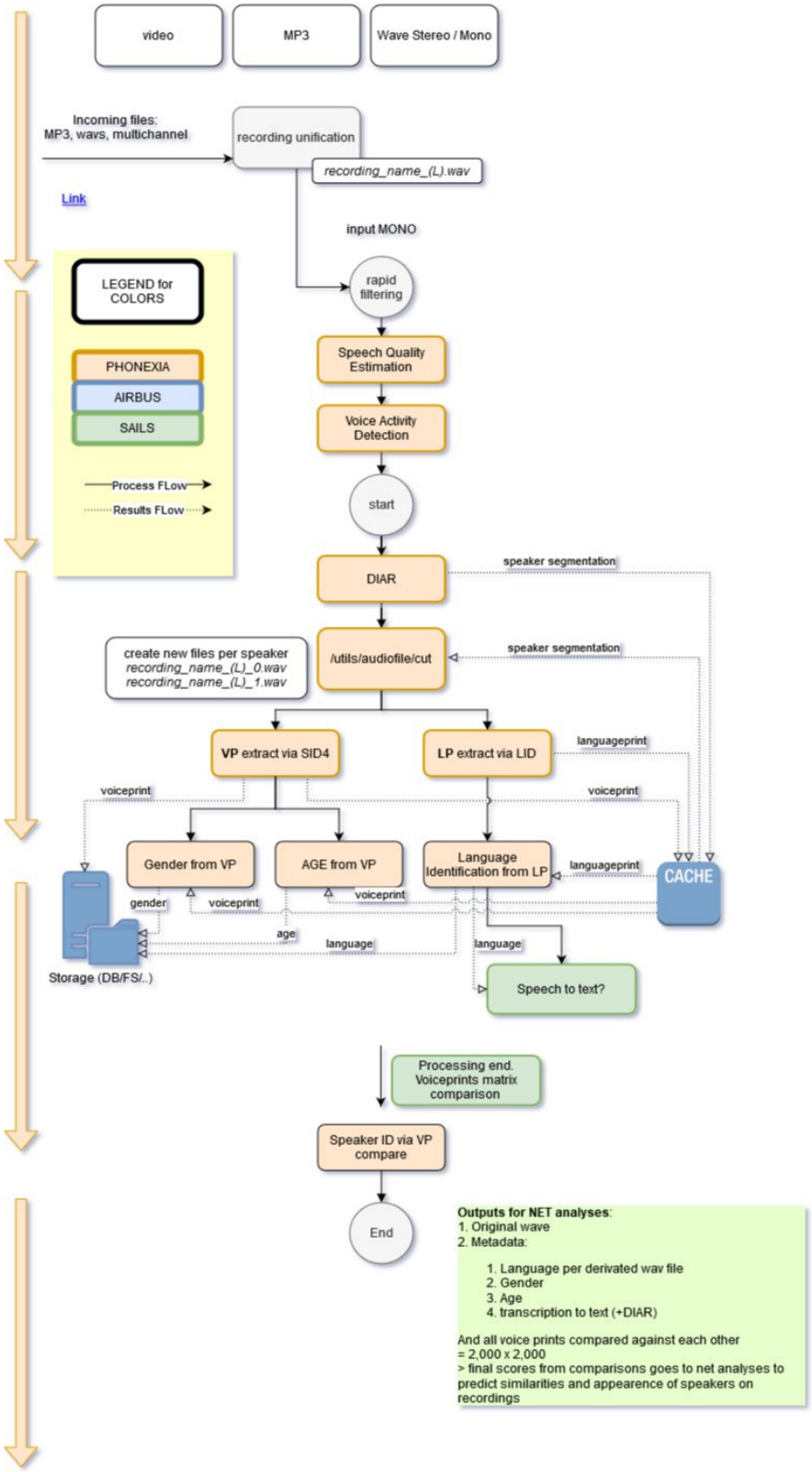


Figure 2 - First version of the audio processing chain

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

The audio data are processed with the following operations:

- They are first converted into wav files supported by all audio processing components,
- If needed, audio files containing 2 or more speakers are re-splitted into 1 sub-file per speaker,
- Waveforms are then generated for a later use in GUIs,
- Voiceprint and Languageprint are extracted,
- Gender and age of each speaker are determined from voiceprint,
- Spoken language is determined from languageprint,
- If identified language is supported, a component convert audio to text,
- Then text is processed according to text processing described in next section,
- At the end of the processing of a whole batch (directory), a voiceprint comparison component is called to build a comparison matrix between all available voiceprints,
- This allows us to identify all conversations in which a same speaker is involved, and also to build a relational network between speakers.
- This network is then passed to the network analysis processing chain.

4.2 Pre-processing

The input audio files must be pre-processed / normalized before they are processed by the audio processing components. Pre-processing operations may include splitting stereo streams into mono, mixing stereo streams into mono, converting the audio file format and resampling.

For the first version of the ROXANNE platform, which will be demonstrated in the first Field Test event (exact time and location not fixed as of the writing of this deliverable), three datasets are expected to be utilized: The CSI corpus, the NIST SRE data collection, and the ROXANNE simulated data. Please refer to D5.1 for detailed information on each of these datasets.

The data involved in these datasets lead to 3 distinct cases:

- Video files with stereo audio:
  - They are first converted to a 16kHz mono wav file
  - Speaker diarization is then applied to split into sub-files, one per speaker
- Stereo telephony audio:
  - They are split into two mono wav files, one for each (left-right) channel (==speaker)
- Mono telephony data:
  - Speaker diarization is then applied to split into one file per speaker

Processing chains are event driven. If all the files in a dataset are already split into a single sub-file per speaker, it could be processed by integrating it directly in the correct message queue.

4.3 Languageprint Extraction

This component takes an audio wav file as input and return the base64 languageprint extracted from audio.

This languageprint will then be used by language identification component.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

Input :

```
audio wave file path as HTTP parameter
```

Output :

```
  {
  "result": {
    "version": 1,
    "name": "LanguagePrintResult",
    "file": "/kelly_julia.wav",
    "model": "S",
    "time_range": {
      "from_time": 2.5,
      "to_time": 5.5
    },
    "languageprints": [
      {
        "languageprint": <Base64 Data>
      },
      {
        "languageprint": <Base64 Data>
      },
      ...
    ]
  }
}
```

4.4 Voiceprint Extraction

This component takes an audio wav file as input and returns the base64 voiceprint extracted from audio.

This voiceprint will then be used by age estimation, gender identification and voiceprints comparison components.

Input :

```
audio wave file path as HTTP parameter
```

Output :

```
  {
  "result": {
    "version": 3,
    "name": "SpeakerIdentification4VoiceprintResult",
    "file": "\/kelly_julia.wav",
    "model": "L4",
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

```
    "time_range": {
      "from_time": 2.5,
      "to_time": 5.5
    },
    "voiceprint_set": [
      {
        "voiceprint": <Base64 Data>,
        "speech_length": 12.7
      },
      {
        "voiceprint": <Base64 Data>,
        "speech_length": 24.8
      },
      ...
    ]
  }
}
```

4.5 Language Identification

This component takes a languageprint as input and returns the corresponding identified language and score.

Input :

```
{
  "languageprint": <Base64 Data>
}
```

Output :

```
  {
  "result": {
    "version": 1,
    "name": "LanguagePrintCompareToLanguagePackResult",
    "language_pack": "default",
    "model": "L",
    "scores": [
        {
          "name": "Afan_Oromo",
          "score": -8.212487
        },
        {
          "name": "sqi",
          "score": -7.554749
        },
        ...
      ]
  }
}
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

4.6 Gender Identification

This component takes a voiceprint as input and returns the corresponding identified gender and probability.

Input :

```
{
  "voiceprint": "<Base64 Data>"
}
```

Output :

```
  {
  "result": {
    "version": 3,
    "name": "GenderIdentificationResult",
    "file": "{voiceprint}",
    "model": "XL3",
    "channel_scores": [
      {
        "channel": 0,
        "scores": [
          {
            "name": "Female",
            "score": 0.9989388,
            "score_llr": 3.4236698
          },
          {
            "name": "Male",
            "score": 0.00106118135,
            "score_llr": -3.423641
          }
        ]
      }
    ]
  }
}
```

4.7 Age Estimation

This component takes a voiceprint as input and returns the corresponding estimated age.

Input :

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

```
{
  "voiceprint": "<Base64 Data>"
}
```

Output :

```
  {
  "result": {
    "version": 2,
    "name": "AgeEstimationResult",
    "file": "{voiceprint}",
    "model": "L",
    "channel_scores": [
      {
        "channel": 0,
        "scores": [
          {
            "name": "0",
            "score": 0
          },
          {
            "name": "1",
            "score": 0
          },
          ...
        ]
      }
    ]
  }
}
```

4.8 Speech to text

This component takes audio wave file and identified spoken language as inputs and returns text extracted from audio.

Input :

- Job Id: UUID
- path to audio/wav file
- [Optional] List of segments for pre-segmented audio[1], each containing
    - Start and end times (in milliseconds)
    - Desired (pre-detected) language in ISO639-3[2]

[1] This input parameter is not available as of the writing of this deliverable and will be implemented in a future version of the platform.

[2] The ISO639-3 statement here (and anywhere else in the document where a language identifier is concerned) is only a placeholder to represent a common (official) code table, not a final decision. The ISO639-3 code table does not differentiate between regional variations of some languages, including English (British/American/Indian, etc.). Therefore for practical purposes of ROXANNE, these language codes may be extended with a country code. Another idea in this respect is to include

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

- Number of desired N-best hypotheses (default 1, may not be supported by all languages)

Ideally, each segment would correspond to a speech section which is to be transcribed. If the segmentation information is not provided, the whole input file will be transcribed.

Output :

- Task id in UTF-8
- Type of the task: ASR (automatic speech recognition)
- A list of segments (if there is only one segment, it should span the whole file)
- Vendor-specific segment-dependent metadata, such as characteristics of audio, estimated noise level and content type

The following items pertain only to segments containing speech

- N-best hypotheses, each one containing
  - Recognized word, its start and end offsets in the source file (in milliseconds) and its confidence score in (0.00,1.00]
  - N-best score
  - Transcription: Text of the recognized speech (the first letter of each segment is capitalized)
- Global confidence score in (0.00,1.00]

Example :

Input :

```
{
  {
    "job_id": "ee76a60a-b32c-4151-810e-c21817e3d142",
    "path": "file:///home/doc/audio.wav",
    "segments": [
      {
        "start": "0",
        "end": "3010",
        "src_language": "eng",
        "nbest": "1"
      },
      {
        "start": "229720",
        "end": "235680",
        "src_language": "eng",
        "nbest": "1"
      }
    ]
  }
}
```

Output :

additional features such as "non-native vs native" through an accent detection component and the central speaker information repository, to allow for definitions like "native French speaker speaking English".

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

```
{
  "response": {
    "task_id": "26a3765b-ea06-4e7b-95b8-1352e7e0b69a",
    "task_type": "asr",
    "segments": [
      {
        "metadata": {
          "start": "0",
          "end": "3010",
          "non_speech_ratio": "0.1411",
          "type": "speech"
        },
        "nbests": [
          {
            "words": [
              {
                "text": "apple",
                "start": "0",
                "end": "1540",
                "confidence": "0.90"
              },
              {
                "text": "bananas",
                "start": "1541",
                "end": "3010",
                "confidence": "0.90"
              }
            ],
            "confidence": "1.00",
            "transcription": "Apple bananas."
          }
        ]
      },
      {
        "metadata": {
          "start": "229720",
          "end": "235680",
          "non_speech_ratio": "0.1035",
          "type": "speech"
        },
        "nbests": [
          {
            "words": [
              {
                "text": "one",
                "start": "229720",
                "end": "230543",
                "confidence": "1.00"
              },
              {
                "text": "two",
                "start": "230544",
                "end": "232863",
                "confidence": "1.00"
              },
              {
                "text": "three",
                "start": "232864",
                "end": "235680",
                "confidence": "1.00"
              }
            ],
            "confidence": "1.00",
            "transcription": "One two three."
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

```
            }
          ]
        }
      ],
      "global_confidence": "1.00"
    }
}
```

4.9 Voiceprint matrix comparison - Speaker Clustering

At the end of the processing of a whole dataset a voiceprint is bound to each speaker audio file.

Goal of this final task is to compare voiceprints so that we could cluster speakers and try to find all conversations in which a specific speaker is involved.

Result of this task is composed of:

- the whole list of all speaker files in the dataset (1 wav file = 1 speaker = 1 voiceprint)
- for each speaker file, age, gender and language identified
- for each speaker file, corresponding transcription (if available)
- for each speaker file, named entities found in transcription (if available)

A score matrix comparing each speaker file with all others (N speaker files -> a NxN matrix)

Interface:

This service is available using following REST interface:

POST request on /vpcompare path

Input :

- jobId: Unique identifier of the job / dataset

Output :

- jobId: Unique identifier of the job / dataset
- taskId: UUID of the task
- conversations : a list of “conversations” involving 2 or more people. 1 original file (stereo) correspond to a conversation. For most of the datasets there will only be 2 peoples involved in conversation, i.e. 2 sub mono wav files
    - channels : list of peoples involved in the conversation (most of time only 2)
        - id: channel file unique identifier
        - gender: gender identified for this channel (Male, Female, Unknown)
        - genderConfidence: float. Confidence for this gender identification.
        - age: age estimated for this channel
        - ageConfidence: float. Confidence for this estimated age.
        - language: identified spoken language for this channel.
        - languageConfidence: float. Confidence for this language identification.
        - transcription: text. Full text of transcripted audio from this channel.
        - entities: list of entities extracted from transcription of this channel
        - topics: topics extracted from the transcripton of this channel
        - timestamp: timestamps of the conversation

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 833635.

- voiceprintsMatrix: The scores for all voiceprints comparison. Index of speakers in this matrix are those used in speakers list element below. (first speaker of the list is speaker_0 in the matrix...)

Input :

```
{
    "jobId":"csi"
}
```

Output :

```
  {
    "jobId": "csi",
     "taskId": "30379fcd-273d-47a9-ad46-278f56a55b62",
     "conversations": [
       {
         "channels": [
            {
              "id": "s01e20_0",
              "ageConfidence" : 1,
              "age" : 44,
              "genderConfidence" : 0.88504344,
              "gender" : "Male",
              "languageConfidence" : -1.2463263,
              "language" : "English_British"
              "transcription": "Very distinctive popped up quite. Blue light on the red
sea pale varied pseudomonas aeruginosa which is a bacteria occasionally found in the
bloodstream soon executive."
              "entities":{"PER":["pseudomonas aeruginosa"], "LOC":[], "ORG" :[]},
              "topics":["SCIENCE", "LIGHT"]

            },
            {
              "id": "s01e20_1",
              "ageConfidence" : 1,
              "age" : 38,
              "genderConfidence" : 0.874985,
              "gender" : "Female",
              "languageConfidence" : -1.548933,
              "language" : "English_British"
              "transcription": "Very distinctive popped up quite. Blue light on the red
sea pale varied pseudomonas aeruginosa which is a bacteria occasionally found in the
bloodstream soon executive."
              "entities":{"PER":["pseudomonas aeruginosa"], "LOC":[], "ORG" :[]},
              "topics":["SCIENCE", "LIGHT"]
            },
            ...
         ]
       ,
       ...
       ],
   "voiceprintsMatrix": [
       [
         <s01e20_0 x s01e20_0 score>,
         <s01e20_0 x s01e20_1 score>,
         ...,
         <s01e20_0 x sNeM_1 score>
       ],
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

```
        [
          <s01e20_1 x s01e20_0 score>,
          <s01e20_1 x s01e20_1 score>,
          ...,
          <s01e20_1 x sNeM_1 score>
        ],
        ...,
        [
          <sNeM_1 x s01e20_0 score>,
          <sNeM_1 x s01e20_1 score>,
          ...,
          <sNeM_1 x sNeM_1 score>
        ]
      ]
}
```

5. Text Processing

While audio files are processed and text is extracted from them more information could be retrieved from this text. The ROXANNE platform could also process text coming from social media or any kind of text documents.

5.1 Text Processing Chain

In the ROXANNE platform, text data are processed in order to retrieve interesting information from them.

- They are first normalized in order to extract text from various input formats (html, pdf, txt...),
- Language identification is applied on text,
- Named Entities are then extracted from text,
- Text is classified according to topic it deals with,
- Finally, sentiment analysis is applied to text.

These components are applied to both text coming from text files and text extracted from audios/videos using Speech to text component.

5.2 Named Entity Recognition and Relation Extraction

This service allows to automatically recognize named entities (person name, location etc.) in the text. After named entities are detected, we further look for relations between these entities (e.g. <Obama, born in, Hawaii>). Both recognized entities and relations will be returned to the end-users.

Input :

• job_id: UUID

• src_language: source language ISO 639-3

• inf_setting_name: when we do inference, i.e. detect entities, we support different settings for it. For example, we want to specify which model we use. we use the default setting if this input is not provided.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

• target entities: list containing entities the user want to detect. Default will be a list containing all entities the model can support.

• input_text: text on which we detect entities. text should be in UTF-8 format.

Output :

• task_id: UUID

• task_type : NER (Named Entity Recognition)

• entities list: a list containing all entities we detect. Each element in the list has the following format.

◦ tokens: entity tokens

◦ uuid: UUID for the detected entity

◦ type: entity type

◦ confidence: the confidence of detected entity. The higher the confidence, the higher the probability that our detection is true positive.

◦ segments: a tuple contains a start and an end position of the detected entity.

▪ startoffset: start position of the detected entity in the text

▪ endoffset: end position of the detected entity in the text

• Relations

◦ relation uuid: UUID for the detected relations

◦ subject entity uuid: a relation is a triple (subject, relation, object) where the subject and the object are detected entities. This is the UUID for the subject entity.

◦ object entity uuid This is the UUID for the object entity.

◦ relation type

◦ confidence: the confidence of the detected relation. The higher the confidence, the higher the probability that our detection is true positive.

◦ segments: Positions of the detected subject and object entities in the text.

▪ subject_startoffset,

▪ subject_endoffset,

▪ object _startoffset,

▪ object _endoffset,

Example :

Input :

```
[
  {
    "job_id":"ee76a60a-b32c-4151-810e-c21817e3d142",
    "src_language":"en ",
    "inf_setting_name":"base ",
    "input_text ":"Fred and Mary got married, ..."
  }
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

```
]
```

Output :


```
[
  {
    "task_id": "30379fcd-273d-47a9-ad37-278f56a55b62",
    "task_type": "ner",
    "entities": [
      {
        "token": "Fred",
        "uuid": "40fd5f77-5564-44e1-9aee-4d0dd3723ee1",
        "type": "PERSON ",
        "confidence ": 0.9,
        "segments": [
          {
            "start": 1,
            "end": 1,
          }
        ]
      },
      {
        "label": "Mary ",
        "uuid": "40fd5f77-5564-44e1-9aee-4d0dd3723ee2",
        "type": "PERSON",
        "confidence ": 0.9,
        "segments": [
          {
            "start": 3,
            "end": 3,
          }
        ]
      }
    ],
    "relations": [
      {
        "uuid_rel": "40fd5f77-5564-44e1-9aee-4d0dd3723ee3",
        "uuid_sub": "40fd5f77-5564-44e1-9aee-4d0dd3723ee1",
        "uuid_obj ": "40fd5f77-5564-44e1-9aee-4d0dd3723ee2",
        "type": "MARRIED",
        "confidence ": 0.9,
        "segments": [
          {
            "sub_start": 1,
            "sub_end": 1,
            "obj_start ": 3
            "obj_end ": 3
          }
        ]
      }
    ]
  }
]
```


5.3 Topic Detection

This component allow to classify text according to topic they are dealing with. The possible topics for the CSI dataset are “Detective”, “Investigator”, “Forensic Evidence”, “Suspicion”, “Murder Investigation”, “Crime Scene”, “Crime Environment”, “Crime Lab”, “Suspect”, and “Investigation”.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 833635.

Input :

- job_id: UUID
- Text content in UTF-8
- Language

Output :

- task_id: UUID
- topic

Example :

Input :

```
[
  {
    "job_id":"ee76a60a-b32c-4151-810e-c21817e3d142",
    "content":"i got the blood samples on way to lab you mean the blood swirls
next to father'\''s body in boys room i studied pictures of the manson murders
this is n'\''t butter it'\''s imitation what'\''s your take explains why the blood
is confined to bed and floor under it gave his life for the little girl there
should be more blood the first suspect well she'\''d need help maybe a boyfriend",
     "language": "en"
  }
]
```

Output :

```
[
  {
    "task_id": "30379fcd-273d-47a9-ad37-278f56a55b62",
    "topic ": "Scene Of The Crime"
 }
]
```

6. Video Processing

The ROXANNE platform also process video data to extract information from them.

6.1 Video Processing Chain

In Roxanne platform, video data are processed this way:

- They are first normalized in order to extract audio from video,
- Audio is then processed following audio processing chain described in section 4
- Place diarization is then applied on video to identify videos containing same place,
- Video to text component is applied to associate videos with some keywords about objects contained in videos.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

6.2 Place diarization

The objective of the place diarization service is (1) to identify video segments related to the same place and (2) linking videos observing the same place, through image content indexing and similarity evaluation.

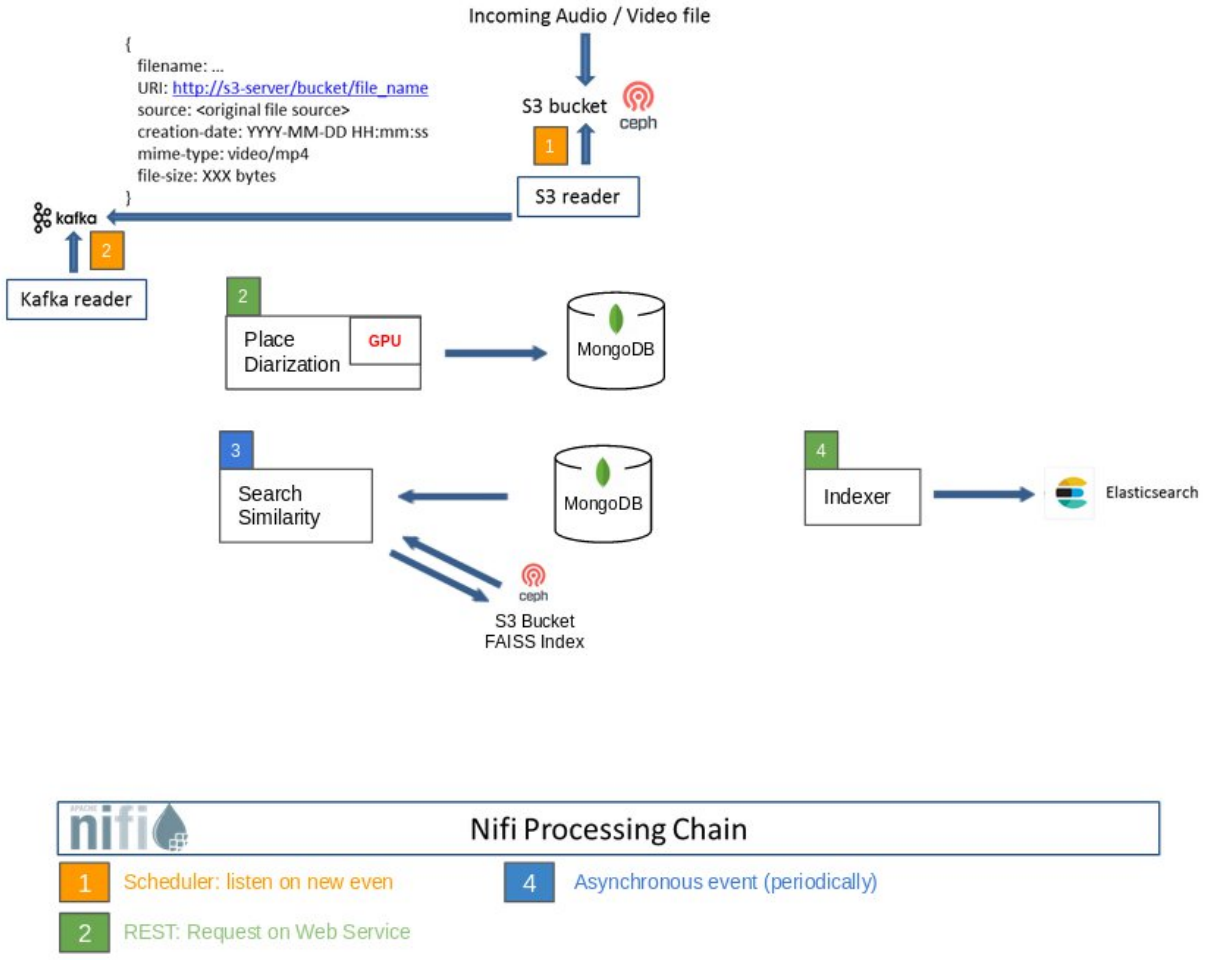


Figure 3 - First version of the place diarization chain

The Place Diarization architecture includes:

- 3 separate modules:
- 2 event listeners : S3 reader and Kafka reader
- 1 kafka topic

Kafka topic (stream of records): A topic is a category or feed name to which records are published, similar to a message queue. Topic in Kafka is always multi-subscriber; that is, a topic can have zero, one or many consumers that subscribe to the data written to it.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

In this case:

- The publisher is the S3 Reader, which publishes messages containing the path of the video uploaded to s3 and metadata. Each published message corresponds to a video uploaded to the s3 bucket. S3 Reader is executed each time a new object (video) is uploaded to the s3 bucket.
- The subscriber is the Kafka Reader (consumer), which has the role of subscribing to a Kafka topic and retrieving new messages. This module must periodically (either after a defined time or after receiving a defined number of messages) call another module via a Rest API. This module receives as input the messages from the Kafka topic and returns these messages to a REST API.

Three modules must run in a Docker:

- Place diarization: This module requires a GPU. It receives as input the messages coming from the Kafka reader (consumer) and analyzes, extracts signatures, performs a place diarization (i.e identification of temporal segments in a given video related to the same location) of each video and pushes the results (as JSON objects) to MongoDB.
- Similarity Search: A periodic task. This module retrieves new records in MongoDB, synchronizes the FAISS index (index stored on a bucket s3) and run similarity search between the new records and the whole database. This module returns results (location matching between several videos) in a JSON object.
- Indexer: Takes as input a JSON from the Similarity Search module and inserts it into an ElasticSearch database for further visualization / query purposes.

The "Place Diarization Service" must run in a docker with a GPU (rtx 2080 ti for example).

Example document mongoDB:

- Job_id : uui
- status instance : new / processed
- Video_id : uui
- task_id : uui
- list of places :
- id_place
- segments:
- start
- end
- signatures

Input :

- job_id : UUI
- video_path : path s3 file

Output :

- Job_id : uui
- Video_id : uui
- task_id : uui

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

- list of places :
- id_place
  - confidence
  - segments:
    - start
    - end
  - signatures

Example :

Input :

```
[
{
"job_id": "ee76a60a-b32c-4151-810e-c21817e3d142",
"video_path": "s3://bucket/video/1365156.mp4",
}
]
```

Output :

```
[
{
"job_id": "ee76a60a-b32c-4151-810e-c21817e3d142",
"task_id": "fds1ds56f-sdfg-sd5f-z8e9-qs561dsq6fs",
"video_id": "79071b23-83f1-4e61-92cb-f2c41750c538",
"places": [
{
"place_id": "97853951-7041-410f-bf3e-89739ad1a2b1",
 "confidence": "0.86"
"segments": [
{
"start": 0,
"end": 3550
},
{
"start": 9826,
"end": 12026
}
],
"signatures" : [{"1582238376":[8.85133922e-01, 1.08143437e+00, 1.20291400e+00,
-1.15219557e+00, -4.11096662e-01, 1.15465784e+00,
3.78168017e-01, 8.45648706e-01]},]
},
{
"place_id": "f2d3fd4f-dd2c-47ce-814c-e6a4cb071ea5",
 "confidence": "0.98"
"segments": [
{
"start": 26532,
"end": 28210
},
{
"start": 89513,
"end": 92513
}
],
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

```
"signatures" : [{"15825317489":[9.5616e-01, 1.54654654e+00, 3.456540e+00,
-1.15219557e+00, -3.11096662e-01, 1.15465784e+00,
3.78168017e-01, 1.45648706e-01]},]
}
]
}
]
```


Search Similarity Service

Input:

- Job Id : UUI

Output:

- job_id
- task_id
- nb matching
- list of matching (relation):
  - id matching
  - confidence
  - list of matches:
    - id video
    - path to video
    - id place

Example :

Input :


```
[
{
"job_id": "ee76a60a-b32c-4151-810e-c21817e3d142",
}
]
```


Output :


```
[
   {
      "job_id": "ee76a60a-b32c-4151-810e-c21817e3d142",
      "task_id": "bf0a1ee4-d88f-436b-b4a1-149adb7a9690",
      "nb_matching": 1,
      "matching": [
         "match_id": "8a171b24-407c-411b-b3a0-47c7bc758ad0"
         "confidence": "0.97"
         "matches": [
         {
            "video_id": "79071b23-83f1-4e61-92cb-f2c41750c538",
            "path_to_video": "s3://bucket/video/1.mp4",
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

```
            "place_id": "97853951-7041-410f-bf3e-89739ad1a2b1",
         },
         {
            "video_id": "8f5b30b7-486d-4ab3-b217-5246b2ce5d34",
            "path_to_video": "s3://bucket/video/4.mp4",
            "place_id": "dec2cc2a-67f5-45ae-a396-3668b647e55c",
         }
         ]
      ]
   }
]
```


6.4 Video to text service

The objective of the Video2Text is to associate video segments with keywords coming from object detectors or semantic segmentation classes.

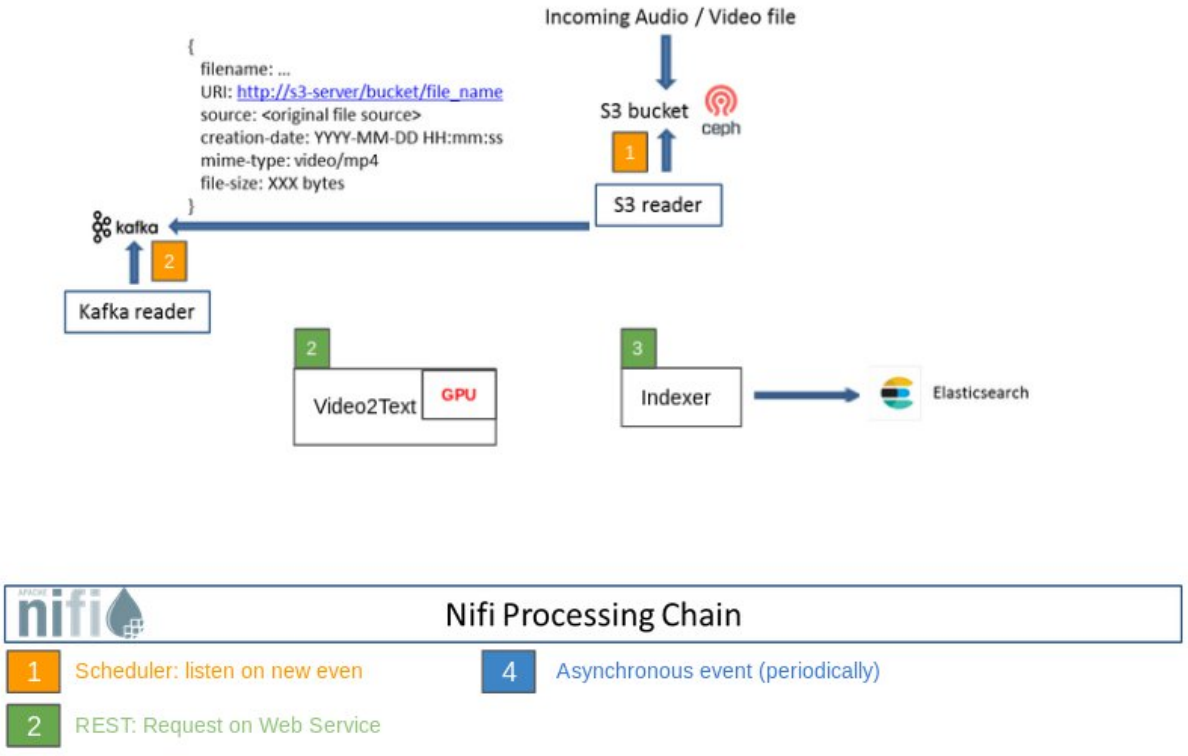


Figure 4 - Video2Text service description

Input:

- Job Id : UUI

Output:

- job_id
- task_id
- list of segments :
  - id segment

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

- start
- end
- list of keywords:
  - id object
  - label
  - confidence

Example :

Input :

```
[ {
    "job_id": "ee76a60a-b32c-4151-810e-c21817e3d142",
 } ]
```

Output :

```
[
   {
      "job_id": "ee76a60a-b32c-4151-810e-c21817e3d142",
      "task_id": "bf0a1ee4-d88f-436b-b4a1-149adb7a9690",
      "segments ": [
         {
             "id_segment" : "sd89f465-iuo456 -915 -qspod651 -7uytjh132ga",
             "start" : 0,
             "end": 162,
             "keywords" : [
                    {
                        "id_class" : 1,
                        "label": "person",
                       "confidence" : 0.95
                    },
                   {
                        "id_class" : 3,
                        "label": "building ",
                       "confidence" : 0.99
                    },
                     {
                        "id_class" : 9,
                        "label": "car ",
                       "confidence" : 0.86
                    }
              ]
         }
      ]
   }
]
```

7. Network analysis

In ROXANNE, the network construction and analysis components will be developed in conjunction with the speech/text/video technologies. Precisely, network components will take as input the output from

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

speech, text, and video processing chains components, which are JSON objects describes in previous sections, then construct the networks and perform the required analysis. In the following subsections, we describe in detail the functionalities provided by these components at this phase of the project. Methods for developing these functionalities are extensively documented in deliverable D6.1 *Preliminary report on network analysis*.

7.1 Network construction

This component allows to construct networks from interactions of individuals that are extracted by speech, text, and video processing chains. The interactions are, for example, phone calls, message exchanges, co-appearance in some video scenes, etc.

Input :

In general, this component take as input a JSON object that contain the following information

- task_id: UUID
- Relations/ Interactions
    - source individual
        - individual id
        - individual properties
            - type, place, age, gender, spoken languages, etc.
            - recognition confidence
    - destination individual
        - individualid
        - individual properties
            - place, age, gender, spoken languages, etc.
            - recognition confidence
    - relation/interaction id
    - relation/interaction properties
        - type, associated content, etc.
        - recognition confidence

Particularly, for the first field test, we also develop another preprocessing toolbox for this component so as it also take as input the JSON object that is the output of speech, text, and video processing chain and is described in Section 4.9

Output :

- task_id: UUID
- nework: list of JSON object, each describes an individual or a link/edge in the constructed network
    - The individual object has the following fields
        - type: “node”
        - id: id of the individual
        - properties: JSON object describing properties of the individual, e.g., gender, language, etc.
    - The edge object has the following fields
        - type: “edge”
        - source: id of an individual who is the source of the edge
        - target : id of an individual who is the target of the edge
        - properties: JSON object describing properties of the edges, e.g, type, weight, etc.

Example :

Input :

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 833635.

```
{
    "taks_Id": "csi",
     "taskId": "30379fcd-273d-47a9-ad46-278f56a55b62",
     "conversations": [
       {
         "channels": [
            {
              "id": "s01e20_0",
              "ageConfidence" : 1,
              "age" : 44,
              "genderConfidence" : 0.88504344,
              "gender" : "Male",
              "languageConfidence" : -1.2463263,
              "language" : "English_British"
              "transcription": "Very distinctive popped up quite. Blue light on the red
sea pale varied pseudomonas aeruginosa which is a bacteria occasionally found in the
bloodstream soon executive."
              "entities":{"PER":["pseudomonas aeruginosa"], "LOC":[], "ORG" :[]}
              "topics ":{["SCIENCE", "LIGHT"]}

            },
            {
              "id": "s01e20_1",
              "ageConfidence" : 1,
              "age" : 38,
              "genderConfidence" : 0.874985,
              "gender" : "Female",
              "languageConfidence" : -1.548933,
              "language" : "English_British"
              "transcription": "Very distinctive popped up quite. Blue light on the red
sea pale varied pseudomonas aeruginosa which is a bacteria occasionally found in the
bloodstream soon executive."
              "entities":{"PER":["pseudomonas aeruginosa"], "LOC":[], "ORG" :[]}
              "topics ":{["SCIENCE", "LIGHT"]}
            },
            ...
         ]
       ,
       ...
       ],
   "voiceprintsMatrix": [
       [
         <s01e20_0 x s01e20_0 0.7 >,
         <s01e20_0 x s01e20_1 0.1>,
         ...,
         <s01e20_0 x sNeM_1 score>
       ],
       [
         <s01e20_1 x s01e20_0 0.4>,
         <s01e20_1 x s01e20_1 0.7>,
         ...,
         <s01e20_1 x sNeM_1 0.3>
       ],
       ...,
       [
         <sNeM_1 x s01e20_0 0.7>,
         <sNeM_1 x s01e20_1 0.125>,
         ...,
         <sNeM_1 x sNeM_1 0.9>
       ]
     ]
}
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

Output :

```
{
       "task_id": "csi",
       "task": "social_network_analysis"
       "network":
                [
               {"type": "node", "id": "speaker1", "properties": {"type": "person", "age ":
"44","gender": "male"}}
           ...
           type":"edge", "source": "speaker2", "target": "speaker3", "properties": {"type":
"call","observed": true,"weight": 1}}
           ...
                ]

}
```

7.2 Social influence analysis

Description: This component allows to identify most important individuals in the networks constructed in the previous sections by computing for each entity one or more importance scores

Input :

- task_id: UUID
- task: "social_network_analysis"
- network: list of JSON objects as described in the output of network construction in Section 7.1
- Options: options for the analysis
  - method: method for performing the analysis
  - parameters: parameters for the method

Output :

- job_id: UUID
- result: result of the analysis, including
  - success: 1 if the analysis is performed successfully, 0 otherwise
  - message: string, indicates information about the analysis
  - scores: list of JSON objects if the analysis is successful, or None otherwise. the JSON objects have the following fields
    - id: id of the individual
    - score: score of the individual

Example :

Input :

```
{
       "task_id": "sia_xxxx",
       "task": "social_network_analysis"
       "network":
                [
               {"type": "node", "id": "Satam_Suqami", "properties": {"type": "person",
"name": "Satam Suqami"}}
           ...
           type":"edge", "source": "Samir_Kishk", "target": "Essid_Sami_Ben_Khemail",
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

```
"properties": {"type": "other_associate","observed": true,"weight": 1}}
          ...
                ]
       "options": {"method": "authority", "parameters": {}}

}
```

Output :

```
{
       "task_id": "sia_xxxx",
       "result":
       {
              "success": 1,
              "message": "The analysis was performed successfully"
              "scores":[{"id":       "Satam_Suqami",       "score":       0.3},       {"id":
"Essid_Sami_Ben_Khemail", "score": 0.2},
           {"id": "Samir_Kishk", "score": 0.1},...]
       }
}
```

7.3 Community detection

Description: This component allows to identify cohesive groups of individuals in the input network

Input :

- task_id: UUID
- task: "community_detection"
- network: list of JSON objects as described in the output of network construction in Section 7.1
- Options: options for the analysis
  - method: method for performing the analysis
  - parameters: parameters for the method

Output :

- task_id: UUID
- result: result of the analysis, including
  - success: 1 if the analysis is performed successfully, 0 otherwise
  - message: string, indicates information about the analysis
  - Communities
    - id: community id
    - members: community members
      - id: individual id
      - score: membership score

Example :

Input :

```
{
       "task_id": "sia_xxxx",
       "task": "community_detection"
       "network":
                [
               {"type": "node", "id": "Satam_Suqami", "properties": {"type": "person",
"name": "Satam Suqami"}}
          ...
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

```
          type":"edge", "source": "Samir_Kishk", "target": "Essid_Sami_Ben_Khemail",
"properties": {"type": "other_associate","observed": true,"weight": 1}}
          ...
                ]
       "options": {"method": "hierarchical", "parameters": {'K':5}}
}
```

Output :

```
{
       "task_id": "scd_xxxx",
       "result":
       {
              "success": 1,
              "message": "The analysis was performed successfully",
              "communities":
              [
                     {"id": "comm_1", "members": [{"id": "Satam_Suqami", "score": 1.0},{"id":
"Essid_Sami_Ben_Khemail", "score": 0.5}]},
                     {"id":  "comm_2", "members": [{"id": Satam_Suqami", "score": 0.7},{"id":
"Satam Suqami", "score": 1.0}]}
              ]
       }
}
```

7.4 Link prediction

Description: This component allows to predict, for the input individuals, the potential latent (or hidden) links to other entities in the same network.

Input :

- job_id: UUID
- task: "link_prediction"
- network: list of JSON objects as described in the output of network construction in Section 7.1
- Options: options for the analysis
  - method: method for performing the analysis
  - parameters: parameters for the method
    - sources: list of individual ids to predict link for

Output :

- task_id: UUID
- result: result of the analysis, including
  - success: 1 if the analysis is performed successfully, 0 otherwise
  - message: string, indicates information about the analysis
  - predictions: Predicted links for source individuals
    - individual
    - predicted destination individuals
      - individual id

Example :

Input :

```
{
       "job_id": "slp_xxxx",
       "task": "link_prediction"
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

```
        "network":
                  [
                {"type": "node", "id": "Satam_Suqami", "properties": {"type": "person",
"name": "Satam Suqami"}}
            ...
            type":"edge", "source": "Samir_Kishk", "target": "Essid_Sami_Ben_Khemail",
"properties": {"type": "other_associate","observed": true,"weight": 1}}
            ...
                  ]
        "options": {"method": "jaccard_coefficient", "parameters": {"sources":["Satam
Suqami"]}}
}
```

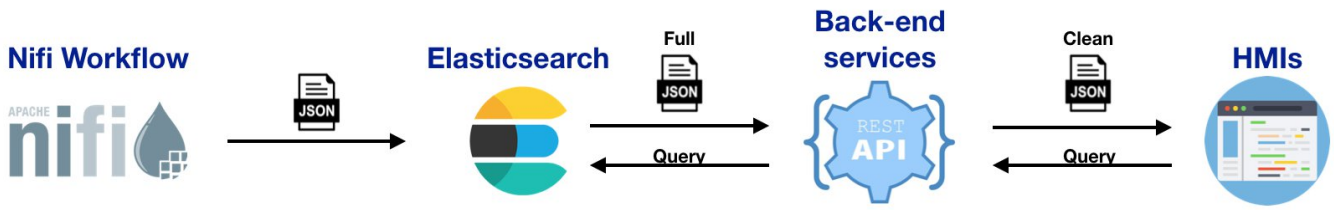
Output :

```
{
        "task_id": "slp_xxxx",
        "result":
        {
                "success": 1,
                "message": "The analysis was performed successfully",
                "predtictions":
                [
                        "Satam Suqami": ["Essid_Sami_Ben_Khemail"]
                ]
}
```

8. HMIs Backend Service

8.1 HMIs Backend service architecture

The HMIs back-end service will be in front of the Elasticsearch index, HMIs services will be able to request the REST API to retrieve data.



8.2 HMIs Backend service interfaces

The back-end REST API will have different endpoints, one for each kind of data processed by the processing work flow. The service responses can be paginated, to have the next results you will need to give in argument where to start.

HMIs will be able to run different queries to access the data.

Input:

- coreJobId : unique id referencing the job bound to the document (==mission).
- start_date : range start date to retrieve data.
- end_date : range end date to retrieve data.
- from : cursor to get the next results of heavy request (0 by default)
- limit: limit number of document returned (max 10 000, 1000 by default)
- relative_time: set a range of result from now.

Examples:

- Using the “coreJobId”:

GET uri/api/version/service_name -d {“coreJobId”: “30379fcd-273d-47a9-ad37-278f56a55b62”, “from” : 0,

“limit”: 1000}

- Using time range :

GET uri/api/version/service_name -d {“start_date”: 27/03/2020 12:00 , “end_date”: 28/03/2020 12:00, “from”: 0, “limit”: 1000}

- Using relative time :

GET uri/api/version/service_name -d {“relative_time”: { “day”: 0, “hours”: 24, “minutes”: 0}, “from”: 0, “limit”: 1000}

Output:

The back-end service will send formatted JSON to HMI, each data will be concatenated into a list of JSON object.

Common response attributes:

- “doc_count” : number of documents return for this query.
- “total”: total number of documents found matching the query.
- “from”: starting position for the results documents .
- “response”: JSON object containing processing chains results.

Response example :

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 833635.

```
{"doc_count": 1,
"total": 10,
"cursor": 1,
"response":{
      "coreCreateDate":  "yyyy-MM-dd'T'HH:mm:ss.SSS'Z",
      "coreJobId": "30379fcd-273d-47a9-ad37-278f56a55b62",
      "file_name": "audio.wav",
      "coreDataURI": "file:///home/doc/audio.wav",
      "format": "audio/x-wav",
      "fileSize": 9410879,
      "speakers": [{
            "id": 1,
            "channelId": 23421,
            "label": "spk1",
            "gender": "male",
            "genderConfidence": 0.9,
            "age": "30",
            "ageConfidence": 0.7,
            "language":"eng",
            "languageConfidence": 0.9,
            "transcriptionConfidence": 0.6,
            "duration": 354643213,
            "wordNumber": 463
            },
            {"id": 2,
            "channelId": ,
            "label": "",
            "gender": "female",
            "genderConfidence": 0.9,
            "age": "40",
            "ageConfidence": 0.7,
            "language":"eng",
            "languageConfidence": 0.9,
            "transcriptionConfidence": 0.6,
            "duration": 3543213,
            "wordNumber": 356
            }],
      "recognitions": {
            "channelId": 3,
            "transcription": {
                  "analyzer": "text_analyzer",
                  "copyTo": "plainText"
            },
            "globalConfidence": 0.8,
            "segments": [{
                  "metadata":{
                        "non-speech-ratio": 0.1539,
                        "type": "speech"
                  },
                  "nbest": [
                        {"words": [
                              {"text": "awesome",
                              "start": 9410,
                              "end": 20545,
                              "confidence": 0.89
                              },
                              {"text": "speech",
                              "start": 20900,
                              "end": 32380,
                              "confidence": 0.95
                              }],
                        "score": 0.9
```

```
                        }
                ]
        }]
    },
    "plainText": "text",
    "speechSegments":{
        "sourceStart": 3453,
        "sourceEnd": 5432,
        "targetStart": 3443,
        "targetEnd": 5532,
        "speaker": 1,
        "language": "eng",
        "languageConfidence": 0.7,
        "wordNumber": 567
        },
    "transcription": "text",
    "global_confidence": 0.9
    },
    "spoken_languages": {
        "language": "eng",
        "global_confidence": 0.8 ,
        "segments": [
                {"start": 9410 ,
                "end": 32380 ,
                "language" : "eng",
                "confidence": 0.8
                }
        ]
    },
    "translations": {
        "executionId": " " ,
        "taskType": " " ,
        "translation": " " ,
        "traLanguage": " " ,
        "globalConfidence": " " ,
        "alignments": {
                "sourceStart": " " ,
                "sourceEnd": " " ,
                "targetStart": " " ,
                "targetEnd":
        }
    },
    "entities": {
        "persons": [{
                "uuid": "40fd5f77-5564-44e1-9aee-4d0dd3723ee2" ,
                "label": "Bush" ,
                "candidate": "false" ,
                "coordinates": " ",
                "segments": [ {
                        "refersTo": {
                                "uuid": " " ,
                                "taskType":
                        },
                        "start": 17 ,
                        "end": 21 ,
                        "confidence": 0.7
                }]
        }]
    },
    "places": {
        "uuid": "40fd5f77-5564-44e1-9aee-4d0dd3723ee1",
        "label": "Washington",
        "candidate": "false",
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

```
        "coordinates":  "(0.0, 0.0)",
        "segments": [{
                "refersTo": {
                        "uuid": " " ,
                        "taskType": " "
                },
                "start": 2 ,
                "end": 12,
                "confidence": 0.6
        }]
},
"organisations": {
        "uuid": "40fd5f77-5564-44e1-9aee-4d0dd37232d",
        "label": "org",
        "candidate": "false" ,
        "coordinates": "(0.0, 0.0)",
        "segments": [{
                "refersTo": {
                "uuid": " ",
                "taskType": " "
                },
                "start": 4,
                "end": 14,
                "confidence": 0.5
        }]
},
"events": {
        "uuid": "40fd5f77-5564-44e1-9aee-4d0dd37256g" ,
        "label": "evt" ,
        "candidate": "false" ,
        "coordinates": "(0.0, 0.0)",
        "segments": [{
                "refersTo": {
                        "uuid": " ",
                        "taskType": " "
                },
                "start": 5 ,
                "end": 8,
                "confidence": 0.7
        }]
},
"equipments": {
        "uuid": "40fd5f77-5564-44e1-9aee-4d0dez7256e" ,
        "label": "equip",
        "candidate": "false",
        "coordinates": "(0.0, 0.0)",
        "segments": [{
                "refersTo": {
                        "uuid": " ",
                        "taskType": " "
                        },
                "start": 7 ,
                "end": 34,
                "confidence": 0.9
        }]
},
"relations": [{
        "uuid_rel": "40fd5f77-5564-44e1-9aee-4d0dd3723ee3",
        "uuid_src": "40fd5f77-5564-44e1-9aee-4d0dd3723ee1",
        "uuid_tgt": "40fd5f77-5564-44e1-9aee-4d0dd3723ee2",
        "type": "PERSON_LOCATION",
        "segments": [{
                "start": "17",
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

```
                        "end": "35",
                        "confidence": "0.3"
                }]
        }],
        "processing": {
                "ner": {
                        "startDate": "yyyy-MM-dd HH:mm:ss.SSS",
                        "endDate": "yyyy-MM-dd HH:mm:ss.SSS",
                        "duration": 24224,
                        "status": "Done"
                },
                "audio-extraction": {
                        "startDate": "yyyy-MM-dd HH:mm:ss.SSS",
                        "endDate": "yyyy-MM-dd HH:mm:ss.SSS",
                        "duration": 24224,
                        "status": "Done"
                },
                "waveform": {
                        "startDate": "yyyy-MM-dd HH:mm:ss.SSS",
                        "endDate": "yyyy-MM-dd HH:mm:ss.SSS",
                        "duration": 24224,
                        "status": "Done"
                },
                "part": {
                        "startDate": "yyyy-MM-dd HH:mm:ss.SSS",
                        "endDate": "yyyy-MM-dd HH:mm:ss.SSS",
                        "duration": 24224,
                        "status": "Done"
                },
                "lid": {
                        "startDate": "yyyy-MM-dd HH:mm:ss.SSS",
                        "endDate": "yyyy-MM-dd HH:mm:ss.SSS",
                        "duration": 24224,
                        "status": "Done"
                },
                "trans": {
                        "startDate": "yyyy-MM-dd HH:mm:ss.SSS",
                        "endDate": "yyyy-MM-dd HH:mm:ss.SSS",
                        "duration": 24224,
                        "status": "Done"
                }
        }
        }
}
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.