Deliverable D2.7
D2.7 Service integration guidelines

Grant agreement nr	688382
Project full title	Audio Commons: An Ecosystem for Creative Reuse of Audio Content
Project acronym	AudioCommons
Project duration	36 Months (February 2016 - January 2019)
Work package	WP2
Due date	31 January 2019 (M36)
Submission date	31 January 2019 (M36)
Report availability	Public (X), Confidential ( )
Task leader	QMUL
Authors	Sasha Rudan, Miguel Ceriani, Fabio Viola, Mathieu Barthet, Sinisha Rudan, Francesco Antoniazzi, György Fazekas
Document status	Draft ( ), Final (X)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

Table of contents

Table of contents 2
Executive Summary 4
1 Introduction 5
1.1 Main objectives and goals 5
1.2 Terminology 5
2 Audio Commons Mediator 6
2.1 Introduction 6
2.2 Technologies 6
2.3 Authentication 6
2.4. Implementation Changes 7
2.4.1 End user perspective 7
2.4.2 Service Integration perspective 8
2.5 Semantic Aspects of the Mediator 8
2.6 Steps towards integrating a new service provider 13
2.6.1. Create a new repository 13
2.6.2. Provide all necessary mappings templates 13
2.6.2.1. External variables 14
2.6.3. Register in ".gitmodules" file 14
2.6.4. Register inside the mediator's config file "mediaconf.yaml" 14
2.6.5. Provide the service uri and key 15
2.7 A short introduction to data models 15
2.7.1 Description of basic AC ontology classes 15
2.7.2 Real-world example 17

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

2.8 Scenarios for Testing and Evaluating Service Integration 23
2.8.1 Isolated test of SPARQL-Generate templates 23
2.8.2 Providing mapping templates to ACE maintainers 24
2.8.3 Deploying one's own AC mediator ecosystem 24
2.8.4. Integration evaluation 24
2.9. Open Challenges 25
2.9.1. Sequential extension of the mediator search workflow 27
2.9.2 Parallel extension of the the mediator search workflow 27
2.9.3 Prototyped Solution 28
3 Conclusion **30**
4 References **31**
5. APPENDIX **32**
5.1 Access Info 32
5.2. SPARQL-Generate Response 32
5.3. JSON-LD response from mediator 37

Executive Summary

This deliverable is part of WP2, a work package that describes the Audio Commons Ontology and Audio Commons API specification. It is **intended for developers** who wish to include new services in the ACE using the latest version of the Audio Commons mediator, our software component for service integration. The Mediator is a component that lies at the heart of the Audio Commons Ecosystem (ACE) and interconnects all Audio Commons services. As part of WP2, an initial version of the Audio Commons Mediator was developed (see D2.6). This deliverable describes the next generation of this component, version 2 (v2), a semantic mediator that brings several improvements over the previous version (v1). The improvements offered by the service integration procedure described in these guidelines consists primarily in the declarative nature of the integration. This should be contrasted with the initial service integration method described in Deliverable D2.6. Using the solution described in D2.6, any new ACE service provider had to be integrated by means of a procedural implementation using a set of interfaces (also called mixins in Python 3 terms which was the implementation language of choice).

In the final implementation the integration happens at a declarative level using a generic software component capable of providing automatic API mappings. Using the SPARQL-Generate language (an extension of the SPARQL query language) we facilitate integration defined through a mapping template that SPARQL-Generate can interpret and translate to standard JSON RESTful API responses also compatible with a semantic RDF graph using the JSON for Linked Data (JSON-LD) syntax. This has two advantages. First, the mappings between service APIs and the Audio Commons API is separated from the procedural implementation, making it easier to maintain. Second, the API responses may be interpreted as a semantic graph and linked with other data sources using linked data principles. At the same time, the simplicity of standard JSON responses is retained and available for developers who do not wish to exploit the advanced features of the semantic representation.

We start this deliverable by introducing the **novel infrastructure** that contributes to the new mechanism for the integration of 3rd party services, as well as **guidelines of service integration** together with references to **documented examples**. At the end of this deliverable, we describe the **open challenges** of the current infrastructure, present a number of **use cases** addressing challenges and provide some **future directions** for future developments of the mediator with its declarative framework and its implications on the Audio Commons Ecosystem.

1 Introduction

1.1 Main objectives and goals

The main objective of this deliverable was building an improved version of the AudioCommons mediator and provide guidelines for the integration of services with the new mediator. The task was planned around the following objectives:

- design a solution that utilises declarative sound providers integration
- Integrate existing sound providers in a declarative manner to test our solution
- describe guidelines for future integration of new services
- evaluate the challenges related to the proposed solution and propose future improvements

1.2 Terminology

AudioCommons: reference to the EC H2020 funded project AudioCommons (grant agreement No 688382).

Audio Commons Initiative: foster the understanding of the AudioCommons project core ideas beyond the lifetime and specific scope of the funded project. The term “Audio Commons Initiative” is used to imply i) our motivation to continue supporting the Audio Commons Ecosystem and its ideas after the lifetime of the funded project, and ii) our intention to engage new stakeholders which are not officially part of the project consortium.

Audio Commons Ecosystem (ACE): a series of technologies and actors involved in the publishing and consumption of the Audio Commons content.

Audio Commons content (AC): audio content released under Creative Commons licenses and enhanced with meaningful contextual information (e.g., annotations, license information) that enables its publication in the ACE.

Content creator: individual users, industries or other actors that create audio content and publish in the ACE through content providers.

Content provider: services that expose content created by content creators to the ACE.

Content user: individual users, industries or other actors who use the content exposed by content providers and created by the content creators in their creative workflows.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N° 688382

2 Audio Commons Mediator

2.1 Introduction

The release of the first mediator (described in the D2.5 deliverable) provided a "hard-coded" mechanism of service integration. It supported a set of interfaces (also called mixins in Python terminology) that were supposed to be implemented in order to integrate a service appropriately. The initial version of the mediator (v1) focused on the following types of services: Search, Authentication, and Licensing. Mediator v2 focuses mainly on search services and this deliverable describes the integration of search services extending the initial mediator by allowing the reuse of its services.

The main incentive for developing mediator v2 was not necessarily to replace mediator v1, but rather to explore and evaluate the feasibility of using an ontological approach in search API implementation and the underlying infrastructure. Therefore, v2 does not yet have all necessary features to completely obsolete mediator v1, but rather it offers new ACE members and prospective clients an advanced environment that supports using the Audio Commons Ontology (D2.3) and the exploration of advanced semantic integrations with 3rd party ontologies. This has been demonstrated during the Abbey Road Hackathon, see in-use evaluations described in D6.12 section 6.3 as well as in D7.7.

The ***source code*** of the mediator v2 is available at the AudioCommons repository:
https://github.com/AudioCommons/semanticMediator

2.2 Technologies

Mediator v2 builds upon the following technologies:

- Python Flask framework (Flask is a micro web framework written in Python and based on the Werkzeug toolkit and Jinja2 template engine; it is BSD licensed)
- MEAN stack: MongoDb, Express.JS, Angular 6 and Node.JS

and integrates following components:

- SEPA (SPARQL Event Processing Architecture) aims to support the development of distributed, Web-based and context aware applications, in particular with reference to dynamic contexts where detecting and reacting to events is critical [SEPA]
- SPARQL-Generate - is a component that implements SPARQL-Generate extension used for translating non-RDF search-services' APIs into AC Ontology [SPARQL-Generate-component]
- ColaboFlow.Audit - Part of the Colabo.Space framework; supports logging and visualizing mediator activities and data/control flows; it also enables particular services (e.g. licensing, sessions, etc.) [Colabo][ColaboFlow]
- ColaboFlow.Go - Part of the Colabo.Space framework; supports executing remote activities/micro-services through gRPC, and data/control flows [gRPC] [ProtoBuff]

2.3 Authentication

The authentication workflow of the semantic mediator (current version, v2) is identical to the previous

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

v1 of the mediator, as it is offered through the mediator v2, which is described in the deliverable D2.4 and additionally at the official documentation of the authentication workflows and API in the mediator v1 [mediator-v1-authentication]. To support the integration of v1 and v2 of the mediator, additional services are provided and their full integration can be seen on Fig. 2.3.1.

From the perspective of the ACE client, the only difference in authentication consists in the usage of the sub-path "***/authenticate/***" in the mediator access. For example, the login access point is now: https://m2.audiocommons.org/authenticate/login instead of the previous access point: https://m.audiocommons.org/login.

Infrastructure-wise, mediator v1 is integrated into the v2 ecosystem in a ***slimmed version*** that only includes the components that are necessary to support authentication. Authentication states are stored in the ***Postgres*** database and later checked through the **authorization component** on behalf of the semantic mediator.

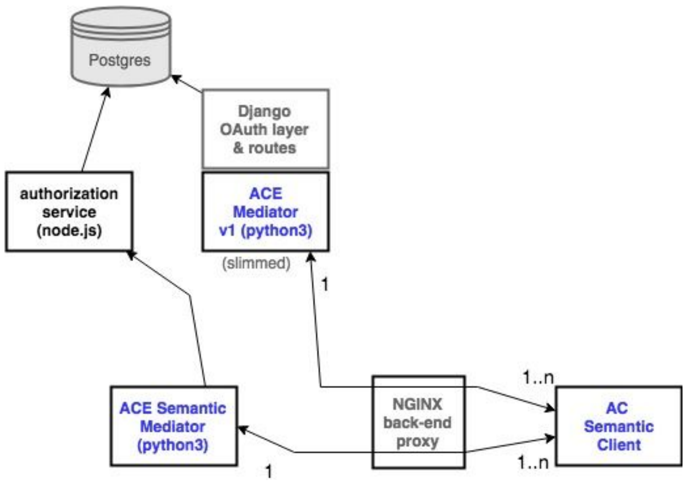

Postgres
Django OAuth layer & routes
authorization service (node.js)
ACE Mediator v1 (python3)
(slimmed)
1
1..n
NGINX back-end proxy
AC Semantic Client
ACE Semantic Mediator (python3)
1
1..n

Fig. 2.3.1. Authentication component of the v2 of the mediator

2.4. Implementation Changes

Mediator v2, also referred to as *semantic mediator* in the rest of this document, is a major revision and it is not backwards compatible with the previous version of the mediator v1 when it comes to the implementation of ACE service integration components. That is, interfaces developed for the initial version cannot be used with the final version.

2.4.1 End user perspective

The end user (client) is now empowered with a full ontology-based API response. This opens various possibilities in continuous knowledge evolution and expansion on the client side (discussed more in D2.8 and D6.12), and allows for support provided by knowledge federation through 3rd party ontological knowledge spaces. However, this means that the previous regular RESTful JSON API is not supported anymore. Instead, an enhanced RESTful JSON API is provided with responses that may

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

be interpreted as both conventional JSON structures or semantic graphs via JSON-LD notation.

On the other hand, some of the services are not updated and are still provided through integration with the previous version of the mediator.

It is important to note that v2 mediator does not provide a separate retrieval access point but it rather works in a ***"sync"-mode*** from the client perspective; it retrieves all search service responses and then returns them all merged in a single response and closes the connection to the client.

2.4.2 Service Integration perspective

The service integration model has changed fundamentally compared to the initial mediator and it is described in this deliverable. It is based on a declarative model of translation of service API into the Audio Commons Ontology. For this purpose, a SPARQL translator is used, which in most cases translates the services' RESTful response into an RDF conforming to the AC Ontology. We use a SPARQL-Generate component for this tasks.

2.5 Semantic Aspects of the Mediator

The semantic support of the mediator is the core change and an essential motivation for v2 of the mediator. A simple model of the mediator can be seen on Fig. 2.5.1. As we can see, each search service is supported using a ***Semantic Adapter*** (SA) component. Unlike mediator v1, SAs are not separate procedural implementations (Python code), but rather a single generic component that is driven using declarative mappings to facilitate adaptation to each unique service provider in the ACE.

This is realized by means of a special extension of SPARQL - a SPARQL-Generate extension implemented within the SPARQL-Generate component; [SPARQL-Generate-extension] and [SPARQL-Generate-component], respectively. SPARQL-Generate is an extension of SPARQL 1.1 for querying not only RDF datasets, but also documents in arbitrary formats. It offers a simple template-based option to generate RDF Graphs from documents and different streams. These templates are described in terms of the SPARQL-Generate language [SPARQL-Generate-language].

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

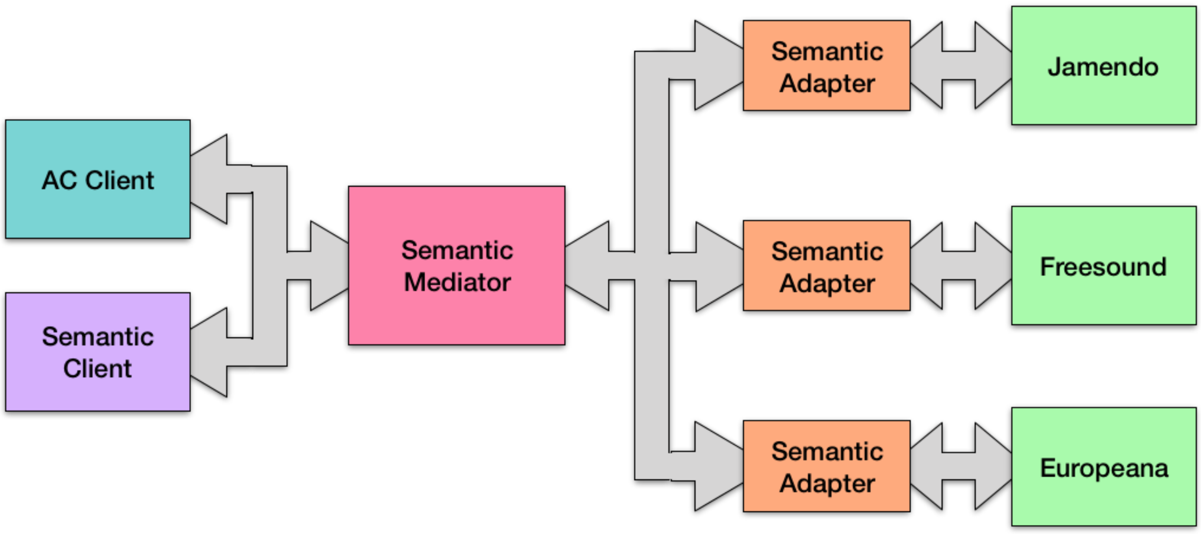


Figure 2.5.1. Simplified model of the semantic mediator (v2)

In a greater details Fig. 2.5.2. presents a detailed model of mediator v2. As one can see, the internal data model for the mediator is semantic, using RDF with terms defined by the AC Ontology. At the entry points (towards sound providers' search services) are JSON->RDF mappers that each new search service that is integrated should provide, and at the exit (towards the (semantic) client) there are translators from RDF to JSON-LD for semantic clients, or JSON for standard clients. JSON-LD Frame is introduced in order to guarantee predetermined structure and representation/mapping of RDF results into JSON results.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

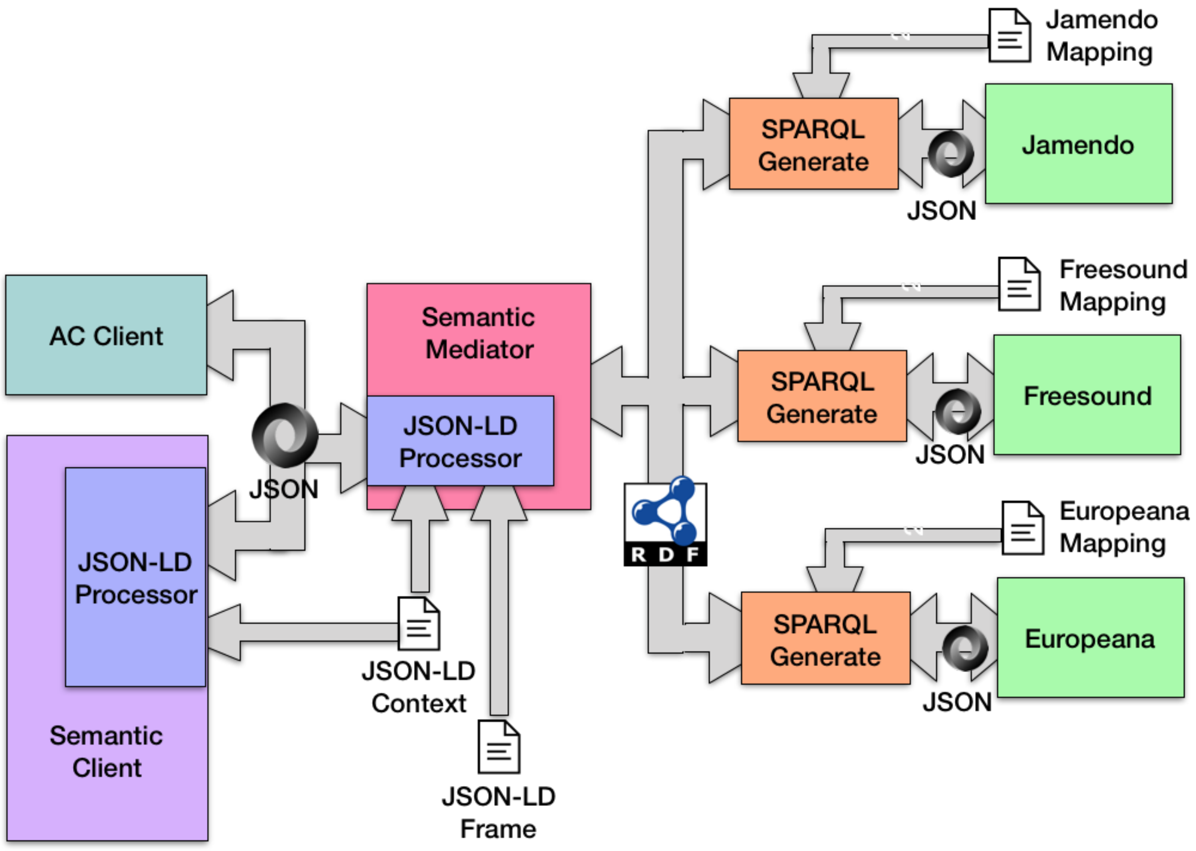


Figure 2.5.2. Detailed model of the semantic mediator (v2)

Fig. 2.5.3. presents a detailed component (services) diagram of the mediator v2 that show all the relevant ACE services and the integration of 3rd party services. One can also understand in greater detail the mechanism whereby the authorization support of the v1 is integrated in the v2 ecosystem.

Mediator v2 is deployed at https://m2.audiocommons.org, with the documentation at the root, API at the "**/api**" path, and the authorization service at the "**/authorization**" path. A dashboard that provides access to logging and action reports is available at "**/dashboard**" path.

Fig. 2.5.4. illustrates the detailed activity diagram of mediator v2. We can see both control flows and data flows running through the mediator during the search request coming from a client. There are N different search services that mediator supports, and for each the mediator spawns a separate thread to run access to the service concurrently, parsing results and storing them into its internal storage. Currently, the mediator supports native triplestore; we utilize Blazegraph but others triplestores compliant with SPARQL-Query can be supplemented.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

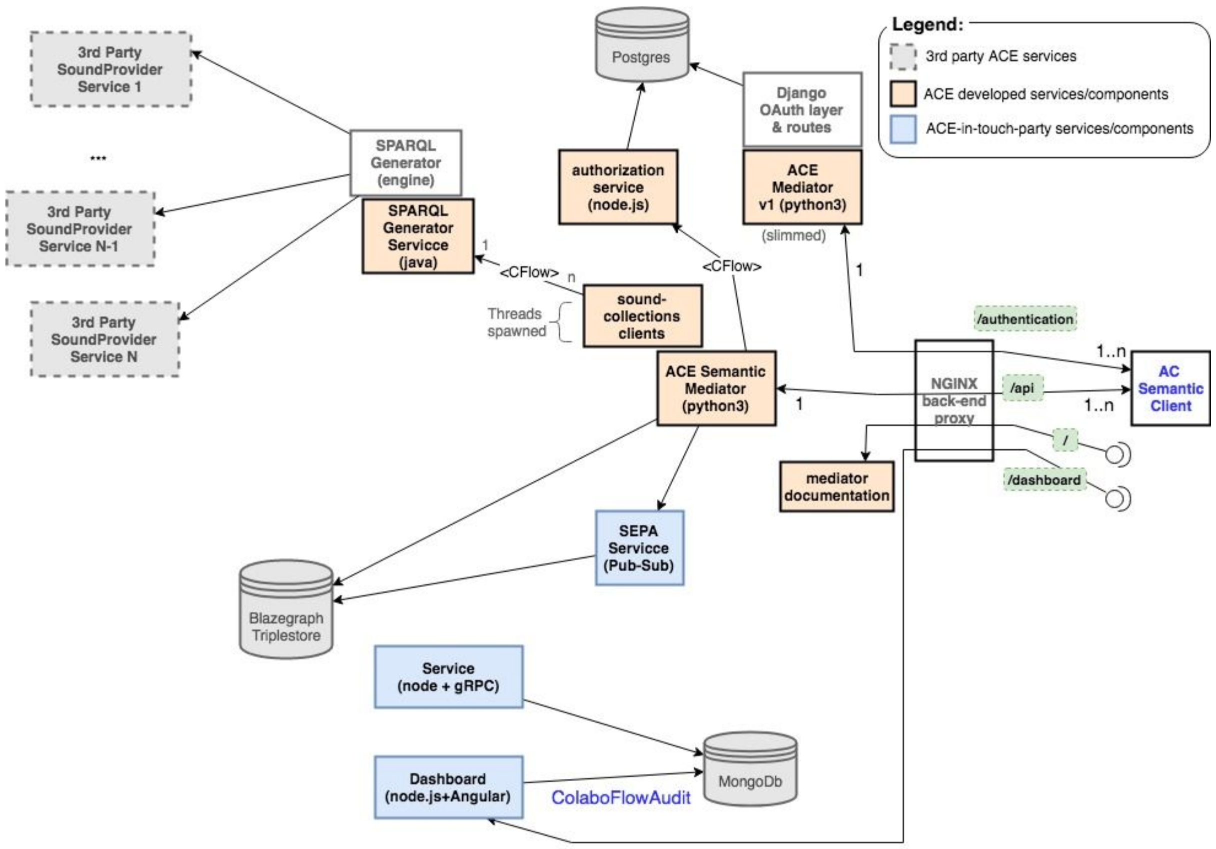

Legend:
3rd party ACE services
ACE developed services/components
ACE-in-touch-party services/components
3rd Party SoundProvider Service 1
...
3rd Party SoundProvider Service N-1
3rd Party SoundProvider Service N
SPARQL Generator (engine)
SPARQL Generator Servicce (java)
Postgres
Django OAuth layer & routes
authorization service (node.js)
ACE Mediator v1 (python3)
(slimmed)
<CFlow>
<CFlow>
1
n
1
Threads spawned
sound-collections clients
ACE Semantic Mediator (python3)
1
NGINX back-end proxy
/authentication
/api
/
/dashboard
1..n
1..n
AC Semantic Client
mediator documentation
SEPA Servicce (Pub-Sub)
Blazegraph Triplestore
Service (node + gRPC)
Dashboard (node.js+Angular)
ColaboFlowAudit
MongoDb

Figure 2.5.3. Detailed component (services) diagram of the semantic mediator (v2)

The mediator has integrated SEPA support that supports a regular SPARQL-Query interface, but it also offers support for a pub-sub pattern where semantic clients can subscribe to listen for changes in the knowledge subspace. We have investigated possibilities for full deployment of SEPA into the lifecycle of the mediator and providing support for a pub-sub pattern for semantic clients. In such a ***pub-sub-mode*** (as opposed to the currently employed ***"sync"-mode***) where they would get access to search results incrementally, after receiving and processing search results from each separate sound provider. Our initial evaluation is positive, and the solution has been prototyped with positive results, but it is not part of this deliverable or of this grant, and has therefore been left for future work.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

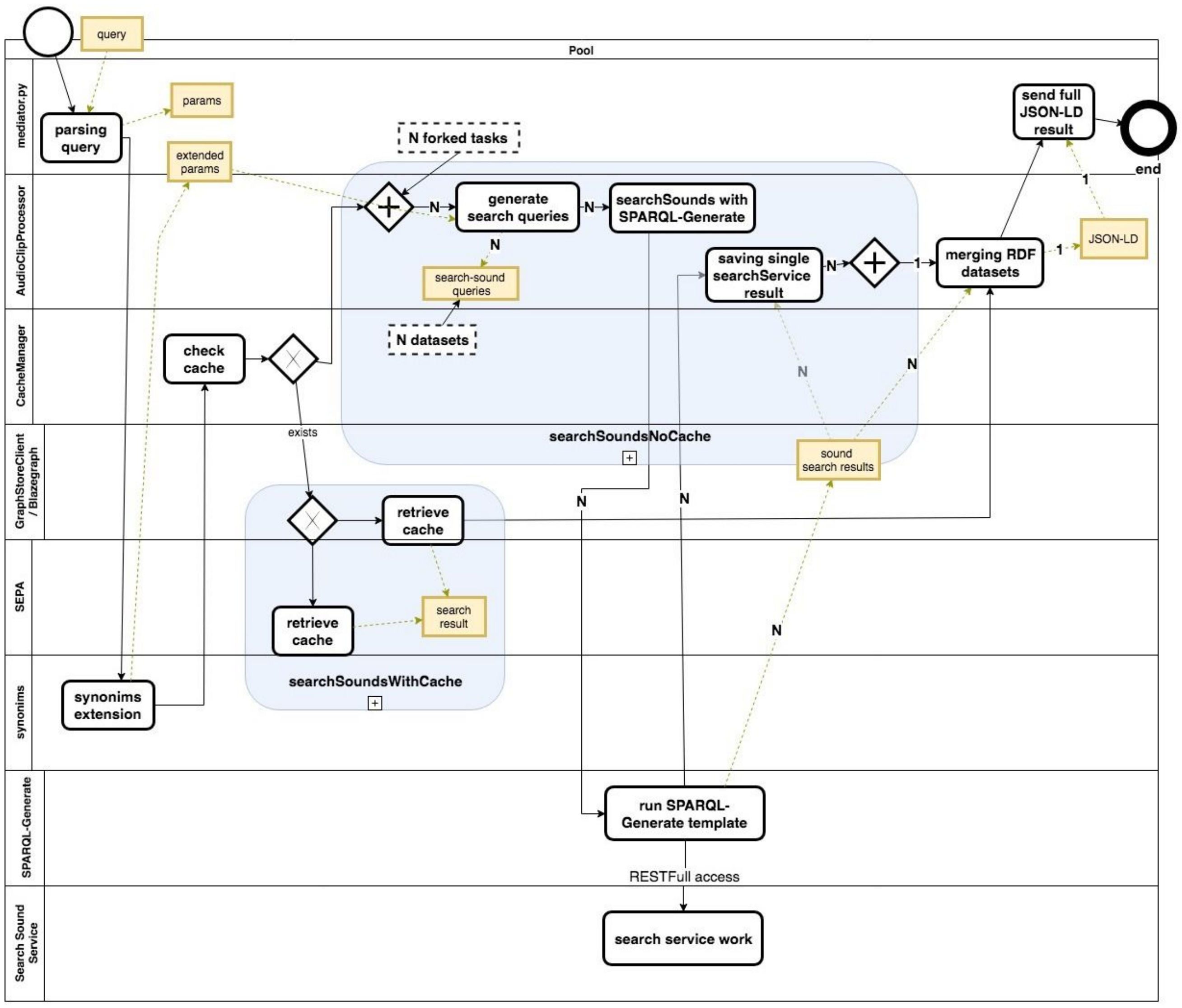


Figure 2.5.4. Detailed action diagram for a **search-sounds** use case in the semantic mediator (v2), with dataflows and workflows

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

2.6 Steps towards integrating a new service provider

This section provides a practical step-by-step guide for developers who wish to integrate a new service provider to the ACE using the semantic mediator. This section focusses on adding search capabilities over a new content repository not currently included in Audio Commons, assuming it has a suitable API which can be mapped using our declarative framework.

2.6.1. Create a new repository

Create a new repository that will contain mappings for the service.

Here are some examples of existing mappings:

- https://github.com/AudioCommons/semantic-mappings-europeana
- https://github.com/AudioCommons/semantic-mappings-freesound
- https://github.com/AudioCommons/semantic-mappings-jamendo

A commented example of SPARQL-Generate mappings in the Freesound service mappings can be found at the following address:

https://github.com/AudioCommons/semantic-mappings-freesound/blob/master/data-adapters/audio-search-by-text.rq

2.6.2. Provide all necessary mappings templates

For each service there might be multiple access points or search types; **1)** searches for **sounds** (by text), **2)** search for **categories** (by text), or just 3) **showing** a particular sound by id. Each of these different ways of accessing to the sound search service has to be described properly by a mapping template. Service integrator is free to provide only needed/supported mappings/access

Creating mapping templates requires knowledge of the SPARQL query language [SPARQL-query-language], as well as the SPARQL-Generate language [SPARQL-Generate-language], which tends to be a natural extension of the SPARQL query language.

In mediator v2 the SPARQL-Generate descriptive templates are used to translate search-services' results, that are service-specific and mostly JSON encoded, into uniform RDF response. The RDF response should conform to Audio Commons ontology described in deliverable D2.3.

The "physiology" of a mappings template is as follows:

1. The SOURCE of the API search entry point is generated based on the search service's basic uris. External variables should be used for the proper generation of the SOURCE
2. Once the remote resource is received (usually as JSON content), we
3. ITERATE through it. Iteration is carried out by specifying the place in the result containing the list of the results (sounds)
4. BIND a set of variables extracted from the original (JSON) result for each iteration

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

5. GENERATE, for each JSON, a set of RDF triples representing the result

It is also possible to use SPARQL-Generate Playground in order to practice mappings before integration: https://ci.mines-stetienne.fr/sparql-generate/playground.html

2.6.2.1. External variables

The template will receive the following external **variables/placeholders**:

- from the client's request
  - ***$pattern*** - search query
  - **$limit** - client's results limit (default 1)
  - ***$page*** - client's page (default 1)
- auto-generated
  - ***$startTime*** - start time of accessing the remote search service
- from the mediator's config file
  - ***$token*** - service's authorization key

2.6.3. Register in ".gitmodules" file

Register a new mappings repository inside the mediator github repository as a git module (inside the "*.gitmodules*" file)

Example:

```
[submodule "src/services/mediator/lib/mappings/freesound-to-audiocommons"]
    path = src/services/mediator/lib/mappings/freesound-to-audiocommons
    url = https://github.com/AudioCommons/semantic-mappings-freesound
```

2.6.4. Register inside the mediator's config file "mediaconf.yaml"

Register mappings inside the mediator's config file "*mediaconf.yaml*", under the ["mappings"] key:

```
 mappings:
   audioclips:
     search:
       freesound:
         file:
"lib/mappings/freesound-to-audiocommons/data-adapters/audio-search-by-text.rq"
         args: ["pattern", "token"]
     show:
       freesound:
         file:
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

```
"lib/mappings/freesound-to-audiocommons/data-adapters/audio-search-by-id.rq"
         args: ["trackId", "token"]

   collections:
     search:
       freesound:
         file:
"lib/mappings/freesound-to-audiocommons/data-adapters/pack-search-by-text.rq"
         args: ["trackId", "token"]
```

2.6.5. Provide the service uri and key

The URI and key should be placed under the **[“contentProviders”] key**.

Example:

```
contentProviders:
 keys:
   freesound: "2aL7UlMAoppS0oDzypXExHjv1PR4UekV7rT6NKQF"
 uris:
   freesound: "http://freesound.org"
```

NOTE: the key will be automatically (by *ConfigManager.py*) injected into the mappings template instead of the “*$token*” variable (placeholder)

2.7 A short introduction to data models

2.7.1 Description of basic AC ontology classes

The final output of the transformation conducted by SPARQL-Generate should conform with the Audio Commons Ontology. The ontology is developed in a public repository[1] and available for online consumption[2] (both by humans, as HTML documentation, and by machines, as OWL).

At the fig. 2.7.1 we can see a simplified graph model of a hypothetical search result regarding the single sound provider; Freesound in this example. We have 3 important ontology classes: ***ac:AudioCollection*** representing whole collection for a single sound provider, ***ac:AudioCollectionNode***

[1] https://github.com/AudioCommons/ac-ontology
[2] https://w3id.org/ac-ontology/aco

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N° 688382

representing the single sound entity, and finally the ***ac:AudioClip*** as a representation of real sound content. These are fundamental elements that each service integration should provide as a result of the mapping process through SPARQL-Generate.

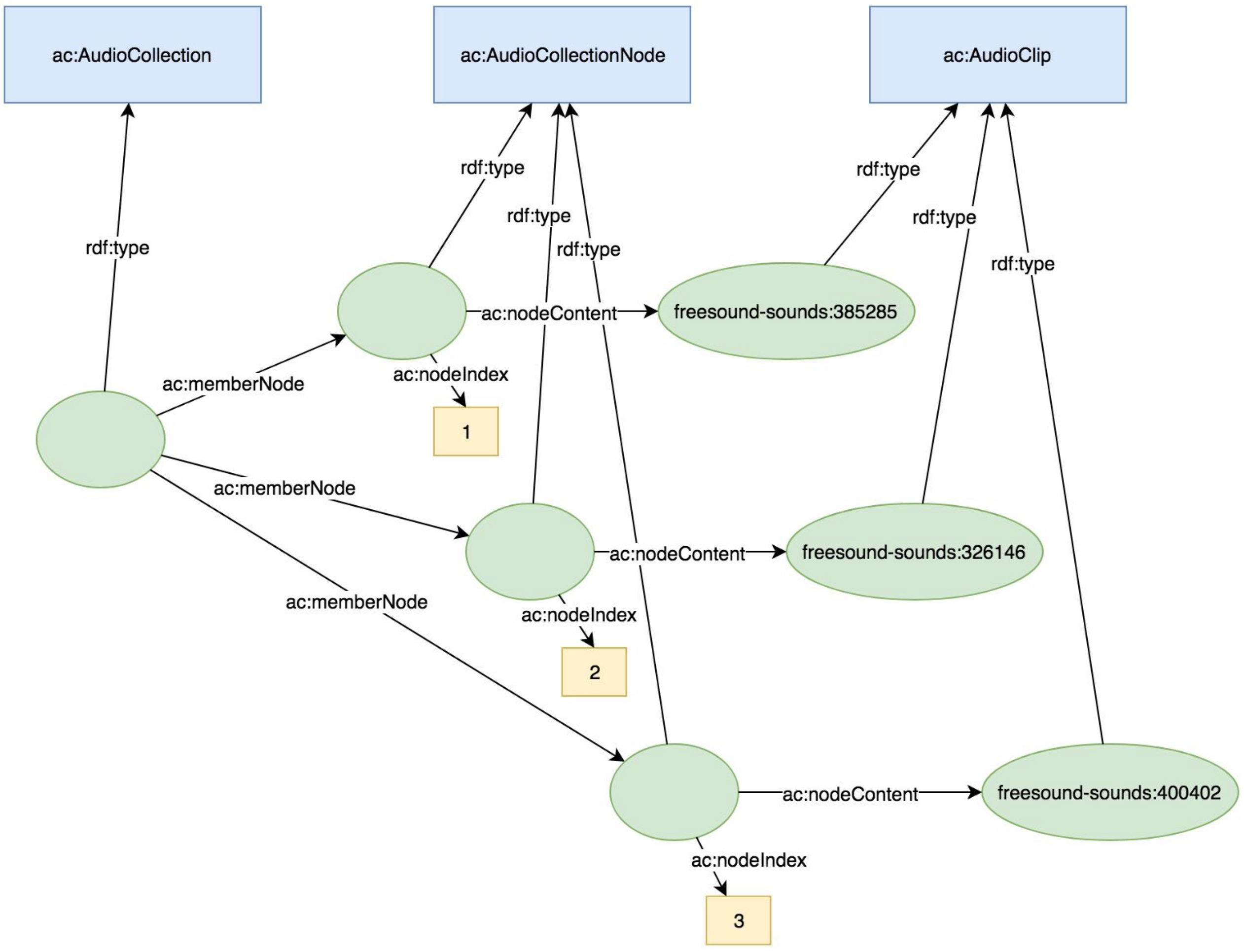


Figure 2.7.1 RDF graph response of search action following AC Ontology

In Fig. 2.7.2 and 2.7.3 we see examples of successful and failed search actions respectively. The mapping should provide either of these results according to the search status.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

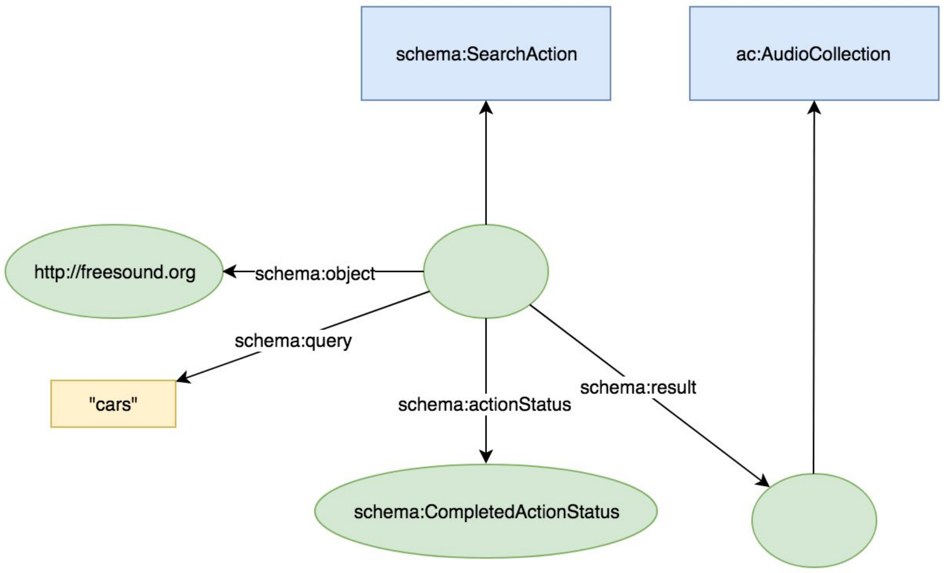


Figure 2.7.2 RDF graph response of search action success result

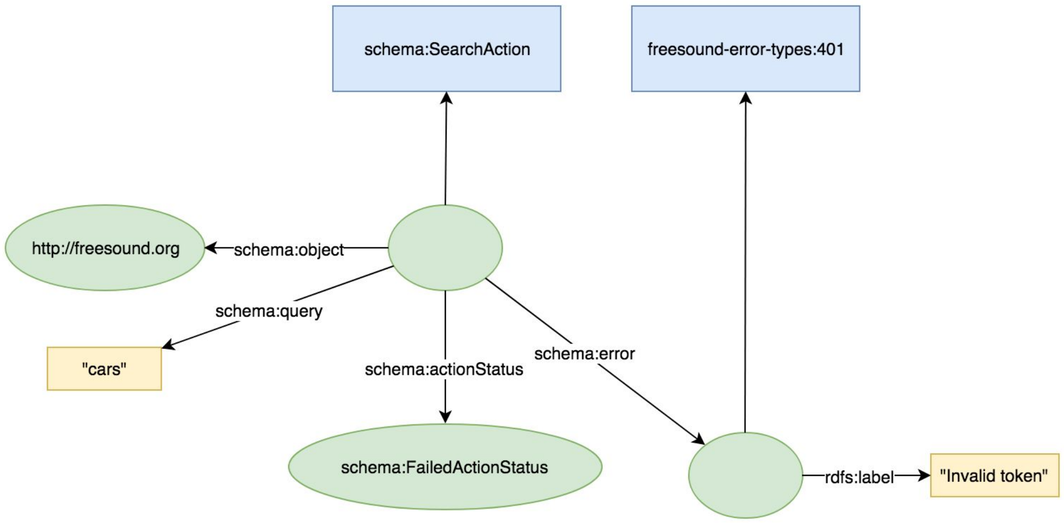
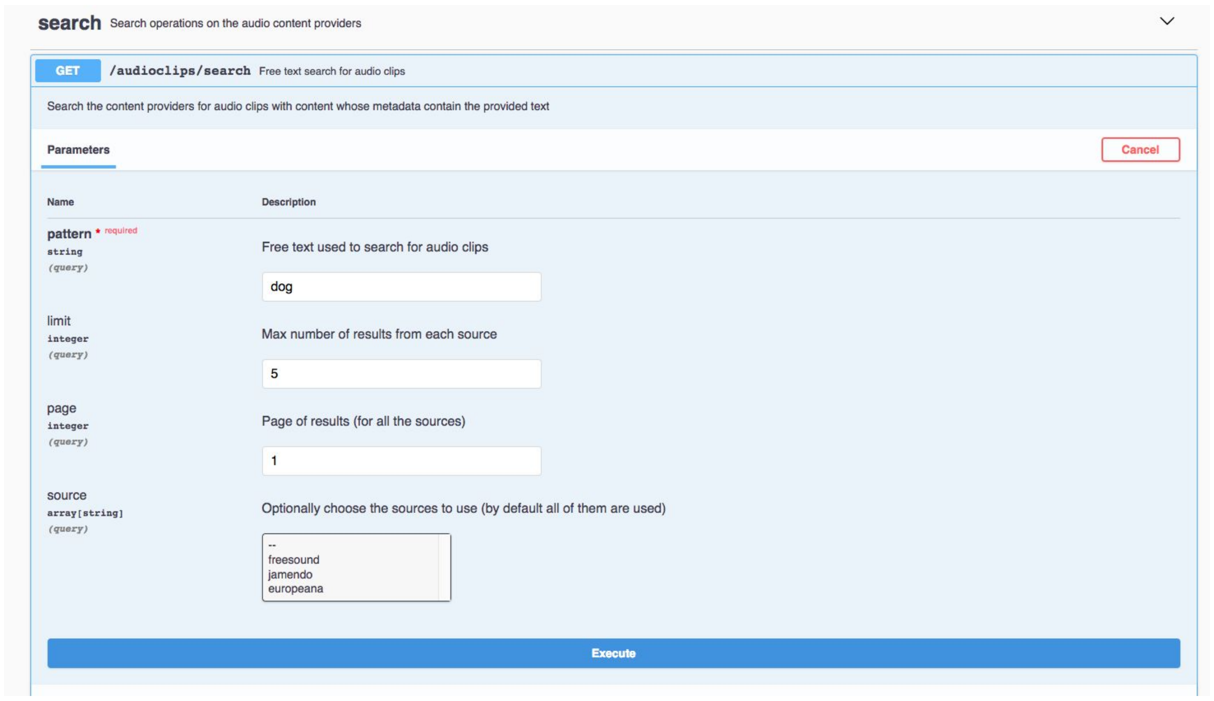


Figure 2.7.3 RDF graph response of search action error result

2.7.2 Real-world example

In order to practice real interactions with currently integrated search services, one can access the official mediator v2 documentation at the https://m2.audiocommons.org/. As part of documentation there is integrated a form for real interactions with services fig 2.7.4.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382


search Search operations on the audio content providers

GET **/audioclips/search** Free text search for audio clips

Search the content providers for audio clips with content whose metadata contain the provided text

Parameters

Name	Description
pattern * required string (query)	Free text used to search for audio clips: dog
limit integer (query)	Max number of results from each source: 5
page integer (query)	Page of results (for all the sources): 1
source array[string] (query)	Optionally choose the sources to use (by default all of them are used): -- freesound jamendo europeana

Figure 2.7.4 Online form for testing the existing service integrations

This search result would render into the following mediator v2 API call:

https://m2.audiocommons.org/api/audioclips/search?pattern=dog&limit=2&page=1&source=freesound

or in the case of running the API call from the *nix terminal:

```
curl -X GET \
"https://m2.audiocommons.org/api/audioclips/search?pattern=dog&limit=2&page=1&source=free
sound" -H "accept: application/json"
```

At tables 2.7.1 and 2.7.2 we can see examples of real results of mediator, initial RDF mapping and final JSON-LD result, retrospectively. We highlighted the skeleton and the most important aspect of the results (please consult the legend). In the Appendix we provide complete results.

LEGEND:

- **Newly introduced entities** are marked as **bold blue**
- **Comments** are marked as **bold red**
- **Important entities** are marked as **bold**
- **parts removed** can be found in the Appendix

NOTE: The order of triplets is changed for the sake of readability

```
# part removed
# Description of the search action
[ a  schema:SearchAction ;
 schema:actionStatus  schema:CompletedActionStatus ;
 schema:endTime      "2019-01-29T09:08:21.588+00:00"^^xsd:dateTime ;
 schema:object       <http://freesound.org> ;
 schema:query        "dog" ;
 schema:result       [ a  ac:AudioCollection ;
                ac:memberNode  [ a ac:AudioCollectionNode ;
                         ac:nodeContent  freesound-sounds:327666 ;
                         ac:nodeIndex    2
                        ] ;
                ac:memberNode  [ a ac:AudioCollectionNode ;
                         ac:nodeContent  freesound-sounds:392617 ;
                         ac:nodeIndex    1
                        ]
               ] ;
 schema:startTime    "2019-01-29T09:08:21.292695"^^xsd:dateTime
] .

# Description of Freesound as a sound provider
<http://freesound.org>
    a        foaf:Organization ;
    foaf:name  "Freesound" .

# Description of sound search results
# Description of the first sound
freesound-sounds:327666
    a  ac:AudioClip ;
    cc:license       <http://creativecommons.org/licenses/by-nc/3.0/> ;
    dc:description   "A furious dog barking.\r\n\r\n" ;
    dc:title        "Dog Bark.wav" ;
    ac:audioCategory  freesound-tags:barking , freesound-tags:woof , freesound-tags:intimidating ,
freesound-tags:breaking , freesound-tags:snarl , freesound-tags:aggressively # part removed;
    ac:author        freesound-users:Juan_Merie_Venter ;
    ac:availableAs   [ a    ac:AudioFile ;
                ebu:audioChannelNumber     2 ;
                ebu:bitRate              128000 ;
                ebu:hasAudioEncodingFormat  encoding_mp3: ;
                ebu:locator
<http://freesound.org/data/previews/327/327666_5632380-hq.mp3> ;
```

```
                        ebu:sampleRate          44100.0e0 ;
                        ebu:sampleSize          24
                      ] ;
        ac:availableAs    [ a ac:AudioFile ;
                        ebu:audioChannelNumber      2 ;
                        ebu:bitRate             192000 ;
                        ebu:hasAudioEncodingFormat  encoding_ogg: ;
                        ebu:locator
<http://freesound.org/data/previews/327/327666_5632380-hq.ogg> ;
                        ebu:sampleRate          44100.0e0 ;
                        ebu:sampleSize          24
                      ] ;
        # part removed
        ac:duration       6472.47e0 ;
        ac:image          [ a ebu:Picture ;
                        ebu:frameHeight  201 ;
                        ebu:frameWidth   900 ;
                        ebu:locator
<http://freesound.org/data/displays/327/327666_5632380_wave_L.png> ;
                        skos:prefLabel   "Waveform"
                      ] ;
        ac:image          [ a ebu:Picture ;
                        ebu:frameHeight  50 ;
                        ebu:frameWidth   50 ;
                        ebu:locator
<http://freesound.org/data/displays/327/327666_5632380_spec_M.jpg> ;
                        skos:prefLabel   "Spectrogram"
                      ] ;
        # part removed
        ac:originalFile   [ a ac:AudioFile ;
                        ebu:audioChannelNumber  2 ;
                        ebu:sampleRate        44100.0e0 ;
                        ebu:sampleSize        24
                      ] .

# Description of the second sound
freesound-sounds:392617
        a  ac:AudioClip ;
        cc:license       <http://creativecommons.org/publicdomain/zero/1.0/> ;
        dc:description    "Dog toy sound fx." ;
        dc:title          "Dog Toy" ;
        ac:audioCategory  freesound-tags:play , freesound-tags:funny # part removed;
        ac:author         freesound-users:ScreamStudio ;
        ac:availableAs           # part removed
        ac:duration       4704.31e0 ;
        ac:image                 # part removed
        ac:originalFile   [ a ac:AudioFile ;
                        ebu:audioChannelNumber  2 ;
                        ebu:sampleRate        44100.0e0 ;
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

```
                        ebu:sampleSize        24
                      ] .

# Description of a the 1st sound as an audio publication
[ a        ac:AudioPublication ;
  event:time                      [ a time:Instant , time:TemporalEntity ;
                             time:inXSDDateTime  "2015-11-02T19:07:34"^^xsd:dateTime
                           ] ;
  ac:publishedAudioManifestation  freesound-sounds:327666
] .

# Description of a the 2nd sound as an audio publication
[ a        ac:AudioPublication ;
  event:time                      [ a time:Instant , time:TemporalEntity ;
                             time:inXSDDateTime  "2017-05-14T16:07:22"^^xsd:dateTime
                           ] ;
  ac:publishedAudioManifestation  freesound-sounds:392617
] .
```

Table 2.7.1 Slimmed version of RDF result of SPARQL-Generate mapping for the provided search example

```
{
  "@id": "acActions:cd911353-13ab-473b-aaeb-798657ef778f",
  "@type": "schema:SearchAction",
  "actionStatus": "schema:CompletedActionStatus",
  "errors": [],
  "object": {
    "@id": "AC_API:v2.4.1",
    "@type": "doap:Version",
    "revision": "2.4.1"
  },
  "query": "dog",
  "results": [
    {
      "from": {
        "@type": "schema:SearchAction",
        "actionStatus": "schema:CompletedActionStatus",
        "endTime": "2019-01-29T09:08:21.588000+00:00",
        "object": "http://freesound.org",
        "startTime": "2019-01-29T09:08:21.292000+00:00"
      },
      "@type": "ac:AudioCollection",
      "members": [
        {
          "@type": "ac:AudioCollectionNode",
          "content": {
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

```
    "@id": "freesound-sounds:392617",
    "@type": "ac:AudioClip",
    "license": "http://creativecommons.org/publicdomain/zero/1.0/",
    "description": "Dog toy sound fx.",
    "title": "Dog Toy",
    "audioCategories": [
      "freesound-tags:animal",
      "freesound-tags:childs-toy"
      // part removed
    ],
    "author": "freesound-users:ScreamStudio",
    "availableAs": [
      {
        "@type": "ac:AudioFile",
        "audioChannelNumber": 2,
        "bitRate": 128000,
        "hasAudioEncodingFormat": "mp3:",
        "locator": "http://freesound.org/data/previews/392/392617_7383104-hq.mp3",
        "sampleRate": 44100,
        "sampleSize": 24
      }
      // part removed
    ],
    "duration": 4704.31,
    "images": [
      {
        "@type": "ebu:Picture",
        "frameHeight": 71,
        "frameWidth": 120,
        "locator": "http://freesound.org/data/displays/392/392617_7383104_wave_M.png",
        "prefLabel": "Waveform"
      }
      // part removed
    ],
    "originalFile": {
      "@type": "ac:AudioFile",
      "audioChannelNumber": 2,
      "sampleRate": 44100,
      "sampleSize": 24
    }
  },
  "index": 1
},
{
  "@type": "ac:AudioCollectionNode",
  "content": {
    "@id": "freesound-sounds:327666",
    "@type": "ac:AudioClip",
    "license": "http://creativecommons.org/licenses/by-nc/3.0/",
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

```
                "description": "A furious dog barking.\r\n\r\n",
                "title": "Dog Bark.wav",
                "audioCategories": [
                  // part removed
                ],
                "author": "freesound-users:Juan_Merie_Venter",
                "availableAs": [
                  // part removed
                ],
                "duration": 6472.47,
                "images": [
                  // part removed
                ],
                "originalFile": {
                  "@type": "ac:AudioFile",
                  "audioChannelNumber": 2,
                  "sampleRate": 44100,
                  "sampleSize": 24
                }
              },
              "index": 2
            }
          ]
        }
      ]
    }
```

Table 2.7.2 Slimmed version of JSON-LD result of mediator v2 sent to client for the provided search example

2.8 Scenarios for Testing and Evaluating Service Integration

There are a number of possible ways to test the new mechanism of service integration:

2.8.1 Isolated test of SPARQL-Generate templates

An integrator can test the isolated mapping templates at the [SPARQL-Generate playground]. The main difference would be that the template should remove any template variable provided through the mediator's injection mechanism, but rather inject/replace them with concrete test values. Apart of that, this should be a fully working mechanism for testing SPARQL-Generate mapping templates.

The reasons why this testing could still fail in a real use scenario, after it is integrated into the deployed mediator, are most likely to be wrong mappings, not following AC Ontology or missing the required RDF triples necessary for the proper integration of results.

To avoid those problems, one needs to test mapping in a real environment, using one of the scenarios described in the following sections.

2.8.2 Providing mapping templates to ACE maintainers

After it has passed 2.8.1 testing, the integrator is welcome to provide mappings and all the necessary meta-data (described under 2.6) to ACE maintainers. After that, they should be able to integrate support for the service and let it be tested in the real environment.

If this solution is not viable, if there are issues with testing or if integrator wants to deploy their own infrastructure, it is possible to do that as well.

2.8.3 Deploying one's own AC mediator ecosystem

The integrator is welcome to deploy their own AC mediator ecosystem and have full control over the integration and evaluation processes. Documentation is provided at the mediator v2's repository on the process of deployment[3].

2.8.4. Integration evaluation

The integrator is able to conduct an additional evaluation of the integration through the online dashboard that supports a view on internal workflows and actions they consist of, as per Fig. 2.8.1. The mediator supports sessions, and the integrator is able to test each separate session, nailing down the problem, and isolating other clients accessing the mediator.

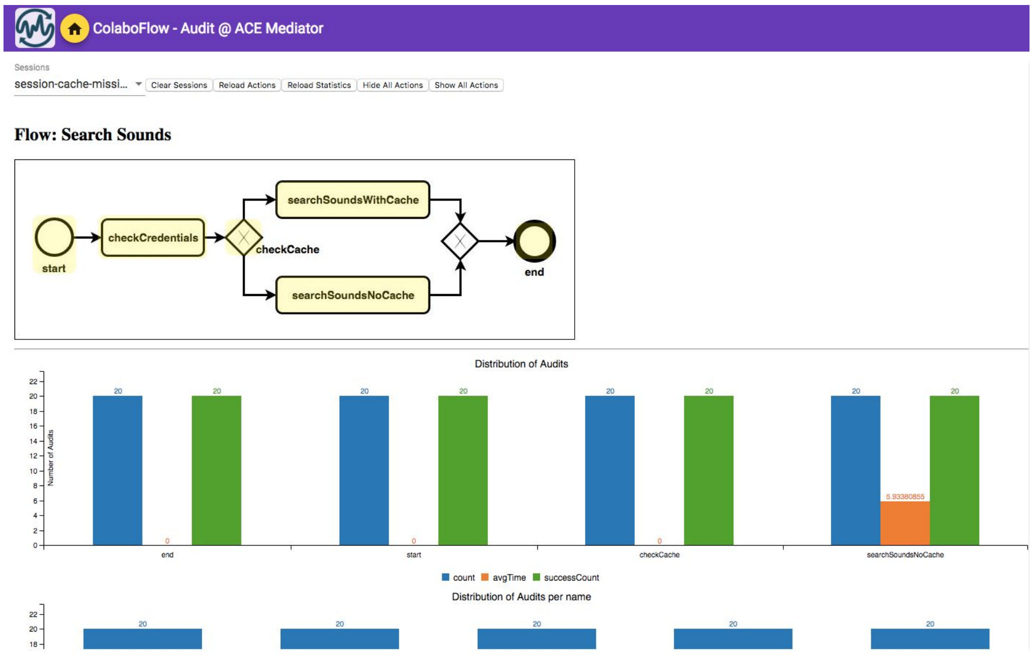


Figure 2.8.1. Example of the mediator's "*Search Sounds*" *workflow analysis with its* actions stats

[3] https://github.com/AudioCommons/semanticMediator/deployment.md

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

In this way, it is possible both to test the real-time efficiency of mapping, and also to understand separate border use-cases. The integrator can also run separate use-cases tests in separate sessions and in this way run comparative analyses.

If the 2.8.3 scenario is chosen, evaluation should be possible with two comparative mappings registered as separate search services (essentially, versions). Afterwards, using the mediator's support to search only through a single search service, the evaluator can address a particular search service mapping and do similar comparative analysis of mappings.

The audits are available inside the dashboard at the following address:
https://m2.audiocommons.org/dashboard/colaboflow-audits

2.9. Open Challenges

Semantic version of mediator v2, successfully demonstrated solution for descriptive service integration. We have 3 services integrated Freesound, Jamendo and Europeana, and did initial experiments with the Internet Archive as part of the external evaluation process. Through this process we both evaluated AC Ontology as feasible, but also evaluated real-world use-cases.

At the same time we still have efficiency issues of the whole mediator v2 ecosystem, mostly nailing down to the SPARQL-Generate service (fig 2.5.4.) which takes long processing time to map the results from original JSON results into RDF results. Utilizing multiple SPARQL-Generate service instances with load-balancing we can reduce the impact as seen on tests with multiple concurrent mediator access.

Figures 2.9.1. and 2.9.2 comparatively show time response improvements after introducing load-balancing [ColaboFlow] with the main impact on reducing average execution time of the last action ***searchSoundsNoCache***.

Here, we will present two use-cases addressing the current semantic mediator implementation and the challenges it is facing. We will also provide possible solutions implemented as early-stage prototypes.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

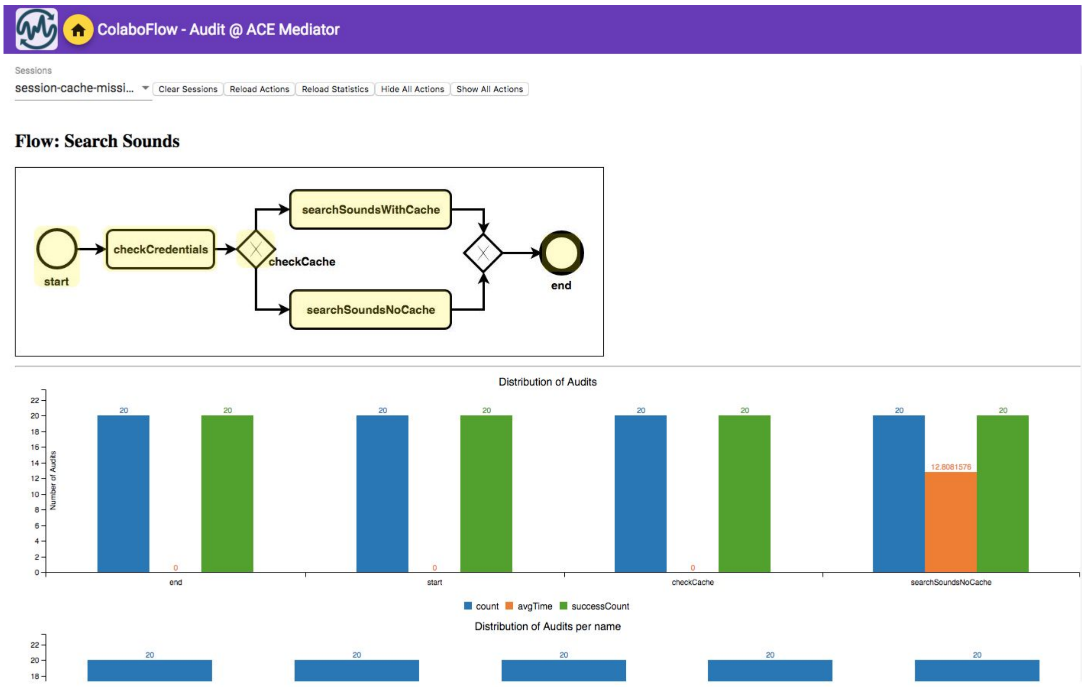


Figure 2.9.1. Example of the mediator's "*Search Sounds*" *workflow analysis with its* actions stats

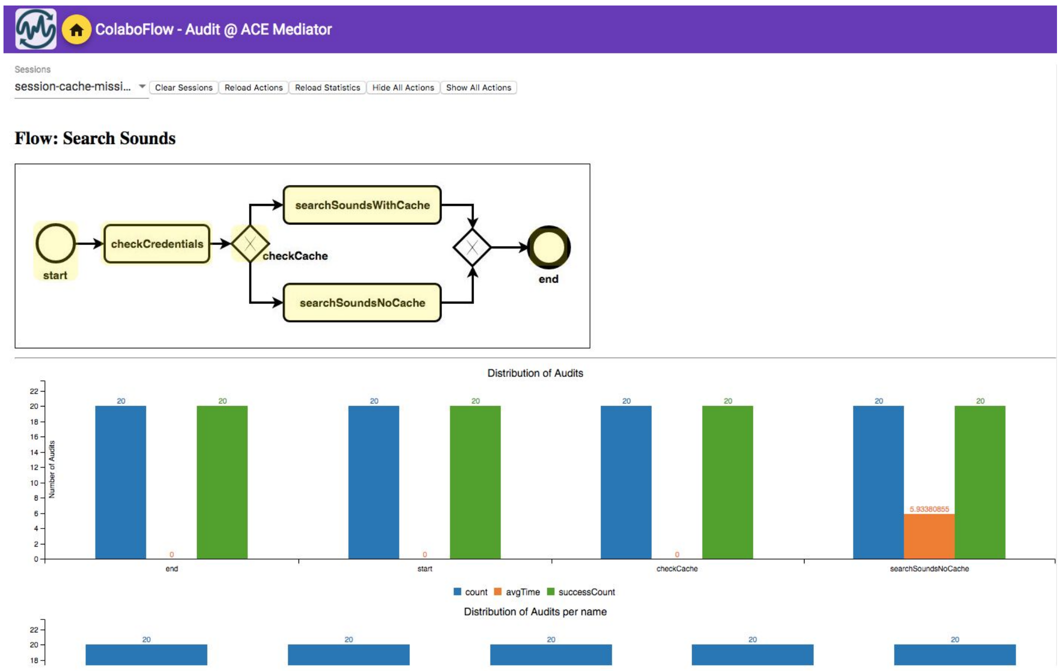


Figure 2.9.2. Example of the mediator's "*Search Sounds*" *workflow analysis with its* actions stats

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

2.9.1. Sequential extension of the mediator search workflow

During the “Audio Commons for socially engaged workshops” (D6.12) we noticed that particular words we were searching sounds for, were underrepresented, and search results were basically useless. That was an interesting trigger on the integration question if and how we can implement “synonyms extension” service that would extend the initial input query sent from the client with word(s) synonyms. At the fig 2.5.4. we see a first example of the use case where we want to extend the mediator with a service that provides support for synonyms for the provided search word.

What we have found is that, although we support declarative mapping as a service integration mechanism, it is localized on the sounds search domain, and even if we technically are able to inject the call to the “synonyms extension” service as a part of mapping template, it would require doing it for each service, issuing unnecessary multiple service access and integrity issues. Clearly we need to place the call to that service earlier in the pipeline of data transformation, closer to the beginning of the search workflow. At the fig 2.5.4. we see the call to the “synonyms extension” service at very beginning of the workflow aiming for supplementing the original search query with the extended version of the search query.

From the perspective of control and data flows this is appropriate solution. However, what we have learned from this use case is that we “rolled-back” to hardcoding service integration into mediator v2, as with mediator v1.

2.9.2 Parallel extension of the the mediator search workflow

The second real use-case and challenge is integration of the service for **annotation service** or more precisely ***automatic semantic description of music pieces*** (described in the deliverable D4.11). In this use-case, client would be able to 1) provide additional query parameters to specify filtering across specific semantic music pieces’ descriptors, or to 2) get result semantically enriched with additional semantic descriptors (like chords, key, etc).

Specific aspects of the service is in its long processing time compared to searching time. This makes it unacceptable to postpone the result waiting for the service to provide the results. Additionally, in some use-cases, the service cannot provide results, so this also need to be taken into consideration on the client implementation. Opposite to the 2.9.1 use-case where we had sequential extension of the workflow, here we are extending it in **the parallel manner**; by adding additional RDF predicates to each key entity; sound results in this case. In that sense, considering RDF nature of constructing complex objects, this is purely additive operation, that is unobtrusive, regarding the standard mediator v2 workflow, which makes implementation simpler and less hardcoded. However, we still need coordination between the mediator and annotation service. We also need to provide pub-sub pattern for client registration on annotation service response. We can see at the fig. 2.9.2.1 possible extension of the mediator v2 infrastructure presented at the fig. 2.5.3.

We are utilizing SEPA service to provide Pub-Sup pattern to semantic client on-behalf of the annotation service. It is important to realize that we HAVE to modify mediator to recognize and coordinate with this additional, parallel, extension of the workflow. Obviously, the complexity of the solution is not ideal, we need to have hardcoded extensions of standard workflow, which is still supporting just this particular use-case and way to understand data and control flow across the mediator infrastructure gets harder and harder.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N° 688382

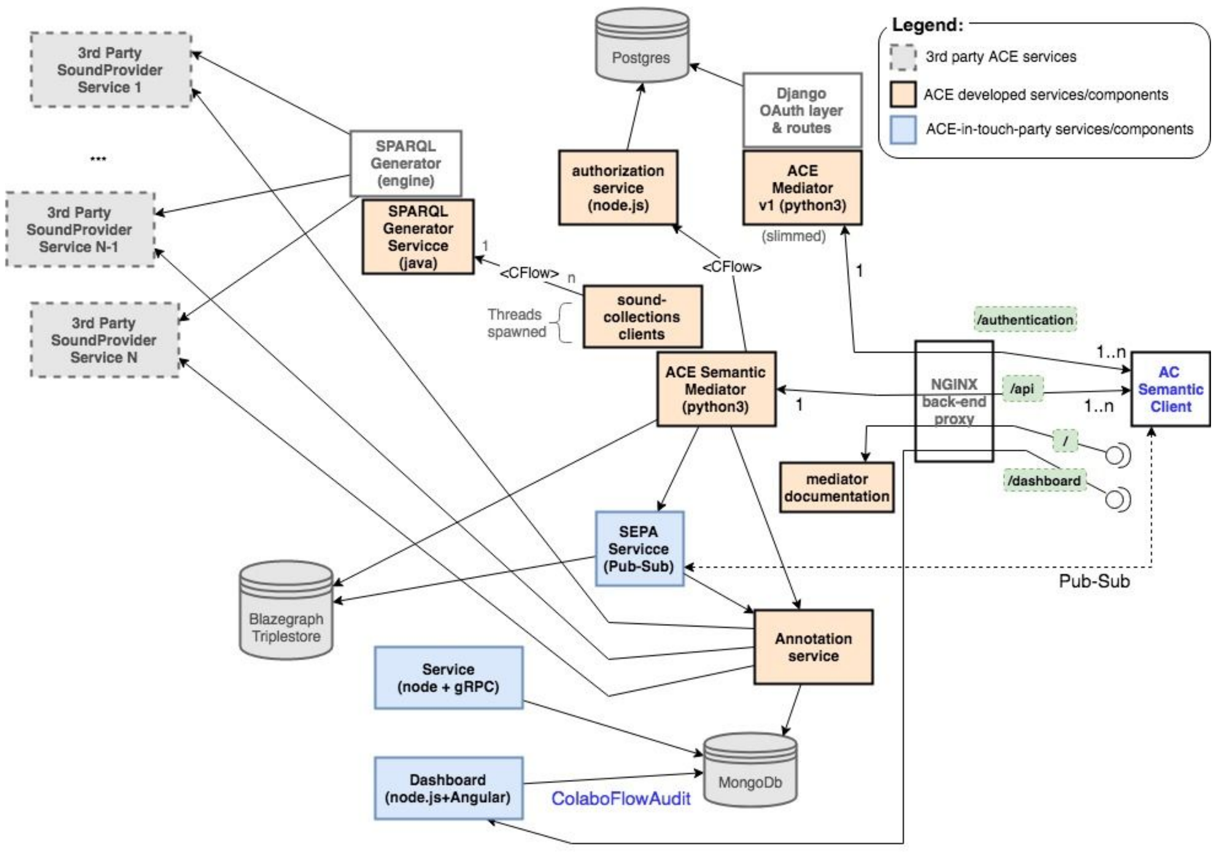

3rd Party SoundProvider Service 1
...
3rd Party SoundProvider Service N-1
3rd Party SoundProvider Service N
SPARQL Generator (engine)
SPARQL Generator Servicce (java)
Postgres
Django OAuth layer & routes
authorization service (node.js)
ACE Mediator v1 (python3)
(slimmed)
Legend:
3rd party ACE services
ACE developed services/components
ACE-in-touch-party services/components
<CFlow>
n
1
Threads spawned
sound-collections clients
ACE Semantic Mediator (python3)
/authentication
1..n
NGINX back-end proxy
/api
AC Semantic Client
/
/dashboard
mediator documentation
SEPA Servicce (Pub-Sub)
Pub-Sub
Blazegraph Triplestore
Annotation service
Service (node + gRPC)
Dashboard (node.js+Angular)
ColaboFlowAudit
MongoDb

Figure 2.9.2.1. Extension of the semantic mediator (v2) infrastructure

2.9.3 Prototyped Solution

The most natural need would be to **flatten** the mediator v2 infrastructure and therefore reduce its complexity. This can be possible if we can **externalize** the orchestration of services and provide the services to the orchestrator in a direct-addressable way.

This is how we approached the problem and came to the model presented at the fig. 2.9.3.1 where we have a direct-addressable flat array of service accessed from the external orchestrator and orchestrated through the data+control workflows using the ColaboFlow solution [ColaboFlow]. We see example of two workflows, ***search*** and ***authentication*** workflows.

In such a way we have managed to support both sequential and parallel extension of the semantic mediator workflow and elasticity of services (though for scaling number of services, we need an external support). We are also able to trace and evaluate mediator requests in controlled and isolated way, by providing separate flow instances (each separate request) sitting under separate sessions (for example, one is regular session, and 2nd one is one used for evaluation, extension or testing).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

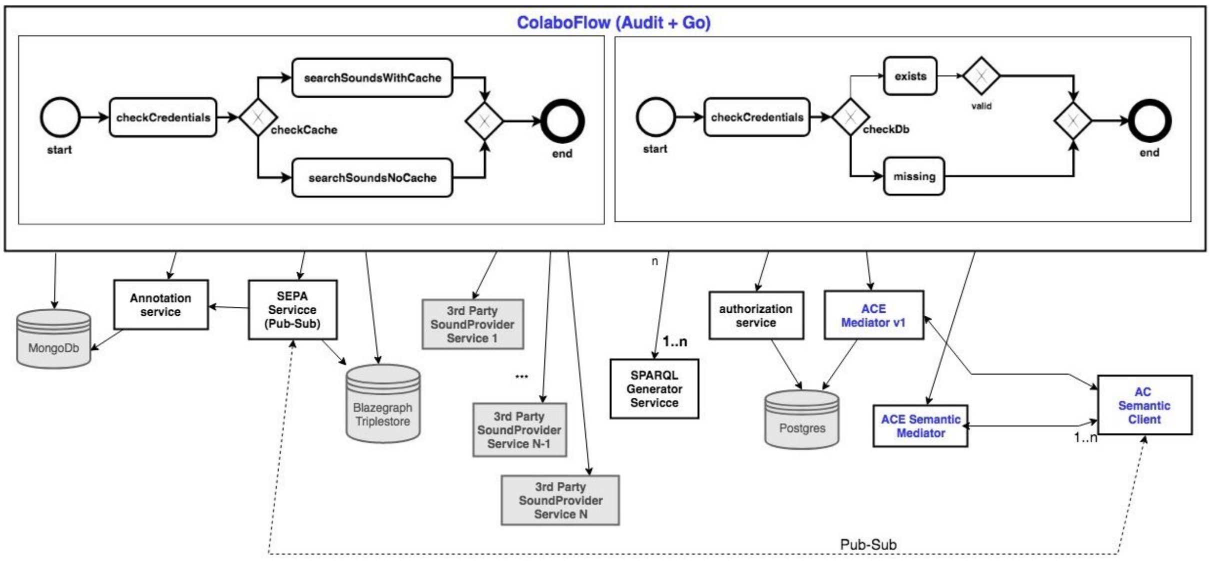


Figure 2.9.3.1 Flattening the infrastructure with externalized descriptive workflows

Additional benefits: by externalizing workflows we can even **hot-swap** integrated services and change the behavior of mediator. We can have business logic that is client or use-case based (i.e. if client request semantic annotation or synonyms extension, etc). We can safely and concurrently test new business logic, by using both auditory and control aspects of workflows we achieved both no-cost logging and process-mining; tracking behavioral patterns of semantic clients and users.

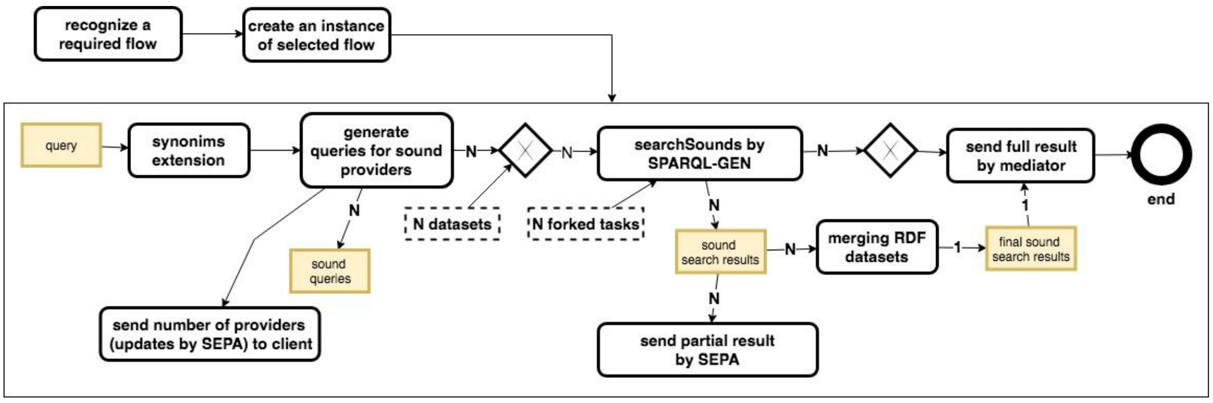

recognize a required flow
create an instance of selected flow
query
synonims extension
generate queries for sound providers
N
N
searchSounds by SPARQL-GEN
N
send full result by mediator
end
N
sound queries
N datasets
N forked tasks
N
sound search results
N
merging RDF datasets
1
final sound search results
1
send number of providers (updates by SEPA) to client
N
send partial result by SEPA

Figure 2.9.3.2 representation of the search flow in the new workflow design pattern of the semantic mediator

3 Conclusion

This document provides guidelines for a search service integration for new v2 mediator. Unlike mediator v1, SAs are not separate (Python code) implementations, but rather a single component that is descriptively driven to adopt to each unique search service. This is possible as the mediator v2 is implemented following a semantic technology and based on the AudioCommons Ontology.

To integrate a search service in the mediator v2, the integrator needs to do it descriptively, using the SPARQL-Generate language, an extension of the SPARQL-query language. The descriptor/mappings template aims to translate the JSON response into an RDF response conforming to the Audio Commons ontology.

Some of the benefits of this approach are that the integrators do not need to get into the technical implementation of the mediator, there is no need for rebuilding and redeploying the mediator, but rather services' integration can be done in a hot-swap manner. Additionally, the whole ACE infrastructure becomes more resilient to malicious code of service integration implementation. This reduces risks and allows for safer and more open politics of federating other services.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

4 References

[Del2.2] Draft ontology specification

[Del2.5] Service integration technologies

[CreativeCommons] Creative Commons - https://creativecommons.org/

[SEPA] SEPA description: http://mml.arces.unibo.it/TR/sepa.html

[mediator-v1-authentication] the official documentation of the authentication workflows and API in the mediator v1
https://m.audiocommons.org/docs/api_overview.html#authentication-in-third-party-services

https://m.audiocommons.org/docs/api_authentication.html

[SPARQL-query-language] SPARQL query language specification https://www.w3.org/TR/2013/REC-sparql11-query-20130321/

[SPARQL-Generate-component] Description of the component implementing SPARQL-Generate extension https://ci.mines-stetienne.fr/sparql-generate/

[SPARQL-Generate-language] Overview of the SPARQL-Generate language https://ci.mines-stetienne.fr/sparql-generate/language.html

[SPARQL-Generate-extension] Description of SPARQL extension: SPARQL-Generate https://ci.mines-stetienne.fr/sparql-generate/language.html

[SPARQL-Generate playground] SPARQL-Generate playground for testing queries in SPARQL-Generate language: https://ci.mines-stetienne.fr/sparql-generate/playground.html

[Colabo] Colabo.Space introduction: http://colabo.space

[ColaboFlow] Colabo.Space introduction: http://colabo.space/flow

[gRPC] gRPC concepts https://grpc.io/docs/guides/concepts.html

[ProtoBuff] Protocol Buffers overview https://developers.google.com/protocol-buffers/docs/overview

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N° 688382

5. APPENDIX

5.1 Access Info

```
curl -X GET \
"https://m2.audiocommons.org/api/audioclips/search?pattern=dog&limit=2&page=1&source=free
sound" -H "accept: application/json"
```

Table 5.2.

5.2. SPARQL-Generate Response

LEGEND:

- **Newly introduced entities** are marked as **bold blue**
- **Comments** are marked as **bold red**
- **Important entities** are marked as **bold**

NOTE: The order of triplets is changed for the sake of readability.

```
@base        <http://example.org/> .
@prefix cc:   <http://creativecommons.org/ns#> .
@prefix freesound-api-packs: <http://freesound.org/apiv2/packs/> .
@prefix schema: <http://schema.org/> .
@prefix ac:   <https://w3id.org/ac-ontology/aco#> .
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#> .
@prefix fn:   <http://w3id.org/sparql-generate/fn/> .
@prefix iter: <http://w3id.org/sparql-generate/iter/> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix freesound-users: <https://w3id.org/audiocommons/services/freesound/users/> .
@prefix ma:   <http://www.w3.org/ns/ma-ont#> .
@prefix freesound-packs: <https://w3id.org/audiocommons/services/freesound/packs/> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix encoding_flac:
<http://www.ebu.ch/metadata/ontologies/skos/ebu_AudioCompressionCodeCS#_22> .
@prefix ebu:  <http://www.ebu.ch/metadata/ontologies/ebucore/ebucore#> .
@prefix freesound-api: <http://freesound.org/apiv2/> .
@prefix time: <https://www.w3.org/2006/time#> .
@prefix encoding_mp3:
<http://www.ebu.ch/metadata/ontologies/skos/ebu_AudioCompressionCodeCS#_8.4> .
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

```
@prefix encoding_ogg:
<http://www.ebu.ch/metadata/ontologies/skos/ebu_AudioCompressionCodeCS#_26> .
@prefix event: <http://purl.org/NET/c4dm/event.owl#> .
@prefix freesound-sounds: <https://w3id.org/audiocommons/services/freesound/sounds/> .
@prefix prov:  <http://www.w3.org/ns/prov#> .
@prefix freesound-tags: <https://w3id.org/audiocommons/services/freesound/tags/> .
@prefix foaf:  <http://xmlns.com/foaf/0.1/> .
@prefix dc:    <http://purl.org/dc/elements/1.1/> .
@prefix encoding_wav:
<http://www.ebu.ch/metadata/ontologies/skos/ebu_AudioCompressionCodeCS#_28> .

# Description of the search action
[ a  schema:SearchAction ;
 schema:actionStatus  schema:CompletedActionStatus ;
 schema:endTime       "2019-01-29T09:08:21.588+00:00"^^xsd:dateTime ;
 schema:object       <http://freesound.org> ;
 schema:query        "dog" ;
 schema:result       [ a            ac:AudioCollection ;
                 ac:memberNode  [ a ac:AudioCollectionNode ;
                         ac:nodeContent  freesound-sounds:327666 ;
                         ac:nodeIndex    2
                        ] ;
                 ac:memberNode  [ a ac:AudioCollectionNode ;
                         ac:nodeContent  freesound-sounds:392617 ;
                         ac:nodeIndex    1
                        ]
                ] ;
 schema:startTime     "2019-01-29T09:08:21.292695"^^xsd:dateTime
] .

# Description of Freesound as a sound provider
<http://freesound.org>
      a         foaf:Organization ;
      foaf:name  "Freesound" .

# Description of sound search results
# Description of the first sound
freesound-sounds:327666
      a  ac:AudioClip ;
      cc:license       <http://creativecommons.org/licenses/by-nc/3.0/> ;
      dc:description    "A furious dog barking.\r\n\r\n" ;
      dc:title         "Dog Bark.wav" ;
      ac:audioCategory  freesound-tags:barking , freesound-tags:woof , freesound-tags:intimidating ,
freesound-tags:breaking , freesound-tags:snarl , freesound-tags:aggressively , freesound-tags:shout
, freesound-tags:yap , freesound-tags:yell , freesound-tags:bay , freesound-tags:angrily ,
freesound-tags:snap , freesound-tags:yelp , freesound-tags:fierce , freesound-tags:growl ,
freesound-tags:dog , freesound-tags:big , freesound-tags:cry , freesound-tags:OWI ,
freesound-tags:dogs , freesound-tags:bawl , freesound-tags:bark , freesound-tags:aggressive ,
freesound-tags:barked , freesound-tags:thunder , freesound-tags:roar , freesound-tags:Dog ,
freesound-tags:bellow ;
        ac:author         freesound-users:Juan_Merie_Venter ;
        ac:availableAs    [ a     ac:AudioFile ;
                          ebu:audioChannelNumber      2 ;
                          ebu:bitRate               128000 ;
                          ebu:hasAudioEncodingFormat  encoding_mp3: ;
                          ebu:locator
<http://freesound.org/data/previews/327/327666_5632380-hq.mp3> ;
                          ebu:sampleRate            44100.0e0 ;
                          ebu:sampleSize            24
                        ] ;
        ac:availableAs    [ a     ac:AudioFile ;
                          ebu:audioChannelNumber      2 ;
                          ebu:bitRate               192000 ;
                          ebu:hasAudioEncodingFormat  encoding_ogg: ;
                          ebu:locator
<http://freesound.org/data/previews/327/327666_5632380-hq.ogg> ;
                          ebu:sampleRate            44100.0e0 ;
                          ebu:sampleSize            24
                        ] ;
        ac:availableAs    [ a     ac:AudioFile ;
                          ebu:audioChannelNumber      2 ;
                          ebu:bitRate               80000 ;
                          ebu:hasAudioEncodingFormat  encoding_ogg: ;
                          ebu:locator
<http://freesound.org/data/previews/327/327666_5632380-lq.ogg> ;
                          ebu:sampleRate            44100.0e0 ;
                          ebu:sampleSize            24
                        ] ;
        ac:availableAs    [ a     ac:AudioFile ;
                          ebu:audioChannelNumber      2 ;
                          ebu:bitRate               64000 ;
                          ebu:hasAudioEncodingFormat  encoding_mp3: ;
                          ebu:locator
<http://freesound.org/data/previews/327/327666_5632380-lq.mp3> ;
                          ebu:sampleRate            44100.0e0 ;
                          ebu:sampleSize            24
                        ] ;
        ac:duration       6472.47e0 ;
        ac:image          [ a ebu:Picture ;
                          ebu:frameHeight  201 ;
                          ebu:frameWidth   900 ;
                          ebu:locator
<http://freesound.org/data/displays/327/327666_5632380_wave_L.png> ;
                          skos:prefLabel   "Waveform"
                        ] ;
        ac:image          [ a ebu:Picture ;
                          ebu:frameHeight  50 ;
                          ebu:frameWidth   50 ;
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

```
                    ebu:locator
<http://freesound.org/data/displays/327/327666_5632380_spec_M.jpg> ;
                    skos:prefLabel  "Spectrogram"
                  ] ;
      ac:image       [ a ebu:Picture ;
                    ebu:frameHeight  201 ;
                    ebu:frameWidth   900 ;
                    ebu:locator
<http://freesound.org/data/displays/327/327666_5632380_spec_L.jpg> ;
                    skos:prefLabel  "Spectrogram"
                  ] ;
      ac:image       [ a ebu:Picture ;
                    ebu:frameHeight  71 ;
                    ebu:frameWidth   120 ;
                    ebu:locator
<http://freesound.org/data/displays/327/327666_5632380_wave_M.png> ;
                    skos:prefLabel  "Waveform"
                  ] ;
      ac:originalFile  [ a ac:AudioFile ;
                    ebu:audioChannelNumber  2 ;
                    ebu:sampleRate        44100.0e0 ;
                    ebu:sampleSize        24
                  ] .

# Description of the second sound
freesound-sounds:392617
      a  ac:AudioClip ;
      cc:license       <http://creativecommons.org/publicdomain/zero/1.0/> ;
      dc:description    "Dog toy sound fx." ;
      dc:title         "Dog Toy" ;
      ac:audioCategory  freesound-tags:play , freesound-tags:funny , freesound-tags:childs-toy ,
freesound-tags:dog , freesound-tags:toy , freesound-tags:whine , freesound-tags:animal ,
freesound-tags:cat , freesound-tags:squeal , freesound-tags:Dog ;
      ac:author        freesound-users:ScreamStudio ;
      ac:availableAs    [ a     ac:AudioFile ;
                    ebu:audioChannelNumber     2 ;
                    ebu:bitRate              192000 ;
                    ebu:hasAudioEncodingFormat  encoding_ogg: ;
                    ebu:locator
<http://freesound.org/data/previews/392/392617_7383104-hq.ogg> ;
                    ebu:sampleRate            44100.0e0 ;
                    ebu:sampleSize            24
                  ] ;
      ac:availableAs    [ a     ac:AudioFile ;
                    ebu:audioChannelNumber     2 ;
                    ebu:bitRate              64000 ;
                    ebu:hasAudioEncodingFormat  encoding_mp3: ;
                    ebu:locator
<http://freesound.org/data/previews/392/392617_7383104-lq.mp3> ;
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

```
                        ebu:sampleRate          44100.0e0 ;
                        ebu:sampleSize          24
                      ] ;
        ac:availableAs    [ a     ac:AudioFile ;
                        ebu:audioChannelNumber     2 ;
                        ebu:bitRate             128000 ;
                        ebu:hasAudioEncodingFormat  encoding_mp3: ;
                        ebu:locator
<http://freesound.org/data/previews/392/392617_7383104-hq.mp3> ;
                        ebu:sampleRate          44100.0e0 ;
                        ebu:sampleSize          24
                      ] ;
        ac:availableAs    [ a     ac:AudioFile ;
                        ebu:audioChannelNumber     2 ;
                        ebu:bitRate             80000 ;
                        ebu:hasAudioEncodingFormat  encoding_ogg: ;
                        ebu:locator
<http://freesound.org/data/previews/392/392617_7383104-lq.ogg> ;
                        ebu:sampleRate          44100.0e0 ;
                        ebu:sampleSize          24
                      ] ;
        ac:duration       4704.31e0 ;
        ac:image          [ a ebu:Picture ;
                        ebu:frameHeight  201 ;
                        ebu:frameWidth   900 ;
                        ebu:locator
<http://freesound.org/data/displays/392/392617_7383104_wave_L.png> ;
                        skos:prefLabel   "Waveform"
                      ] ;
        ac:image          [ a ebu:Picture ;
                        ebu:frameHeight  201 ;
                        ebu:frameWidth   900 ;
                        ebu:locator
<http://freesound.org/data/displays/392/392617_7383104_spec_L.jpg> ;
                        skos:prefLabel   "Spectrogram"
                      ] ;
        ac:image          [ a ebu:Picture ;
                        ebu:frameHeight  71 ;
                        ebu:frameWidth   120 ;
                        ebu:locator
<http://freesound.org/data/displays/392/392617_7383104_wave_M.png> ;
                        skos:prefLabel   "Waveform"
                      ] ;
        ac:image          [ a ebu:Picture ;
                        ebu:frameHeight  50 ;
                        ebu:frameWidth   50 ;
                        ebu:locator
<http://freesound.org/data/displays/392/392617_7383104_spec_M.jpg> ;
                        skos:prefLabel   "Spectrogram"
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

```
                    ] ;
      ac:originalFile   [ a ac:AudioFile ;
                         ebu:audioChannelNumber  2 ;
                         ebu:sampleRate        44100.0e0 ;
                         ebu:sampleSize        24
                    ] .

# Description of a the 1st sound as an audio publication
[ a        ac:AudioPublication ;
  event:time                 [ a time:Instant , time:TemporalEntity ;
                              time:inXSDDateTime  "2015-11-02T19:07:34"^^xsd:dateTime
                            ] ;
  ac:publishedAudioManifestation  freesound-sounds:327666
] .

# Description of a the 2nd sound as an audio publication
[ a        ac:AudioPublication ;
  event:time                 [ a time:Instant , time:TemporalEntity ;
                              time:inXSDDateTime  "2017-05-14T16:07:22"^^xsd:dateTime
                            ] ;
  ac:publishedAudioManifestation  freesound-sounds:392617
] .
```

5.3. JSON-LD response from mediator

```
{
  "@id": "acActions:cd911353-13ab-473b-aaeb-798657ef778f",
  "@type": "schema:SearchAction",
  "actionStatus": "schema:CompletedActionStatus",
  "errors": [],
  "object": {
    "@id": "AC_API:v2.4.1",
    "@type": "doap:Version",
    "revision": "2.4.1"
  },
  "query": "dog",
  "results": [
    {
      "from": {
        "@type": "schema:SearchAction",
        "actionStatus": "schema:CompletedActionStatus",
        "endTime": "2019-01-29T09:08:21.588000+00:00",
        "object": "http://freesound.org",
        "startTime": "2019-01-29T09:08:21.292000+00:00"
      },
      "@type": "ac:AudioCollection",
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

```
"members": [
  {
    "@type": "ac:AudioCollectionNode",
    "content": {
      "@id": "freesound-sounds:392617",
      "@type": "ac:AudioClip",
      "license": "http://creativecommons.org/publicdomain/zero/1.0/",
      "description": "Dog toy sound fx.",
      "title": "Dog Toy",
      "audioCategories": [
        "freesound-tags:animal",
        "freesound-tags:childs-toy",
        "freesound-tags:funny",
        "freesound-tags:cat",
        "freesound-tags:play",
        "freesound-tags:whine",
        "freesound-tags:squeal",
        "freesound-tags:toy",
        "freesound-tags:Dog",
        "freesound-tags:dog"
      ],
      "author": "freesound-users:ScreamStudio",
      "availableAs": [
        {
          "@type": "ac:AudioFile",
          "audioChannelNumber": 2,
          "bitRate": 128000,
          "hasAudioEncodingFormat": "mp3:",
          "locator": "http://freesound.org/data/previews/392/392617_7383104-hq.mp3",
          "sampleRate": 44100,
          "sampleSize": 24
        },
        {
          "@type": "ac:AudioFile",
          "audioChannelNumber": 2,
          "bitRate": 64000,
          "hasAudioEncodingFormat": "mp3:",
          "locator": "http://freesound.org/data/previews/392/392617_7383104-lq.mp3",
          "sampleRate": 44100,
          "sampleSize": 24
        },
        {
          "@type": "ac:AudioFile",
          "audioChannelNumber": 2,
          "bitRate": 80000,
          "hasAudioEncodingFormat": "ogg:",
          "locator": "http://freesound.org/data/previews/392/392617_7383104-lq.ogg",
          "sampleRate": 44100,
          "sampleSize": 24
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

```
        },
        {
          "@type": "ac:AudioFile",
          "audioChannelNumber": 2,
          "bitRate": 192000,
          "hasAudioEncodingFormat": "ogg:",
          "locator": "http://freesound.org/data/previews/392/392617_7383104-hq.ogg",
          "sampleRate": 44100,
          "sampleSize": 24
        }
      ],
      "duration": 4704.31,
      "images": [
        {
          "@type": "ebu:Picture",
          "frameHeight": 71,
          "frameWidth": 120,
          "locator": "http://freesound.org/data/displays/392/392617_7383104_wave_M.png",
          "prefLabel": "Waveform"
        },
        {
          "@type": "ebu:Picture",
          "frameHeight": 50,
          "frameWidth": 50,
          "locator": "http://freesound.org/data/displays/392/392617_7383104_spec_M.jpg",
          "prefLabel": "Spectrogram"
        },
        {
          "@type": "ebu:Picture",
          "frameHeight": 201,
          "frameWidth": 900,
          "locator": "http://freesound.org/data/displays/392/392617_7383104_wave_L.png",
          "prefLabel": "Waveform"
        },
        {
          "@type": "ebu:Picture",
          "frameHeight": 201,
          "frameWidth": 900,
          "locator": "http://freesound.org/data/displays/392/392617_7383104_spec_L.jpg",
          "prefLabel": "Spectrogram"
        }
      ],
      "originalFile": {
        "@type": "ac:AudioFile",
        "audioChannelNumber": 2,
        "sampleRate": 44100,
        "sampleSize": 24
      }
    },
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

```
        "index": 1
      },
      {
        "@type": "ac:AudioCollectionNode",
        "content": {
          "@id": "freesound-sounds:327666",
          "@type": "ac:AudioClip",
          "license": "http://creativecommons.org/licenses/by-nc/3.0/",
          "description": "A furious dog barking.\r\n\r\n",
          "title": "Dog Bark.wav",
          "audioCategories": [
            "freesound-tags:bay",
            "freesound-tags:aggressively",
            "freesound-tags:barking",
            "freesound-tags:bark",
            "freesound-tags:intimidating",
            "freesound-tags:bellow",
            "freesound-tags:snap",
            "freesound-tags:snarl",
            "freesound-tags:yell",
            "freesound-tags:growl",
            "freesound-tags:shout",
            "freesound-tags:fierce",
            "freesound-tags:Dog",
            "freesound-tags:thunder",
            "freesound-tags:aggressive",
            "freesound-tags:OWI",
            "freesound-tags:dogs",
            "freesound-tags:woof",
            "freesound-tags:yelp",
            "freesound-tags:roar",
            "freesound-tags:barked",
            "freesound-tags:angrily",
            "freesound-tags:bawl",
            "freesound-tags:big",
            "freesound-tags:yap",
            "freesound-tags:breaking",
            "freesound-tags:cry",
            "freesound-tags:dog"
          ],
          "author": "freesound-users:Juan_Merie_Venter",
          "availableAs": [
            {
              "@type": "ac:AudioFile",
              "audioChannelNumber": 2,
              "bitRate": 128000,
              "hasAudioEncodingFormat": "mp3:",
              "locator": "http://freesound.org/data/previews/327/327666_5632380-hq.mp3",
              "sampleRate": 44100,
        "sampleSize": 24
      },
      {
        "@type": "ac:AudioFile",
        "audioChannelNumber": 2,
        "bitRate": 192000,
        "hasAudioEncodingFormat": "ogg:",
        "locator": "http://freesound.org/data/previews/327/327666_5632380-hq.ogg",
        "sampleRate": 44100,
        "sampleSize": 24
      },
      {
        "@type": "ac:AudioFile",
        "audioChannelNumber": 2,
        "bitRate": 64000,
        "hasAudioEncodingFormat": "mp3:",
        "locator": "http://freesound.org/data/previews/327/327666_5632380-lq.mp3",
        "sampleRate": 44100,
        "sampleSize": 24
      },
      {
        "@type": "ac:AudioFile",
        "audioChannelNumber": 2,
        "bitRate": 80000,
        "hasAudioEncodingFormat": "ogg:",
        "locator": "http://freesound.org/data/previews/327/327666_5632380-lq.ogg",
        "sampleRate": 44100,
        "sampleSize": 24
      }
    ],
    "duration": 6472.47,
    "images": [
      {
        "@type": "ebu:Picture",
        "frameHeight": 71,
        "frameWidth": 120,
        "locator": "http://freesound.org/data/displays/327/327666_5632380_wave_M.png",
        "prefLabel": "Waveform"
      },
      {
        "@type": "ebu:Picture",
        "frameHeight": 201,
        "frameWidth": 900,
        "locator": "http://freesound.org/data/displays/327/327666_5632380_spec_L.jpg",
        "prefLabel": "Spectrogram"
      },
      {
        "@type": "ebu:Picture",
        "frameHeight": 201,
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

```
                    "frameWidth": 900,
                    "locator": "http://freesound.org/data/displays/327/327666_5632380_wave_L.png",
                    "prefLabel": "Waveform"
                  },
                  {
                    "@type": "ebu:Picture",
                    "frameHeight": 50,
                    "frameWidth": 50,
                    "locator": "http://freesound.org/data/displays/327/327666_5632380_spec_M.jpg",
                    "prefLabel": "Spectrogram"
                  }
                ],
                "originalFile": {
                  "@type": "ac:AudioFile",
                  "audioChannelNumber": 2,
                  "sampleRate": 44100,
                  "sampleSize": 24
                }
              },
              "index": 2
            }
          ]
        }
      ]
    }
```

Table 5.2. JSON-LD response from mediator

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382