TimeSide
An open web audio processing framework

Guillaume Pellerin[1], Thomas Fillon[1,2], Paul Brossier[1],

[1]Parisson, Paris, France

[2]LAM, Institut Jean Le Rond d'Alembert, UPMC Univ. Paris 06, UMR CNRS 7190

SoundSoftware Workshop - 8/07/2014

Table of contents

1. The Telemeta Project
   - Goals
   - CREM's platform
   - Technologies & Key features
   - Architecture
2. Goals
3. Use cases
   - General
   - The DIADEMS project
4. Architecture
   - Engine
   - Processors
   - Analyzer Result
5. Demos
6. ToDo lists
7. Lessons

The Telemeta Project

http://telemeta.org/

Main goals

- Archive, preserve and manage large audio database and related metadata
- Play audio data and read metadata synchronously
- Process audio data on demand through a modular architecture (no pre-processing needed)
- Index and share audio data through a collaborative web app
- Link audio data to various ontologies, external services and related multimedia files
- Manage users, share and access rules, copyrights easily through time

History of the project

- 2006: Define objectives = *open source web audio collaborative platform*
- 2007: First partner: french Center for Research in Ethnomusicology (CREM)
- 2011: Release of Telemeta 1.0 and deployment of the "*Sound archives of the CNRS - Musée de l'Homme*" http://archives.crem-cnrs.fr
- 2013 - 2014: Provide audio processing capabilities through the DIADEMS project

The Telemeta Project

http://telemeta.org/

Main goals

- Archive, preserve and manage large audio database and related metadata
- Play audio data and read metadata synchronously
- Process audio data on demand through a modular architecture (no pre-processing needed)
- Index and share audio data through a collaborative web app
- Link audio data to various ontologies, external services and related multimedia files
- Manage users, share and access rules, copyrights easily through time

History of the project

- 2006: Define objectives = *open source web audio collaborative platform*
- 2007: First partner: french Center for Research in Ethnomusicology (CREM)
- 2011: Release of Telemeta 1.0 and deployment of the "*Sound archives of the CNRS - Musée de l'Homme*" http://archives.crem-cnrs.fr
- 2013 - 2014: Provide audio processing capabilities through the DIADEMS project

CREM's platform

CREM
Centre de Recherche en Ethnomusicologie

Recherche

Bienvenue, Guillaume Pellerin | Profil | Aide | Déconnexion

Bureau | Archives | Géo-Navigateur | Recherche avancée | Utilisateurs | Admin

Archives sonores du CNRS - Musée de l'Homme

Le fonds d'archives sonores du CNRS – Musée de l'Homme rassemble des enregistrements inédits et publiés de musique et de traditions orales du monde entier, de 1900 à nos jours. Constitué de supports variés (cylindres, 78 tours, disques vinyles, cassettes, supports numériques), ce fonds se positionne parmi les plus importants d'Europe en terme de qualité, de quantité et de diversité.
Pour une présentation historique du fonds, voir le site du CREM.

Contenu

Géré par le Centre de Recherche en Ethnomusicologie (CREM) cette base de données répertorie :

✔ Plus de 30 000 documents inédits, dont les 2/3 sont sonorisés, répartie dans plus de 1 000 collections, représentant près de 4 000 heures d'enregistrements de terrain non publiés.
✔ Plus de 13 000 enregistrements édités, dont 3 000 sonorisés, dans plus de 4 600 collections, pour environ 3 700 heures (incluant plus de 5 000 disques dont beaucoup sont très rares).
✔ 199 pays sont représentés à travers plus de 1 200 groupes ethniques ou sociaux, donnant à entendre une large palette d'expressions musicales et chantées, de langues et de dialectes.

Certains enregistrements sont consultables avec un code d'accès. Pour l'obtenir écrivez à crem.lesc (at) mae.u-paris10.fr en expliquant les motifs de votre demande. Le fonds d'archives est également consultable sur les postes dédiés disponibles au CREM, à la Bibliothèque Eric de Dampierre, à la Médiathèque du Musée du Quai Branly et à la Bibliothèque du Muséum National d'Histoire Naturelle.

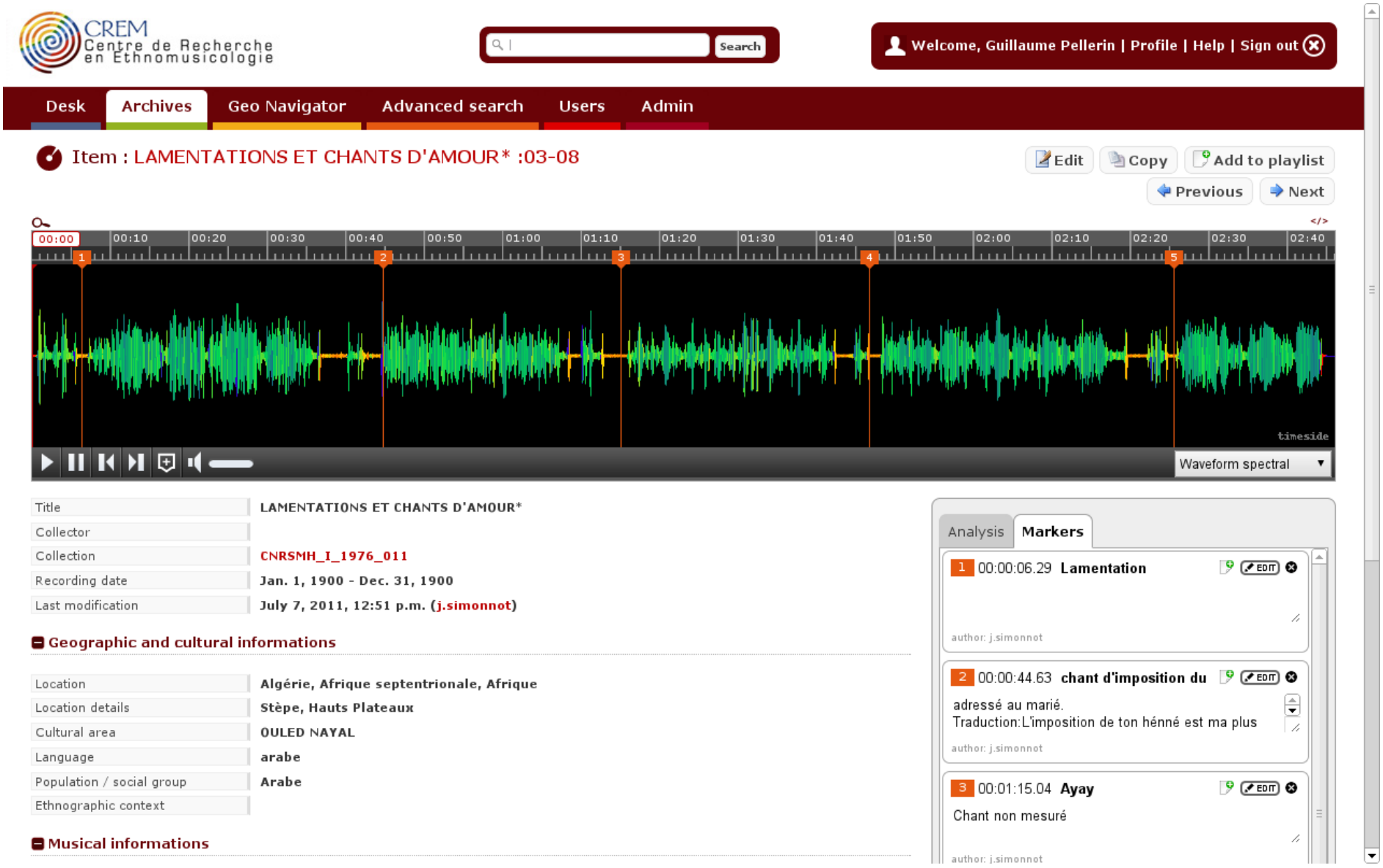
Organisation du catalogue

Le catalogue est organisé en 4 niveaux : Fonds, Corpus, Collection et Items. Le niveau principal de description est la Collection. Chacune regroupe un ensemble cohérent de fichiers audio (items) correspondant le plus souvent à des enregistrements collectés au cours d'une même mission de recherche ou à un disque publié. Certaines collections sont elles-mêmes regroupées en corpus et en fonds associés à des collecteurs.

Le nombre d'enregistrements mis en ligne sur la plateforme est en constante augmentation. Les fiches descriptives sont renseignées de manière collaborative par les usagers de la plateforme : chercheurs, étudiants, documentalistes.

Le CREM accueille toutes les collaborations visant à enrichir et valoriser ce précieux patrimoine. Ecrivez-nous à crem.lesc (at) mae.u-paris10.fr.

Telemeta - Web UI



http://archives.crem-cnrs.fr/archives/items/CNRSMH_I_1976_011_003_08/

Telemeta - Technologies & Key features

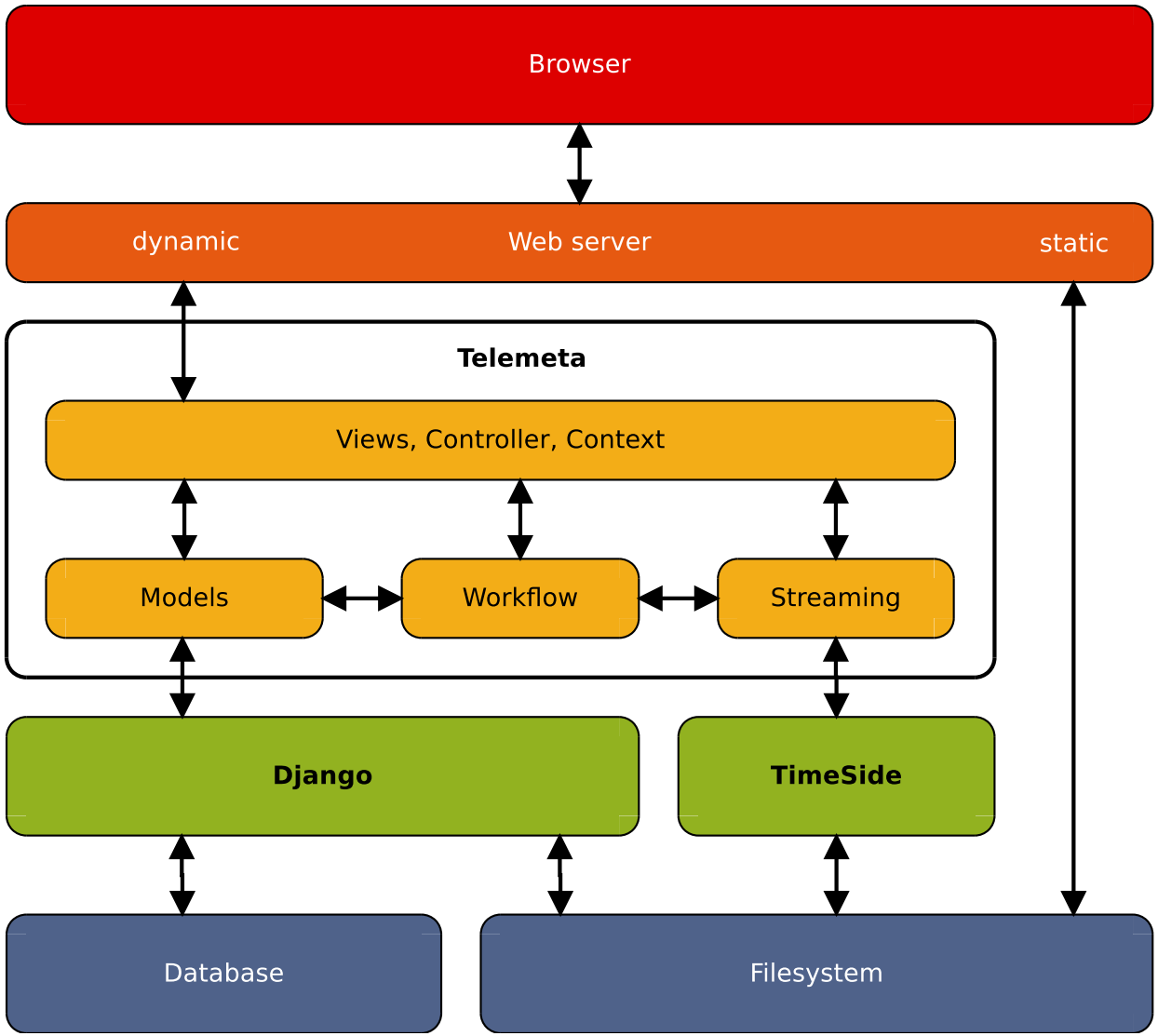
Technologies $\longrightarrow$ 100% Open Source!

- GNU / Linux : applications, libraries and kernel
- Python (cool and smart object oriented language with web and scientific libraries), Django (web platform, GStreamer (multimedia framework)
- MySQL, PostgreSQL, others : relational databases
- TimeSide : open web audio processing framework

Key features

- Pure HTML5 web user interface including dynamical forms and smart workflows
- On the fly audio analyzing, transcoding and metadata embedding in various formats
- Social editing with semantic ontologies, smart workflows, realtime tools, human or automatic annotations and segmentations
- User management with individual desk, playlists, profiles and access rights
- High level search engine (geolocation, instruments, ethnic groups, etc...)
- Data providers : DublinCore, OAI-PMH, RSS, XML, JSON and other
- Multi-language support (now english and french)

Telemeta - Architecture



TimeSide - Goals

Server side - TimeSide Engine

- Do asynchronous and fast audio processing with Python,
- Decode audio frames from ANY format into numpy arrays,
- Analyze audio content with state-of-the-art audio feature extraction libraries (Aubio, Yaafe, Vamp (experimental),
- Organize, serialize and save analysis metadata through various formats,
- Draw various fancy waveforms, spectrograms and other cool graphers,
- Transcode audio data in various media formats and stream them through web apps,

Client side - TimeSide UI

- Playback and interact on demand through a smart high-level HTML5 extensible player,
- Index, tag and organize semantic metadata

  (see Telemeta which embeds TimeSide).

Use cases

Usages

- Analyze large music audio datasets on demand over a robust and scalable platform
- Share audio data and metadata with experts to make them collaborate in editing, processing and discovering
- Build large statistical campaigns and vizualizations from ontologies, geographic data and sounds
- Scale the audio data through the web (URL indexes)

Domains

- Musicology
- Anthropology
- Museology
- Computer science
- Biology
- Ecology

Use cases

Usages

- Analyze large music audio datasets on demand over a robust and scalable platform
- Share audio data and metadata with experts to make them collaborate in editing, processing and discovering
- Build large statistical campaigns and vizualizations from ontologies, geographic data and sounds
- Scale the audio data through the web (URL indexes)

Domains

- Musicology
- Anthropology
- Museology
- Computer science
- Biology
- Ecology

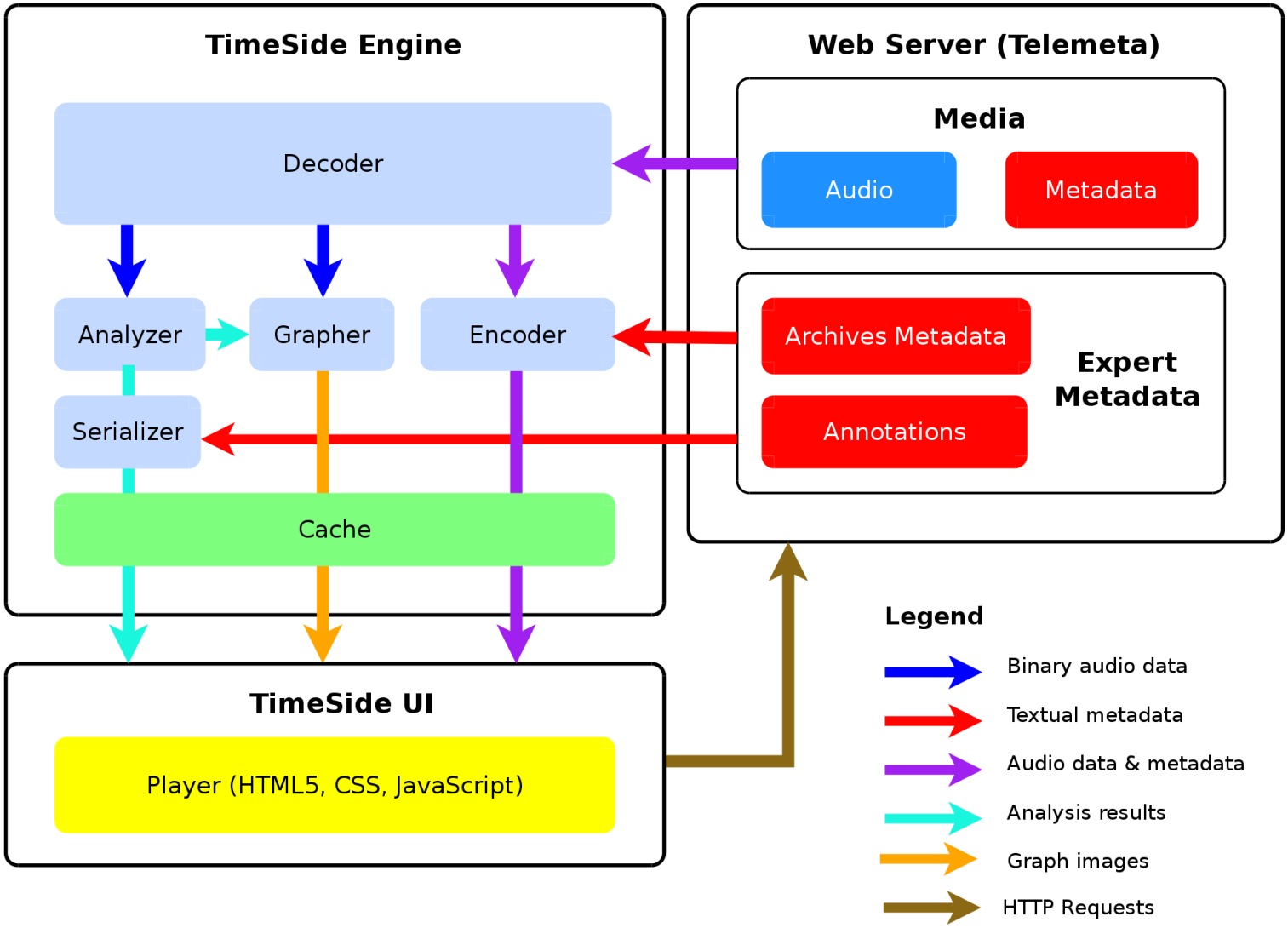
The DIADEMS project

- DIADEMS : Description, Indexation, Access to Sound and Ethnomusicological Documents
- Granted by ANR : french national research agency (ANR-12-CORD-0022)
- 3 years, 8 partners, 850 k€
- Apply and test MIR algorithms on large scale ethnomusicological data
- Define some high level interfaces to find new ways of explorations in large complex musical corpus
- New modes of collaboration between human science and computer science laboratories and researchers
- Define the vocabulary describing musical events in the usecase of ethnomusicilogy vs. signal processing
- http://www.irit.fr/recherches/SAMOVA/DIADEMS/fr/welcome/
- http://diadems.telemeta.org

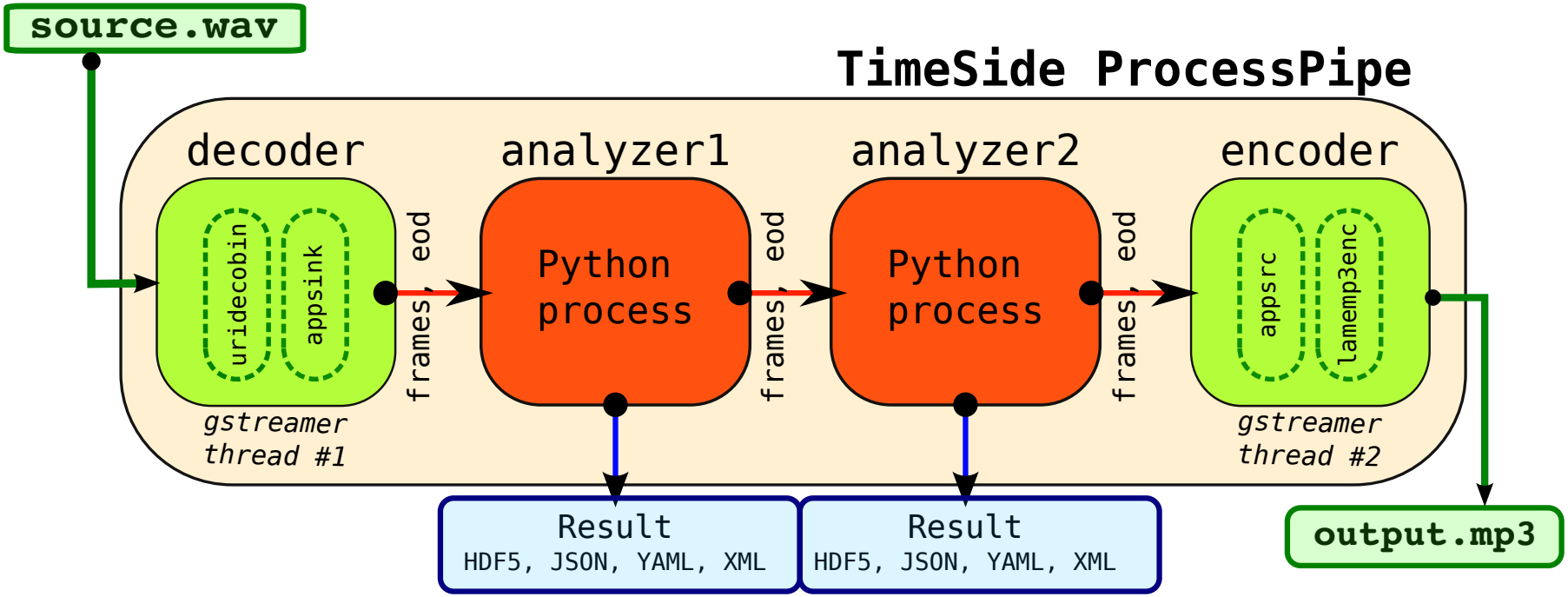
DIADEMS - Partners

- Partners :
  - IRIT (université Paul Sabatier, Toulouse 3)
  - LIMSI (universités Pierre et Marie Curie (UPMC, Paris 6) et Paris-Sud)
  - LAM (institut Jean Le Rond d'Alembert, UPMC)
  - LABRI (université de Bordeaux)
  - CREM (université Paris Ouest Nanterre La Défense)
  - LESC (université Paris Ouest Nanterre La Défense)
  - Museum d'Histoire Naturelle de Paris
  - Parisson
- Sponsors:

TimeSide - Architecture



TimeSide - Engine




```
pipe = (decoder | analyzer1 | analyzer2 | encoder)
```

Process Pipe

- On-the-fly audio processing by simultaneous processors (decoder, encoders, analyzers, graphers)
- Use of *Gstreamer* for audio decoding and encoding

Processors (plugins)

Decoders

- FileDecoder
- ArrayDecoder
- LiveDecoder

Encoders

- VorbisEncoder
- WavEncoder
- Mp3Encoder
- FlacEncoder
- AacEncoder
- WebMEncoder
- OpusEncoder
- AudioSink

Processors (plugins)

Decoders

- FileDecoder
- ArrayDecoder
- LiveDecoder

Encoders

- VorbisEncoder
- WavEncoder
- Mp3Encoder
- FlacEncoder
- AacEncoder
- WebMEncoder
- OpusEncoder
- AudioSink

Processors (plugins)

Analyzers

- AubioTemporal
- AubioPitch
- AubioMfcc
- AubioMelEnergy
- AubioSpecdesc
- Yaafe
- Spectrogram
- Waveform
- VampSimpleHost
- IRITSpeechEntropy
- IRITSpeech4Hz
- OnsetDetectionFunction
- LimsiSad

Processors (plugins)

Graphers

- Waveform
- WaveformCentroid
- WaveformTransparent
- WaveformContourBlack
- WaveformContourWhite
- SpectrogramLog
- SpectrogramLinear
- Display.aubio_pitch.pitch
- Display.odf
- Display.waveform_analyzer
- Display.irit_speech_4hz.segments

Analyzer Result

Result types: *time mode* x *data mode*

- Data modes:
  - Label
  - Value
- Time modes:
  - Global
  - Event
  - Segment
  - Framewise

Result Container

- ID Metadata
- Audio Metadata
- Parameters
- Data object
  - Label
  - Label Metadata (label, label_id, ...)
  - Value
  - Time
  - Duration
  - Frame Metadata (sample rate, blocksize, stepsize)

Export

- Serialization: HDF5, JSON, YAML, XML
- Display: *Ad hoc* rendering methods (depending on time and data modes)

Analyzer Result

Result types: *time mode* x *data mode*

- Data modes:
  - Label
  - Value
- Time modes:
  - Global
  - Event
  - Segment
  - Framewise

Result Container

- ID Metadata
- Audio Metadata
- Parameters
- Data object
  - Label
  - Label Metadata (label, label_id, ...)
  - Value
  - Time
  - Duration
  - Frame Metadata (sample rate, blocksize, stepsize)

Export

- Serialization: HDF5, JSON, YAML, XML
- Display: *Ad hoc* rendering methods (depending on time and data modes)

Analyzer Result

Result types: *time mode* x *data mode*

- Data modes:
  - Label
  - Value
- Time modes:
  - Global
  - Event
  - Segment
  - Framewise

Result Container

- ID Metadata
- Audio Metadata
- Parameters
- Data object
  - Label
  - Label Metadata (label, label_id, ...)
  - Value
  - Time
  - Duration
  - Frame Metadata (sample rate, blocksize, stepsize)

Export

- Serialization: HDF5, JSON, YAML, XML
- Display: *Ad hoc* rendering methods (depending on time and data modes)

Analyzer Result

Result types: *time mode* x *data mode*

- Data modes:
  - Label
  - Value
- Time modes:
  - Global
  - Event
  - Segment
  - Framewise

Result Container

- ID Metadata
- Audio Metadata
- Parameters
- Data object
  - Label
  - Label Metadata (label, label_id, ...)
  - Value
  - Time
  - Duration
  - Frame Metadata (sample rate, blocksize, stepsize)

Export

- Serialization: HDF5, JSON, YAML, XML
- Display: *Ad hoc* rendering methods (depending on time and data modes)

Analyzer Result

Result types: *time mode* x *data mode*

- Data modes:
  - Label
  - Value
- Time modes:
  - Global
  - Event
  - Segment
  - Framewise

Result Container

- ID Metadata
- Audio Metadata
- Parameters
- Data object
  - Label
  - Label Metadata (label, label_id, ...)
  - Value
  - Time
  - Duration
  - Frame Metadata (sample rate, blocksize, stepsize)

Export

- Serialization: HDF5, JSON, YAML, XML
- Display: *Ad hoc* rendering methods (depending on time and data modes)

Analyzer Result

Result types: *time mode* x *data mode*

- Data modes:
  - Label
  - Value
- Time modes:
  - Global
  - Event
  - Segment
  - Framewise

Result Container

- ID Metadata
- Audio Metadata
- Parameters
- Data object
  - Label
  - Label Metadata (label, label_id, ...)
  - Value
  - Time
  - Duration
  - Frame Metadata (sample rate, blocksize, stepsize)

Export

- Serialization: HDF5, JSON, YAML, XML
- Display: *Ad hoc* rendering methods (depending on time and data modes)

Analyzer Result

Result types: *time mode* x *data mode*

- Data modes:
  - Label
  - Value
- Time modes:
  - Global
  - Event
  - Segment
  - Framewise

Result Container

- ID Metadata
- Audio Metadata
- Parameters
- Data object
  - Label
  - Label Metadata (label, label_id, ...)
  - Value
  - Time
  - Duration
  - Frame Metadata (sample rate, blocksize, stepsize)

Export

- Serialization: HDF5, JSON, YAML, XML
- Display: *Ad hoc* rendering methods (depending on time and data modes)

Analyzer Result

Result types: *time mode* x *data mode*

- Data modes:
  - Label
  - Value
- Time modes:
  - Global
  - Event
  - Segment
  - Framewise

Result Container

- ID Metadata
- Audio Metadata
- Parameters
- Data object
  - Label
  - Label Metadata (label, label_id, ...)
  - Value
  - Time
  - Duration
  - Frame Metadata (sample rate, blocksize, stepsize)

Export

- Serialization: HDF5, JSON, YAML, XML
- Display: *Ad hoc* rendering methods (depending on time and data modes)

Analyzer Result

Result types: *time mode* x *data mode*

- Data modes:
  - Label
  - Value
- Time modes:
  - Global
  - Event
  - Segment
  - Framewise

Result Container

- ID Metadata
- Audio Metadata
- Parameters
- Data object
  - Label
  - Label Metadata (label, label_id, ...)
  - Value
  - Time
  - Duration
  - Frame Metadata (sample rate, blocksize, stepsize)

Export

- Serialization: HDF5, JSON, YAML, XML
- Display: *Ad hoc* rendering methods (depending on time and data modes)

Analyzer Result

Result types: *time mode* x *data mode*

- Data modes:
  - Label
  - Value
- Time modes:
  - Global
  - Event
  - Segment
  - Framewise

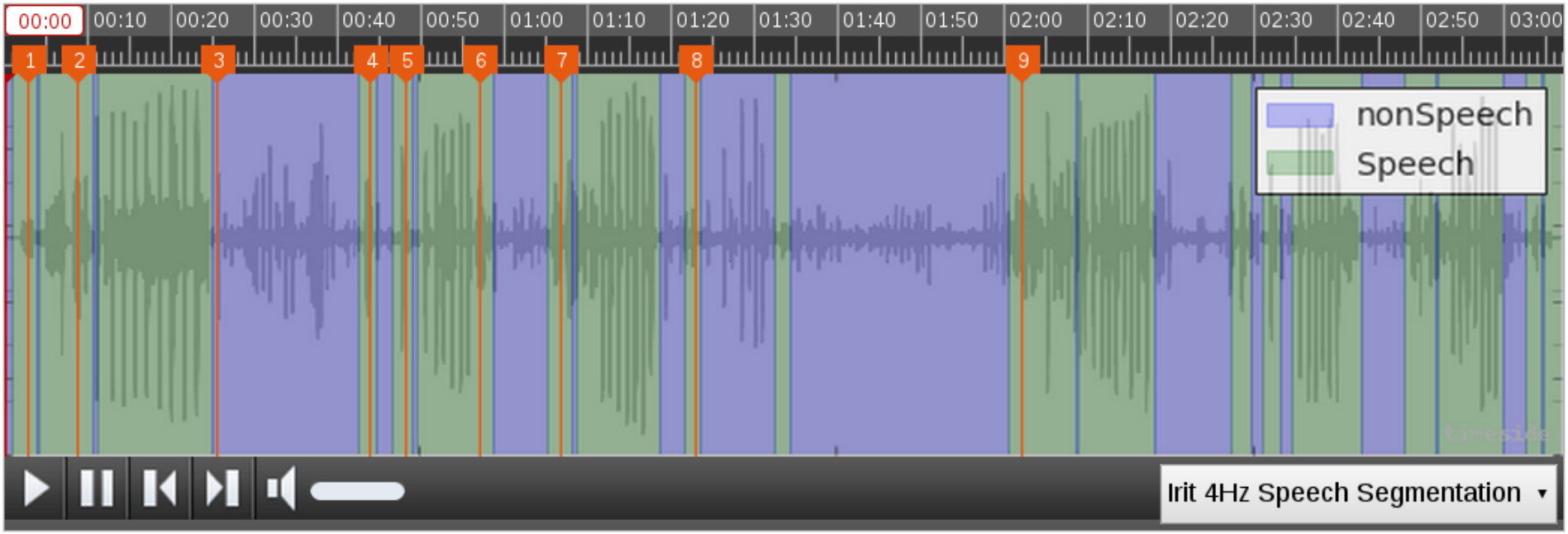
Result Container

- ID Metadata
- Audio Metadata
- Parameters
- Data object
  - Label
  - Label Metadata (label, label_id, ...)
  - Value
  - Time
  - Duration
  - Frame Metadata (sample rate, blocksize, stepsize)

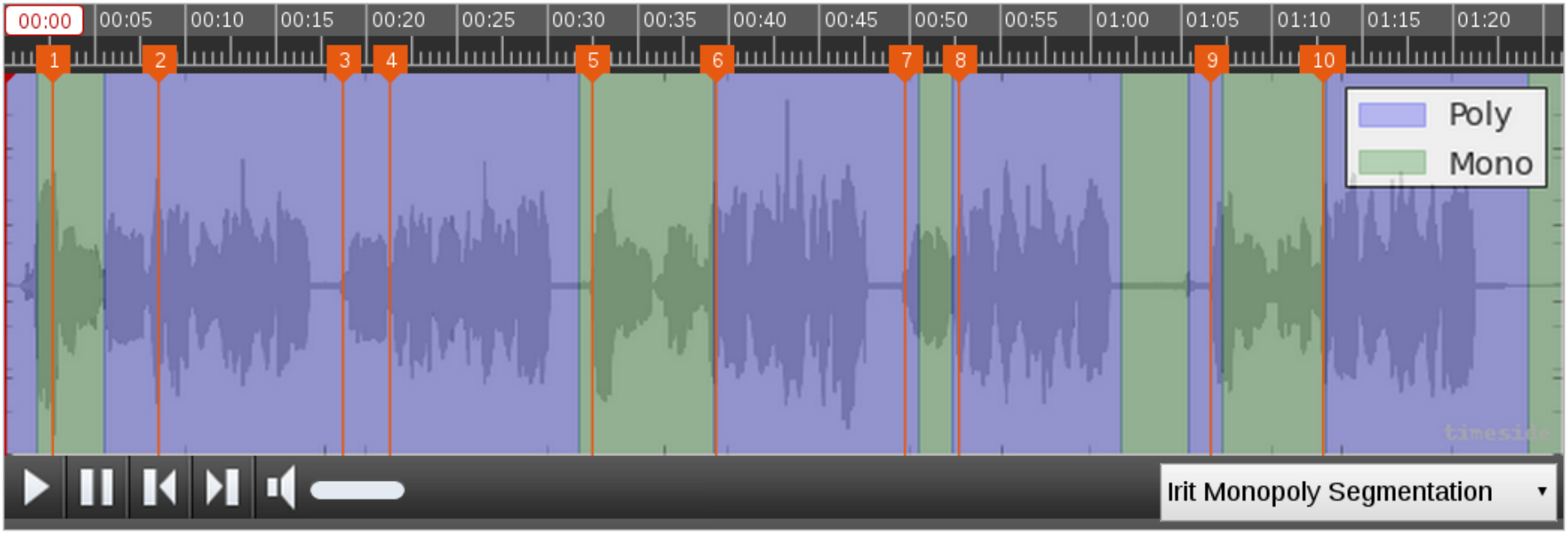
Export

- Serialization: HDF5, JSON, YAML, XML
- Display: *Ad hoc* rendering methods (depending on time and data modes)

Analyzer result examples



Analyzer result examples



Documentation and demos

Links

- Official documentation
- Notebooks
- Online example 1
- Online Example 2
- DIADEMS datasets

ToDo lists

Telemeta

- Upgrade against the web framework (Django 1.7) and geolocation services
- Enhance user interface (full HTML 5 + web audio API)
  - For annotations and segmentations in a collaborative manner
  - Provide import capabilities and feedback loop between manual and automatic annotations
  - Fancy displays of automatic analysis results (zoomable + synchronized with audio)
  - Add a User interface to control and tune the analysis parameters
  - Add public and enhanced user playlists
- More documentation!

TimeSide

- Tiny web server based on Django (done)
- Process task manager (done)
- Add more audio & acoustic analysis tools for automatic analysis
- Add more automatic segmentation and classification tools to support various semantic ontologies (cf. thesaurus)

ToDo lists

Telemeta

- Upgrade against the web framework (Django 1.7) and geolocation services
- Enhance user interface (full HTML 5 + web audio API)
  - For annotations and segmentations in a collaborative manner
  - Provide import capabilities and feedback loop between manual and automatic annotations
  - Fancy displays of automatic analysis results (zoomable + synchronized with audio)
  - Add a User interface to control and tune the analysis parameters
  - Add public and enhanced user playlists
- More documentation!

TimeSide

- Tiny web server based on Django (done)
- Process task manager (done)
- Add more audio & acoustic analysis tools for automatic analysis
- Add more automatic segmentation and classification tools to support various semantic ontologies (cf. thesaurus)

Lessons

Lessons learned from a 7 years old project

- Simplicity is better than complexity (KISS)
- Modularity is only accessible with a flexible language (thanks Python!)
- Models and Objects are more important than Technologies
- A good workflow is defined by the users themselves through feedback and constant revisions
- Prototyping is a crucial part of the development process
- A good platform relies on standards, not on formats
- The Open Source ecosystem provides some tremendous possibilities to develop, deploy and scale a platform project

The End

Thank you!
We are looking for new collaborations in various use cases...
Let's keep in touch!

Links

- github.com/yomguy/TimeSide
- github.com/Parisson/Telemeta-doc
- telemeta.org
- @telemeta

Contact me

- guillaume@parisson.com
- @yomguy
- github.com/yomguy/
- +GuillaumePellerin
- fr.linkedin.com/in/guillaumepellerin