

COAL Metadata Extraction for Audio

podcast.mp3 (binary)

...öÖç,éä<lä yWRQ@«gÂ][2 ~S !!- T FÍ Y+
,Vò*ΔHGΩØI + „I sH,‡Um9_aeg
‡µ'÷RÁ^áß÷€^Vg U% 3^FU Ó Δ
VÉU'¢?O.öùmú¶r7âi á-Âv gôùMñÂë≤kK©
YÉSÜΔ&«íó¥ådxYfæmj3Jç ÁçåaBUlðfÙ-
ð\$g_«Åns^t- á Aæøràß÷ùΔ„% ≤¥e 'Fó÷ð
h-‡ Gç"@O»l'CEí^t%ojc'ÓΔ Ø"flö,¡Ω
°Ó;ÅÖ-¶lx]ÆìÜØ"‰o¿viiΔ",X®AZ5£A...
je-œ\$ñåÖç,éä<lä yWRQ@«gÂ][2 ~S !!- T
FÍ Y+ ,Vò*ΔHGΩØI + „I sH,‡Um9_aeg...
....‡µ'÷RÁ^áß÷€^Vg U% 3^FU Ó Δ
VÉU'¢?O.öùmú¶r7âi á-Âv gôùMñÂë≤kK©

Audio-Dateien sind nicht maschinenverstehbar. COAL ändert das.

(COAL = "Web Media Content Analysis Framework")

- curl -H "accept:text/turtle" "http://example.org/coal/resource?url=http://example.com/testpodcast.mp3"
Descriptive RDF for the audio file requested from the server



Information extraction process

Processing of received file by COAL Worker

2

DownloadWorker



MediaInfo

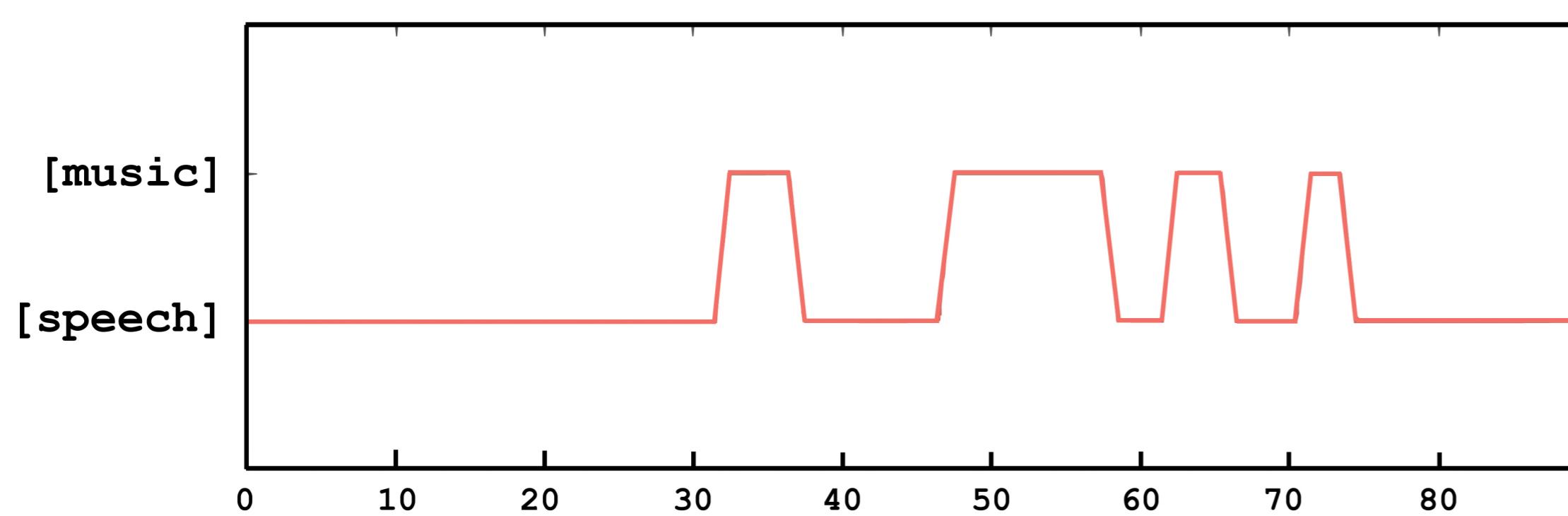
After the audio file is downloaded by the **DownloadWorker**, the **MediaInfoWorker** extracts information such as file size and format, length, BPM, channels, etc..

This is achieved using the *MediaInfo Library*.



ID3-Tag

Parallel to the **MediaInfoWorker**, the **ID3TagWorker** is also started. This uses the *mutagen Library* to extract information such as Interpret, Title, Album, Genre, Release Year, etc. from the mp3 file.



To segment the podcast content - also into "speech"- and "music"-segments - the file must be in .wav format. Therefore, the **ConverterWorker** is responsible for this.

The segmentation itself is then performed by the **SegmentationWorker** using the *pyAudioAnalysis Library*.

ConverterWorker

SegmentationWorker



Google Cloud Platform

Speech Recognition

The "speech" segments are recognized by the **SpeechRecognitionWorker** and sent to the *Google Speech API*. This identifies the spoken text and returns it in text form. The received content is then written to an RDF file.

SpeechRecWorker



MusicBrainz

Music Recognition

Parallel to this, the **MusicRecognitionWorker** processes the "music" segments. It generates a fingerprint for each "music" segment, which is then sent to *AcoustID*. As a result, we receive the *MusicBrainz ID* of the identified songs. These can then be used to query information such as Title, Interpret, etc. of the songs from *MusicBrainz*.

MusicRecWorker

@prefix foaf: <<http://xmlns.com/foaf/0.1/>> .

@prefix dc: <<http://purl.org/dc/elements/1.1/>> .

<<http://coal.s16a.org/resource?url=http://static.nico.is/testpodcast.mp3>>
a foaf:Document ;
dc:identifier "cache/2dc2ca1ca9f8a330b7e9e3aa2d04be46/data.ttl"^^xsd:string ;
foaf:topic "<http://static.nico.is/testpodcast.mp3>"^^xsd:string .

<<http://static.nico.is/testpodcast.mp3>>
dc:format "audio/mpeg"^^xsd:string ;
dcterms:extent "1446600"^^xsd:int ;
dcterms:modified "Thu, 16 Jun 2016 10:21:38 GMT"^^xsd:string ;
ebu:audioChannelNumber "2"^^xsd:int ;
nie:averageBitrate "128000"^^xsd:int ;
nie:bitRateMode "CBR"^^xsd:string ;
nie:duration "90592"^^xsd:long ;

- Erzeugtes RDF an den Client zurücksenden