

# COAL Metadata Extraction for Audio

podcast.mp3 (binary)

```
...öÖç,éä<lä yWRQ@«gÂ][2 ~S !!- T FÍ Y+
,Vò*ΔHGΩØI „I sH,Úm9_aeg
...RÁ^áß÷€^Vg U% 3^FU Ó Δ
VÉU'¢?O.öùmúÙr7âr á-ÅvgóùMñÅè≤kKÓ
YÉSÜΔ&<íó¥ådxYfæmj3Jç ÁçåaBUlðfÙ-
ð$g_«Åns^t- á Aæørãß÷ùΔ„% ≤¥e 'Fó÷ð
h^z Gç" @O»l'CEí^t%ojc'ÓΔ Ø"flö,¡Ω
°Ó;ÅÖ-¶ix}ÆlÙØ"‰o;viiΔ",X®AZ5£A...
je-œ$ñåÖç,éä<lä yWRQ@«gÂ][2 ~S !!- T
FÍ Y+,Vò*ΔHGΩØI „I sH,Úm9_aeg...
...RÁ^áß÷€^Vg U% 3^FU Ó Δ
VÉU'¢?O.öùmúÙr7âr á-ÅvgóùMñÅè≤kKÓ
```

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"
- Deskriptive RDF für die Audio-Datei vom Server anfordern



Informationextraktionsprozess

Verarbeitung der empfangenen Datei durch COAL Worker

2

DownloadWorker



## MediaInfo

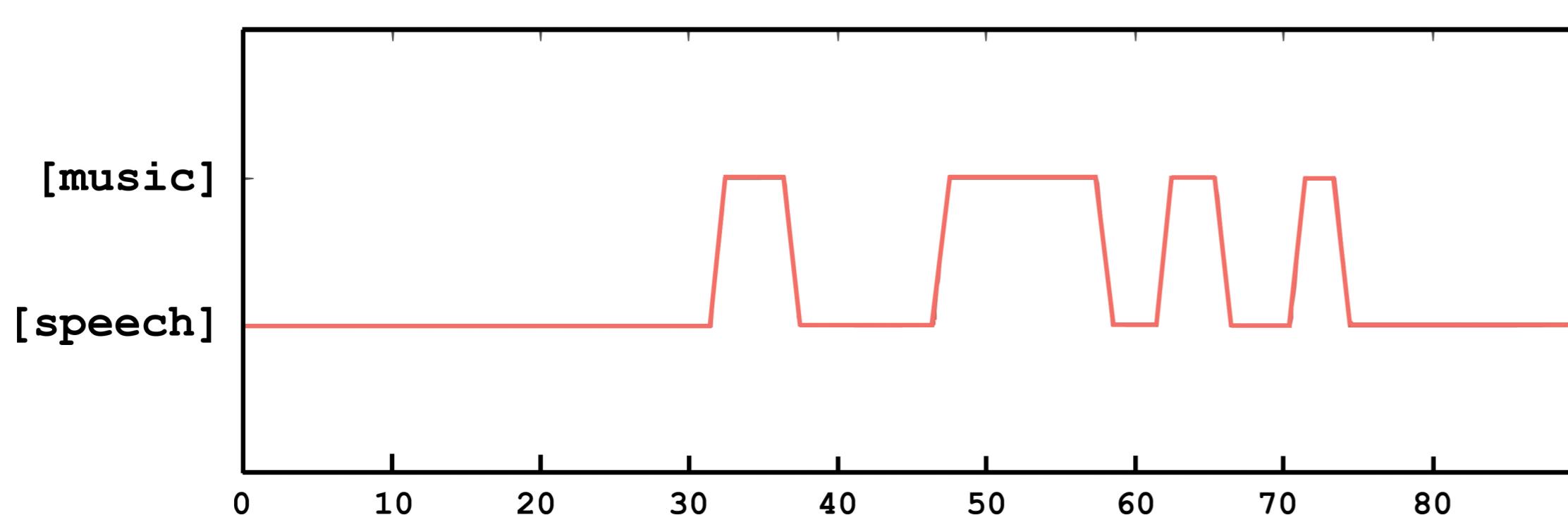
Nachdem die Audio-Datei vom **DownloadWorker** heruntergeladen wurde, extrahiert der **MediaInfoWorker** Informationen wie **Dateigröße und -format, Länge, BPM, Channels, etc..**

Hierzu wird die *MediaInfo-Library* genutzt.



## ID3-Tag

Parallel zum **MediaInfoWorker** wird auch der **ID3TagWorker** gestartet. Dieser nutzt die *mutagen-Library* um in der mp3-Datei gespeicherte Informationen wie **Interpret, Titel, Album, Genre, Veröffentlichungsjahr, etc.** zu extrahieren.



Um die Inhalte des Podcast zu segmentieren - also in "speech"- und "music"-Segmente zu unterteilen - muss die Datei als .wav vorliegen. Darum kümmert sich der **ConverterWorker**.

Das Segmentieren selber wird dann vom **SegmentationWorker** durchgeführt. Hierzu wird die *pyAudioAnalysis-Library* verwendet.

ConverterWorker

SegmentationWorker



Google Cloud Platform

## Speech Recognition

Die als "speech" erkannten Segmente werden vom **SpeechRecognitionWorker** an die *Google Speech API* gesendet. Dieses erkennt den **gesprochen Text** und sendet diesen in Textform zurück. Die empfangen Inhalte werden dann in die rdf-Datei geschrieben.

SpeechRecWorker



MusicBrainz

## Music Recognition

Parallel dazu verarbeitet der **MusicRecognitionWorker** die als "music" erkannten Segmente. Dazu erzeugt er für jedes "music"-Segment einen Fingerprint, welcher an *AcoustID* gesendet wird. Als Ergebnis erhalten wir die *MusicBrainz-ID* des erkannten Songs. Diese können wir nutzen um Informationen wie **Titel, Interpret, etc. des Songs** von *MusicBrainz* abzurufen.

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