STREAMLINE. FAST REACTIVE ANALYTICS

D4.1 - Target events type per industry and benchmark per KPI

688191	**Project Number**
STREAMLINE	**Project Acronym**
R: Report	**Nature**
Public	**Dissemination Level**
WP 4	**Work Package**
30th November 2016	**Due Delivery Date**
30th November 2016	**Actual Delivery Date**
November 24th, 2017	**Updated Date**
IMR	**Lead Beneficiary**
Jorge Teixeira (ALB) Vânia Gonçalves (NMusic) Filipe Correira (NMusic) Philippe Rigaux (IMR) Henri Heiskanen (Rovio) Björn Hovstadius (SICS – project coordinator)	**Authors**

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 688191.

Executive Summary

STREAMLINE aims to provide a real-time analytics platform that is capable to improve Apache Flink framework on online stream learning, data mining and fusing data *at*-rest and data *in*-motion, and apply it to four major sectors: telco, media content, games and web content. With a focus on predictive contextualisation and cross-sector data fusion, the platform should be suitable for non-technical users, providing an easy to use query language.

The present document investigates the potential of contextualization for machine-learning algorithms in general, and recommendation methods in particular. Contextualization denotes the ability to take into account *events* that affect the behavior of agents, e.g., end users. WP4 ambitions to introduce contextualization data as part of the input of machine-learning algorithms, and to support the evaluation of continuous event impact on the business activities of the four use case partners of STREAMLINE.

The document details the design of the contextualization engine that will be developed until the end of the project, and study requirements and opportunities that are specific to each use case.

Table of Contents

1 Introduction

The work conducted in WP4 aims at investigating the impact of *context* on the activity of users, and more specifically how the influence of contextual information can be taken into account to explain and predict changes affecting the user behavior. Indeed, if we were able to build a typology of the external conditions that impact some business activity (e.g., music tracks or videos chosen by online media distribution platforms), along with a modeling of how some change in these conditions affect user as a function of their profiles, we could build more informed analytical and predictive models.

Work Package 4 conducts a study of these general ideas by focusing on the following aspects.

1. Context changes are represented by *events* captured from various sources, including extraction from Web pages, news sites, feeds, and specialized channels. An event, as we define it, is any piece of information that features a (geo)location and a temporal qualification (timestamp or time range).
2. The impact of events is evaluated with respect to some *business activity.* The exact nature of this activity depends on the specific domain. In the context of StreamLine we can investigate the needs of several industries represented by the four StreamLine use case partners: NMusic, Altice Labs (ALB), Rovio, and IMR.

During the initial phase of the project we aimed at achieving the following objectives, as required by the Description of Work:

- **Scale source identification and processing using stream classification and machine learning to enable critical mass of events time series to be created.**

  At the beginning of the project: the scraping process was semi-supervised. We now operate a fully automatic extraction, relies on ML methods, and application domain heuristics. Scalable event extraction is now possible. The scraping technology has been demonstrated with the Bomerce app. *Our investigations on events collection scalability progress are reported in Section 2.*

- **Have business experts (marketing, BI) identify relevant types of events potentially impactful for their business KPI**.

  We evaluated and tested various event source types evaluated and tested. In particular, we closely worked with our partner NMusic to identify and scrap 40 sites of interest, collected business stats. The main conclusion is that business activities is affected by a very specific type of events that can mostly be found on specialized web sites. *We report our study on that matter in Section 3*.

- **Calculate a method to measure Context Impact per event type for each business case to determine where they are useful to augment predictive power of current models**.

  We devised a methodology to match events and business time series and carried out a preliminary evaluation of off-the-shelf algorithms (TS decomposition, and rule mining). Although seductive in theory, the methodology appears at this point quite speculative, as it requires a very dense collection of relevant events, and qualified business descriptors (i.e., including geolocation). *The design of a Contextualization Engine is introduced in Section 4.*

Based on this first phase, and following the departure of NMusic, we proposed a redirection of WP4 towards objectives that emerged amongst the remaining industrial partners. *Our motivations are reported in the concluding section, along with plans for a reorientation of WP4 during the second phase of the project.*

In terms of analytics, contextualization mostly relates to evaluating how external events might affect *recommendation* algorithms. For NMusic for instance, which provides online audio content, relevant context might include the publication of a new audio record, a concert given by a specific artist in a specific city, a TV show featuring a list of artists, and this context is likely to boost their audience on the NMusic channel, etc. The same kind of contextual information holds for the Quadruple-play industry represented by Altice Labs. Gaming (Rovio) might be influenced by events that keep players from focusing on their game activity: weather conditions, highly popular sport event, etc.

This contextualization study is also designed as an enabler of the STREAMLINE technology. Events are typically supplied as data streams, and need to be cleaned, annotated, sorted, and grouped in real time to constitute a sound basis for analytic algorithms. Taking into account the impact of events on user behavior requires a mixed processing of these events flows together with the (mostly) static mass of user profiles. Serious challenges potentially result from the accurate capture and qualification of events, at scale, combined with a fusion with internal data based on a matching of temporal and location information at the appropriate level of precision. The contextualization endeavor represents therefore an interesting target to assess the new capabilities of Flink, developed during the course of STREAMLINE.

The goal of WP4 is to build a database of events (collected by IMR from several Web sources) and user activities (supplied by the use case partners of STREAMLINE), in order to use this database as a basis for developing and evaluating a contextualization engine built on top of Flink. In the present deliverable, we detail the initial design of this engine, and examine the use case specificities regarding the relevant types of events that might potentially be useful to improve the accuracy and relevancy of the ML algorithms, and particularly of the recommendation algorithms. To this end we proceed as follows:

1. We establish a typology of the event sources that can be publicly accessed and studied, experimented with acquisition methods, and developed generic, specialized wrappers.

2. We present the results of the discussions with informed experts of the use case partners in order to identify representative sources of events, likely to produce impactful events for the use cases activity in general, and user behaviors in particular. A database of events has been initialized with events collected from these sources.
3. We aim at formalizing and measuring the improvements expected from the consideration of events and context information in the recommendation algorithms, in the specialized settings of each use case.

The rest of the deliverable is structured as follows. Section 2 is devoted to *events*. We propose a typology of the event sources that we investigated during the first phase of the project in order to determine their quality and ability to supply flows of informed events. In Section 3, we focus on *business activities*. We examine the industry use cases of each StreamLine partner, identifying specific needs, KPIs, and relevant event sources. Finally, Section 4 describes the initial design of the *Contextualization Engine*, in order to found the basis of the services expected from the system. We propose a high-level description of the future contextualization functionalities, and the main workflows that need to be supported by the underlying Flink infrastructure.

2 Events

Our aim is to collect events, at large scale, from public Web sources. We first give a brief overview of the data acquisition workflow in the system that we envisage and then detail the range of event types.

2.1 Event acquisition workflow

The following figure shows a high-level view of the event acquisition workflow.

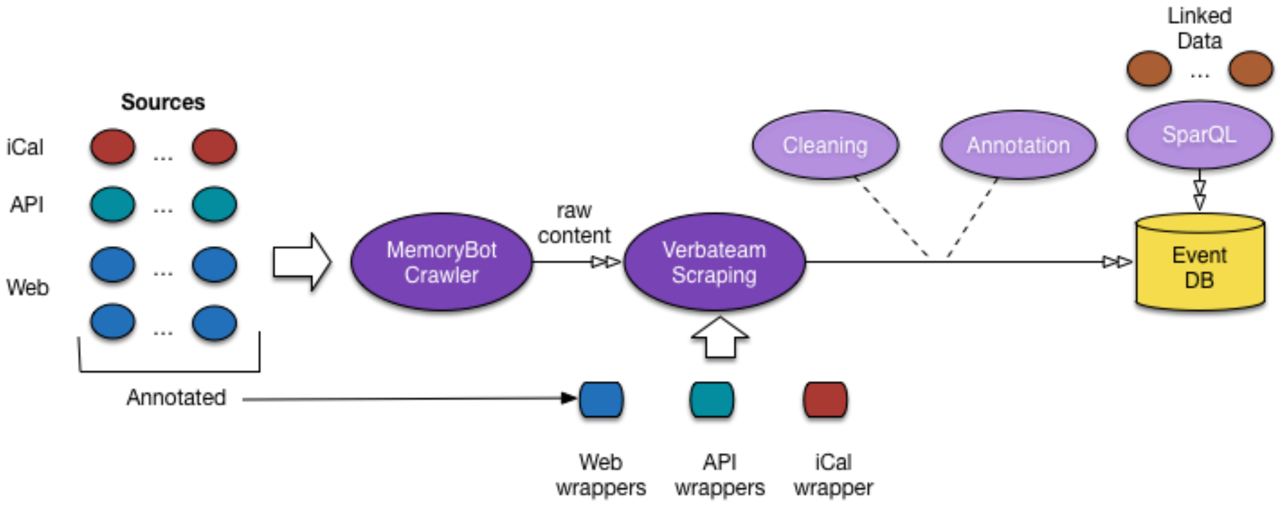

Fig. 1: the event acquisition workflow.

2.1.1 Crawling and scraping

The two main components are the MemoryBot crawler, and the Verbateam Scraper. The combination of these components allows to extract structured data from unstructured web sources.

MemoryBot is a Web crawler developed by Internet Memory Research. It is designed to collect data at Web scale in a distributed setting, and can operate in two modes:

1. **Discovery**: samples are extracted from web sites, and a classifier determines the type of the site (e.g., portal, eCommerce, news, institutional). The site is then recorded in the Wep Map, a catalog maintained by IMR which references millions of sites annotated with semantic features: type, activity, structure.
2. **Analysis**: when a decision to extract structured data from a given site is made, MemoryBot scans the pages, learns the graph structure of the site, and identifies the type of pages. Two recurrent types are *catalog*, which typically shows a list of items, along with a link to the individual item, and *detail*, which typically shows the complete public information related to a specific item. A supervised analysis allows to produce a wrapper for these pages (see below).

3. **Extraction**: in extraction mode, MemoryBot carries out a systematic exploration of the catalog or detail pages, download each page, and relies on the Scraping module to apply the relevant wrapper and obtain a structured record made of features selected from the page. The workflow shown in the above figure includes our crawler/scraper running in extraction mode.

A sample of *catalog* and *detail* pages are selected, and sent to the Scraper management tool, called Verbateam. Scrapping is a supervised process. The page samples are annotated by Web experts, who find and select the part of the page that contains some relevant information. For *catalog* pages, for instance, this information consists in the item reference and the item links, both referred to by an XPath expression relative to the DOM structure of the page. For *detail* pages, it typically consists of the list of features of the exposed item: title, price, author, etc.

Finally, based on the sample of annotated pages, Verbateam is able to produce a *wrapper*, i.e., a tool apt at extracting catalog entries or items features from any page sharing the corresponding structure. The wrapper is used at extraction time.

2.1.2 Extending the scraping process for events

MemoryBot is used intensively by IMR for searching and analysing retail sites (refer to the IMR use case in Deliverables 5.1 and 5.2). For the StreamLine contextualization project, we adapted its functionalities to satisfy the goal of selecting and extracting event data. This involves two extensions:

1. The *Catalog* and *Detail* page types have been derived to obtain an *EventList* and *Event* type. This extension is minor. An EventList page contains a date-sorted list of events. The classifier takes into account the presence of characteristic features such as a date, a location, an artist name, etc.
2. More significantly, MemoryBot has been extended to capture Web data beyond standard HTML pages. We indeed identified a list of potential event source types that seemed to be interesting candidates for event collection. They are represented on the above figure, at the same level than HTML web input: iCal, APIs, RSS feeds, and finally the Wikipedia log entries, that often contain material that can be interpreted as events. We report our study on those source types below.

For each new source types, we implemented a “wrapper” taking as input a source document, and delivering a structured record describing the event. These “wrappers” turned out to be much simpler than HTML wrappers that require complex XPath expressions inferred from a few manual annotations.

1. For iCal documents, we wrote a simple parser that takes the fields from the iCal format; the parser is generic and can be used whatever the source.
2. For RSS feeds, standard XML parsing is used. Since the RSS format is fixed, the parser is generic as well.
3. APIs are a bit more complex since the supplied documents vary from one provider to the other. Still, since documents are JSON-encoded, implementing a parser for a specific source is trivial.

4. Finally, Wikipedia, as an event source, is a special case. Data can be obtained in RDF, and therefore queried with SparQL to retrieve events of interest. Although, at this point, we did not deeply investigate this option, it seems a quite promising candidate for capturing high-quality, up-to-date events descriptions.

2.1.3 Events annotation and the Event DB

Although the scraping part is an essential part of the Event workflow, it is by no way sufficient to obtain information that can support the functionalities of the Contextualization Engine. In order to favor the accuracy of the process that matches events with business activities, we aim at extracting from the raw description of an event the following distinctive values:

1. **The GeoLocation**. Recall that we classify as event any piece of data that can be related to a location and to a time reference (point or range). However, the location is usually found in the form of raw, unstructured, and unnormalized text, and need to be cleaned, and linked to a GeoSpatial reference in order to make sense.
2. **Temporal information**. The same statement holds. Time reference is to be found with many encoding variants and needs to be normalized.

The following figure shows the Event acquisition workflow, starting from external, public Web sources, and feeding a continuously expanding Event Database, denoted Event DB in the following.

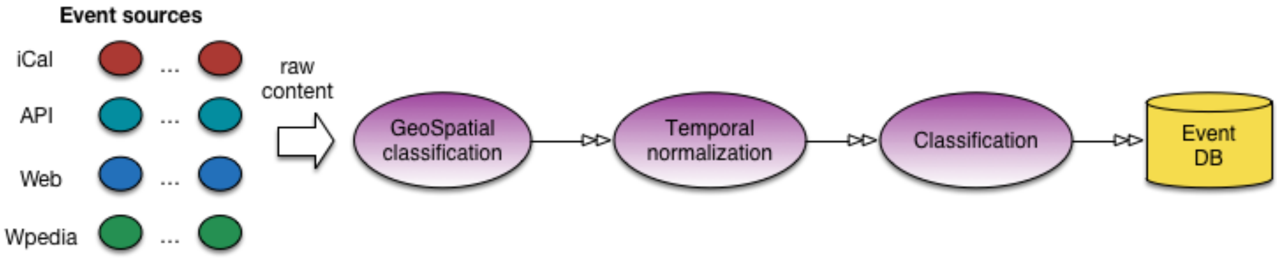

Fig. 2: details of the event workflow

2.2 Typology of event sources

We now report the result of our study on the various types of event sources investigated in the preliminary phase of the project.

2.2.1 Web scraping

Web sites are the main source of events that we chose to focus on. As explained above, our scraping mechanism was an almost ready-to-use candidate for structure event description extraction, and it quickly turned out from discussions with our partners (mostly NMusic) that web sites of high interest were identified, with unique and important data that had to be captured.

The downside of Web scraping is that it requires a time-consuming process. For each site proposed by a partner, a few (between 5 and 10) representative pages have to be annotated by our experts, the produced wrapper has to be tested, validated and possibly adjusted by our content manager, before finally being put in production. This is the price to pay to obtain accurate data. The following show an example of a structured record extracted from http://www.culturgest.pt/, one of the NMusic targets.

```
{
      "itemFeatures":[
      {"_source":{"StartDate":"DANÇA QUA 1, QUI 2 DE JUNHO",
                  "EventTitle":"Sur les traces de Dinozord",
                  "ArtistName":"de Faustin Linyekula",
                  "Image":"Destaque"},
            "StartDate":"DANÇA QUA 1, QUI 2 DE JUNHO",
            "EventTitle":"Sur les traces de Dinozord",
            "ArtistName":"de Faustin Linyekula","Image":"Destaque"},
            {"_source":{"StartDate":"TEATRO TER 7, QUA 8 DE JUNHO",
            "EventTitle":"La nuit des taupes",
            "Image":"Destaque"},
            "StartDate":"TEATRO TER 7, QUA 8 DE JUNHO",
            "EventTitle":"La nuit des taupes",
            "Image":"Destaque"
      }
      ],
      "pageFeatures":{},
      "crawlTimestamp":"2016-06-01T15:03:00+0200",
      "url":"http://www.culturgest.pt/",
      "sourceId": 1
}
```

2.2.2 Improved Scalability

At the beginning of the project, the scraping methodology relied on an annotation mechanism. In order make the method scalable, we achieved a fully automatic crawl and extraction technology, which relies on a combination of ML algorithms and domain specific heuristics.

Recall that we aim at finding two categories of pages: *catalog pages* refers to a list a items, and *item pages* fully describe a specific item. In the context of event sites (Work Package 4), the catalog page refers to a list of events, the details of which are exposed in item pages; in the context of eCommerce sites (Work Package 5), we deal with list of products, and product offers. The semi-supervised approach has led us to constitute a large DB (approx. 2,5 M) of wrappers for either catalog and item pages. We used this DB to learn a classification model and detect, at crawl time, whether a crawl page belongs to one of these categories.

In order to produce a wrapper for an item page, we developed a set of heuristics that currently focus on the eCommerce domain. They are based on micro data exploitation, on-line analysis of the page structure, and matching of the data common to the catalog and item page. In short, one can find in the catalog page a summary of each referred item (for instance the name and the price of a product) and this allows to detect the location of the common features in the item page.

This technology, now fully operational for the eCommerce use case can be transposed to events detection and extraction. This requires the production of a training base, in order to automatically detect event sites ate crawl discovery time, and investigations on event-specific heuristics.

2.2.3 iCal

iCalendar is a textual format designed to represent temporal data, including events. iCalendar data data can be found everywhere on the web as .ics files. The format is trivial to parse, and seemed initially a quite promising candidate for collecting events at large scale and low cost. The following is an example of iCal file featuring a VEVENT entry.

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//hacksw/handcal//NONSGML v1.0//EN
BEGIN:VEVENT
UID:uid1@example.com
DTSTAMP:19970714T170000Z
ORGANIZER;CN=John Doe:MAILTO:john.doe@example.com

DTSTART:19970714T170000Z
DTEND:19970715T035959Z
SUMMARY:Bastille Day Party

END:VEVENT
END:VCALENDAR
```

As it appears clearly on this example, the design of iCal is strongly oriented toward the description of business meetings, and more generally activities that can take place in a personal calendar. This favors the representation of live events: concerts, sports events, open public meetings, that anyone can register (if required) and attend to. On the other end, TV shows, or events related to a new occurrence or publication in the cultural sphere (some new movie, book) are less likely to be found in iCal. And, finally, important events that do not imply any participation (and thus do not need to be recorded in a personal calendar) turn out to be ignored by iCal channels. This covers for instance events affecting famous people, e.g., the death of David Bowie, which nevertheless is of potential interest due to its potential impact on user activities.

We searched for regular sources of iCal records. After some investigations, we managed to collect events from iCal sources, mostly related to the following types of events.

1. **Sport**. Calendars for popular sport competitions are published on specialized channels. The matches of the Real Madrid football team for instance can be obtained by registering to events webcal://ical.mac.com/kusandore/Real%20Madrid.ics. The same holds for many other kinds of sports, with a coverage of most of the famous teams or players. It becomes quickly obvious, though, that the publication process is highly anarchical, with a lot of redundancy between the various calendars (e.g., the Arsenal football team calendar and the First League calendar) and a poor representation of the less popular actors.
2. **Holidays**. Holidays, with sport, are the most represented event types in iCal channels. See for instance webcal://icalx.com/public/icalshare/US%20Holidays.ics. This makes sense, since holidays are of interest to many people and institutions. Holiday calendars are also those that enjoy the highest quality (coverage and accuracy).
3. **TV programs**. Finally, the third most important type of event source as iCal records are TV programs. The French TNT programs for instance can be obtained at webcal://dynical.com/iCal/television.ics/?lng=fr&zone=TNT&zone_=E-Y-S-6. The coverage, however, appears to be poor, as well as the quality content.

Overall, the attempts to collect events from iCal sources turned out to be disappointed. There are many reasons.

1. Publication of iCal source often comes from individuals, leading to highly unreliable channels: the publication periodicity appears to be erratic, the coverage is quite narrow, and the content is often misleading or incorrect.
2. The format is poor and limited, and iCal records often contains basic information which is insufficient to derive some useful knowledge. TV shows calendars for instance, most often, do not mention the precise content of TV programs, which prevents from identifying known personalities. In general, we found that the format is often misused, and therefore unable to serve the needs of an automatic acquisition at large scale without a constant a costly manual intervention.
3. Last, but not least, the content of iCal documents can often be found in more reliable sources, particularly APIs and feeds.

We thus decided not to develop our investigations on this format, and in particular we stopped evaluating the quality of iCalendar sites. The technology currently implemented captures events for a list a known and reliables iCal sources, mostly for holidays and sport events. We do not plan to expand this list.

2.2.4 APIs

Public APIs are the second event source type that we examined. As a matter of fact, Web APIs deliver the most accurate data representation, compared to Web scraping or iCal sources. Most services allow to retrieve structured content, encoded either in JSON or XML.

Some online services are specialized in delivering events. We chose to try one of those services to avoid the heavy process of discovering local sources, which involves an implementation of a specialized extractor. Eventful (http://eventful.com) is a platform that collects events, and enables licensed partners access Eventful's data, features and functionality via the Eventful API.

The query API allows to select events based on their main features: location, time, and keywords matching the textual description of events.

We registered to the service and implemented a periodic pull mechanism based on a set of subscription queries. Each such query targets localized events, and is intended to cover the needs of a particular use case. In the context of our cooperation with NMusic for instance, we subscribed to concerts, festivals, and a few highly important artists (as defined by the use case) in Portugal. We then retrieved the Eventful data during three months through daily calls to the Eventful API.

The results are partly satisfying. Collected events are well described, and the level of precision is the highest with respect to other event sources. On the other hand, events collected via EventFul are sparse, and can only constitute a marginal complement of the events obtained by scraping web sites of high interest to the use case.

2.2.5 Wikipedia

We mention to conclude an event source that we only partially investigated, namely Wikipedia updates. It turns out that, for many highly popular events, relevant data is published in Wikipedia, and updated almost in real time. The UEFA Euro for instance, held in France in summer 2016, is referred to in Wikipedia by the following page: https://en.wikipedia.org/wiki/UEFA_Euro_2016. All the matches were accurately described as they were planned, and their result reported without delay.

Such events can be captured thanks to a Logstash (a tool in the ElasticSearch suite) interface at https://logstash-beta.wmflabs.org. This source appears to be quite promising for high-impact events, in particular those that can hardly be related to a precise location (e.g., events affecting some famous artist), and are therefore of interest to most use cases. Although we did not develop a dedicated acquisition mechanism for Wikipedia data, the source type appears to be promising for future extensions of the Event collection platform.

2.3 Status of the Event DB

In summary, we studied and implemented the core functionalities of a generic Event acquisition platform. The platform combines extractors from several types of event sources, and feeds a Postgres Event database. At the time of writing, this DB contains 5+ million events.

1. Hollidays: collected from iCal and APIs referring to holiday periods for more than 70 countries
2. Weather: full coverage for US and Europe (from public APIs)
3. TV programs (APIs): for a few European countries
4. Sports (ICal, APIs): basket, football,
5. Music events, festivals, concerts (specialized API, namely Eventful)
6. Music events, obtained by website scraping (list defined by NMusic)

The DB is constantly expanding by pulling new data from these registered source, and we expect to regularly add new sources, proposed in particular by the other use cases.

The main conclusion of the study conducted during the first year of the project is that web scraping is the most satisfying source for collecting events of interest. It allows to focus on specialized web site, highly relevant to a specific use case, and supply a good coverage of local events as soon as a set of representative sites have been identified. This requires a close coordination between the events collector (e.g., IMR in our case) and the use case manager (so far, we mostly worked with NMusic) in order to identify reliable sites, and define the fields that must be extracted.

The cost of web scraping is an obstacle to scale the method to a very large number of sites, though. Other source types that we examined seems to be only able to supply a complement to this main source of information. Automatization of the web scraping, with a minimal human supervision, is the key to extend the capture of events at very large scale.

3 Business activities

The Contextualization Engine aims at matching events with *business activity* statistics (*"stats"* in short) to measure the impact of the former on the latter. These statistics will we stored in a designated database, called the Stats DB in the following.

In the current scenario, both events and stats will be collected by IMR and stored in a repository in our premises. The Event DB and the Stats DB will serve as input to estimate the impact of events on statistical data, and adapt recommendation algorithms accordingly. Variants can be envisaged, such as for instance a context impact analysis based on a sample of business data, in order to identify the event types of interest. The IMR partner could then register to the Event DB services to get relevant event data and carry out the context-aware analytic process in its own premises. This avoids to fully depend on IMR, and preserves data privacy since only a sample is required.

The following Figure shows the Stats acquisition workflow. It is essentially similar to that of the Event acquisition workflow. Structured data supplied by some institution wishing to enrich its analytic algorithms with contextual information supplies a flow of statistical information related to its business activity. This information essentially takes the form of structured records featuring temporal data (at least) and location data (if possible). Those two items are cleaned, normalized and linked to external sources in order to prepare the next steps, namely a join-like operation that associates events and statistics based on their common spatio-temporal properties.

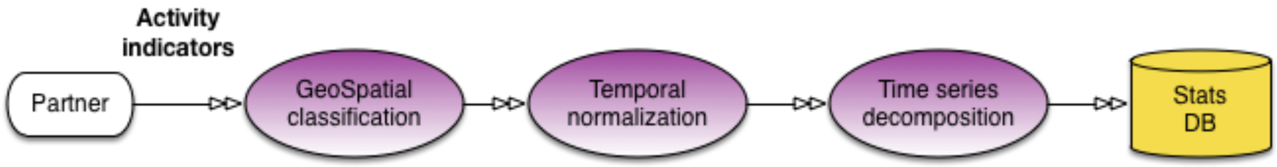

Fig. 3: workflow for statistical data

This general scenario admits several variants, based on the specific use case, and on data protection issues that may arise.

In this section, we describe the outcome of discussions involving the use partners, devoted to the introduction of contextualization data in their respective data mining operations.

3.1 NMusic

The NMusic use case focuses on recommendation, with two variants

1. Events can be used to suggest relevant editorial content to content curators. An event affecting an artist (concert, new CDs) is for instance incitative to the creation of playlists and highlights by NMusic's content curators. These would then be available to the users through the music-streaming apps.
2. Events can also intervene in proposing recommendations and playlists to the NMusic end user, based on their profile, location and past choices.

In addition, NMusic is interested by building a searchable database of events that would let users browse forthcoming events related to their favorite artists, and to filter them according to their location and interests.

3.1.1 Event types

The NMusic use case requires an acquisition of the following event types:

1. Concerts.
2. Festival.
3. Interviews / TV shows.
4. News related to an artist / album (publication, award, etc.).

A common features is that the location of these events should be in Portugal. As explained in the previous section, those events can be collected from several source types. The APIs that we found (mostly Eventful), although promising, have been unable to supply more than a few events. We, therefore, resort to web scraping as the main source to capture events. The following sources have been crawled and scraped on behalf of NMusic by IMR.

```
http://www.casadamusica.com/pt/agenda?lang=pt#tab=calendario
http://www.culturgest.pt/
http://www.serralves.pt/pt/actividades/
http://www.egeac.pt/
https://www.visitportugal.com/pt-pt/encontre/grandes-eventos/list
https://www.guiadacidade.pt/pt
http://www.tnsj.pt/home/programacao/
https://www.lazer.publico.pt
http://ticketline.sapo.pt/
http://arena.meo.pt/agenda
http://www.coliseulisboa.com/agenda.aspx
https://www.ccb.pt/Default/pt/CCBProgramacao
http://www.gulbenkian.pt/inst/pt/Agenda
```

At the time of writing, the content of the Event DB is summarized by the following tables.

Catalog-level pages:

3 315	Number of URLs

186 399	Number of crawls
250 MBs	Size

Item-level pages:

88 464	Number of URLs
712 212	Number of crawls
6.34 GBs	Size

3.1.2 Prototype

NMusic use case matches the main scenario of the contextualization engine given above. Events are collected by IMR and business activities are pulled from the use case partner in the IMR Stats DB, from where the matching process between stats and events can take place. This gives the potential to fully carry out at the same place the both analytic process that determines influent event types, and the adapted recommendation algorithm that takes into account use events occurrences.

A Kafka server has been set up to let STREAMLINE partners, and in particular IMR, retrieve the business activities. Stats are published to the following Kafka topic:

```
user.activity.tracktransactions.grouped.anonymized
```

Each item describes when some user listened to a specific track of a specific CD. The data features the user location and some useful descriptive information such as the artist name. All these features are packages in JSON documents containing the following fields (table taken from Del. 5.1, for completeness).

Description	Type	Field
The date for the reported number of plays/skips.	string	`Date`
The user id for the reported number of plays/skips.	string	`user_id`
The artist for the reported number of plays/skips.	object	`main_artist`
The id of the artist.	string	`Id`
The name of the artist.	string	`Name`

The location for the reported number of plays/skips	string	`Location`
Number of plays.	integer	`plays_count`
Number of skips (NMusic is currently working to make the `skips_count` more reliable, as older versions of the app are reporting erroneous values).	integer	`skips_count`

At the time of writing, NMusic user activity has been collected since May 2016. The content of the Stats DB is summarized below.

51 410	Number of users
19 025	Number of artists
353 942	Number of tracks

3.1.3 KPI

The use of contextualization data is expected to have an impact on the KPIs of NMusic's use cases, to different extents. The following four KPIs are expected to be positively impacted to the extent that contextualization data will improve the quality of the recommendations of content to the end-users:

- KPI 1 – Number of users that consume recommended content per day.
- KPI 2 – Number of recommendations consumed more than 50% of their length.
- KPI 3 – Share of session time spent consuming recommended content.
- KPI 4 – Timeliness of recommendations.

The following two KPIs are expected to be impacted to the extent that the quality of the editorial content is improved, by the recommendations provided for content curation:

- KPI 5 – Time spent curating content.
- KPI 6 – Quantity of curated content.

In the current state of the contextualization design, we found in premature to target a precise level of improvement for those KPIs. When the contextualization system is in place, we will conduct a comparative study. We will define a subset of the NMusic users, statistically representative of the full population, and will experiment the impact of the context-aware recommendation algorithm, to be compared with the results of standard, context-agnostic algorithm still proposed to the rest of the population.

3.2 Altice Labs

Altice Labs use case is close to that of NMusic by its motivation, but differs with respect to the data communication workflow. The goal is to recommend content (in that case, IP TV programs) to users based on their past activity and to benefit from contextual data to improve the recommendation results. However, Altice Labs, due to privacy users concerns (please refer to D7.2 - Ethics Management Plan) is not able to provide users data to an external..to an external institution, and the context-aware recommendation algorithm, therefore, cannot be run in the Event provider (here, IMR) cluster.

We must, therefore, adapt the processing logic of the contextualization engine by splitting it in two parts.

1. Altice Labs will send a sample of anonymized data to the contextualization engine. From this sample, an analytic process will infer the event types that impact the user activity, and will determine a characterization of these types.
2. Altice Labs will subscribe to the event types characterized by the above analysis, in order to retrieve flows of events collected by IMR and matching their ongoing user activity. Altice Labs will then use internally these event as input of the context-aware recommendation algorithm.

The following figure summarizes the workflow of exchanges and the two main algorithms involved in the process.

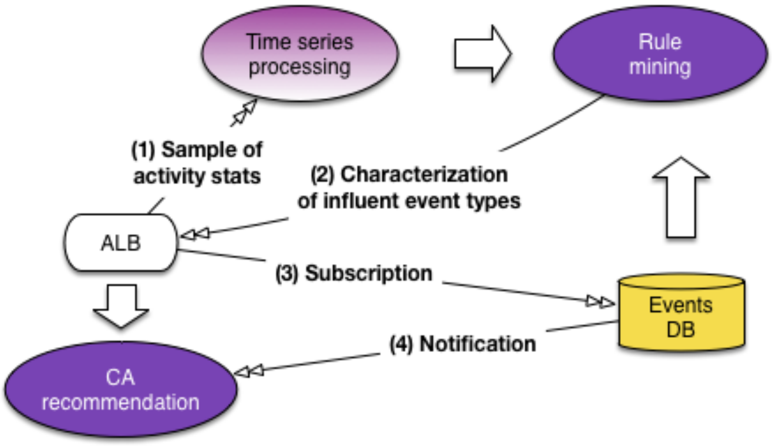

Fig. 5: the ALB workflow

Altice Labs sends first a representative sample of user activities (1). Based on past collected events, stored in the EventDB, IMR will produce *rules* that describe how specific events influence user activity.

This involves a Rule Mining algorithm that essentially identifies outliers in user activities, modeled as times series, and matches these outliers with outstanding events based on common location and temporal properties. We made a first study of this approach. It combines two well-known techniques: time series decomposition, and frequent item sets mining. We plan to work on a Flink implementation of these algorithms together with Sztaki as part of Work Package 2 and 3.

The contextualization engine will send back to Altice Labs a characterization of influent event types, determines by the rule mining process (Step 2 in the figure). Altice Labs can then subscribe to an event delivery service (Step 3) and gets notified of newly collected events (Step 4). The second main algorithm is the context-aware recommendation algorithm (*CA Recommendatio*n on the figure). In this case, unlike the NMusic scenario, the recommendation algorithm will be executed by Altice Labs, thereby avoiding dissemination of user-related data.

These requirements, motivated by the need to protect sensitive information about Altice Labs users and their activity, dictate an adaptation of the architecture of the contextualization engine and of the services it provides. We will summarize at the end of this section the foreseen architecture and its main components.

3.2.1 Event types

The following event types have been identified as relevant to Altice Labs recommendations.

1. Release of a new movie
2. Programs of movie theaters, see 5.1 for some examples. Includes location.
3. News about an actor, a director (from ImDB for instance)
4. TV programs and series
5. Music festivals (to recommend music programs, based on the genre)
6. Sport events

Unlike NMusic, ALB users are not precisely geolocalized. ALB will therefore provide an initial set of web sites to capture events from, covering the Portugal area. The rest of the event processing will be similar to that of NMusic, the main difference being that the matching process will not take the geolocation into account.

3.2.2 KPI

The following KPIs are expected to be impacted by the use of contextualization data.

- KPI: accuracy of the prediction model for the behavior of a given customer, taking context into account. Here, “behavior” must be understood as “usage pattern”. We will use the correlation between a profile and specific event types, as determined by the Rule Mining process, to determine /predict how the audience can be impacted.
- KPI: rate of recommendations, which is measured using the number of recommendations each customer receives under a particular scenario. The goal of this KPI is to evaluate the capability of the system to provide recommendations to customers, and it does not take into account, at this stage, the quality of the recommendations
- KPI: Customers rejection measures the rate of rejected recommendations provided to customers. This is measured by the number of times each customer premeditatedly removes a particular recommended content or category.
- KPI: Customers engagement is measured based on the number of recommendations that each customer followed. Whenever a customer receives a recommendation, either

because he specifically looked for by navigating through the set top box menu or because it showed in the screen, it is assumed that the recommendation has a positive impact in the customer – and thus improves engagement – if the customer selects or watch that particular recommended content.

- KPI: Recommendation success rate is a combination of Customers Engagement, Rate of Recommendations and Customers Rejection KPIs that aims to assign a success rate to the recommendations provided to the customers' under context constrains such as a particular time frame or set of customers'.

3.3 Rovio

Rovio expressed some interest to the perspective of a contextualization service, but since the recommendation use case is no longer considered as part of Streamline, plans for using context information as a support for other services is not yet finalized nor definitely confirmed.

At the moment, the biggest interest would be a global calendar view of events that affect the spending behavior so we can plan live operations accordingly. The most relevant event types would be:

- holidays in global level
- tax refunds
- black fridays of the world
- etc.

The underlying scenario does not fully match the contextualization project principles, since there is no business activity, in the form of times series of statistical indicators, involved. Rovio gaming business is rather impacted by trending topics in pop culture. However, trend detection is beyond the scope of the event-centered contextualization approach.

In summary, the opportunity of applying context-aware algorithms for Rovio remains to be confirmed. While we plan to continue discussion and discuss perspectives on this matter, the focus will, therefore, remain on using context data as side input of the recommendation algorithms, which appears to be central motivation of the two other uses cases, namely NMusic and Altice Labs.

3.4 Summary: design of the Contextualization Engine

The figure below summarizes what, at this point, is expected from the forthcoming contextualization engine (CE) in terms of functionalities and services. The focus is put on using Flink as a data processing support, and on machine-learning algorithm that process streams of events and stats.

Two main acquisition workflows need to be set up. The time series processing workflow receives timestamped statistical indicators. The CE must process these flows of indicators to:

1. Clean and normalize data, regarding in particular geolocation (link to some GeoNaming service), temporal representation normalization, and analysis of descriptive data (disambiguation of artist names for instance).
2. Apply time series decomposition to identify trends, seasonality effects and outliers.

The second set of operators (times series processing) should be integrated in Flink, of a library associated with Flink. This is part of cooperation with WP 2&3.

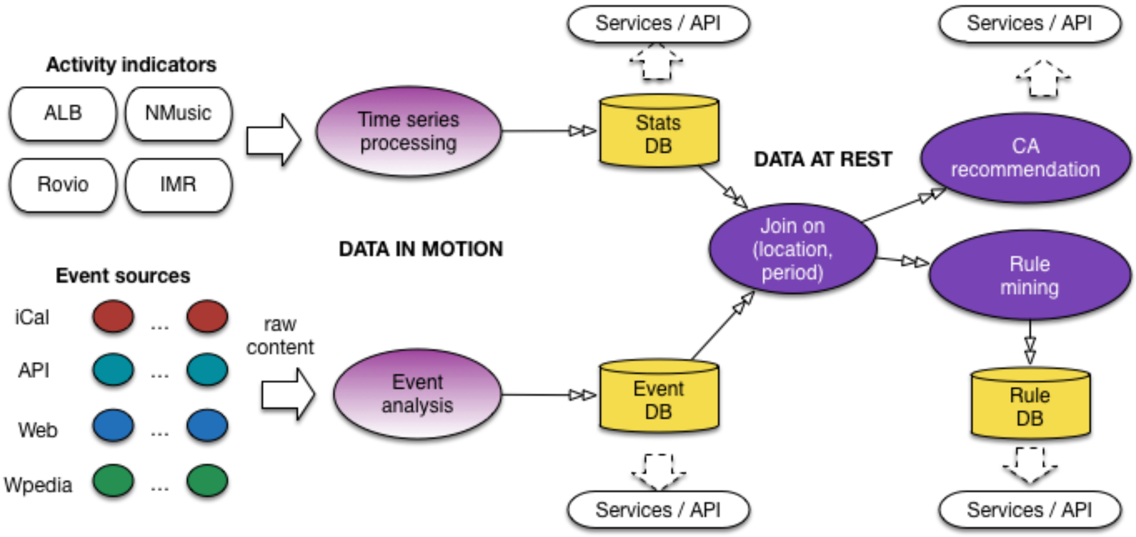

Fig. 6: global view of operators and services

The second workflow receives flows of events and carries out an event normalization and annotation process, as described in what precedes.

Both workflows feed two data warehouses of, respectively, statistical data and events. Services to access these DB have been required by our partners.

1. The Stats DB should be accessed to deliver the decomposition of time series, showing trends, seasonalities and outliers.
2. The Event DB should be accessed to support information retrieval operation. This is required, for instance, by NMusic to enrich the user interface with a search mechanism on events matching user interests.

Both the Stats DB and the Events DB serve as input of a join/matching process that associates events and statistical indicators based on spatio-temporal information. This process will deliver combined flows of (events, indicator) data that can be used for two analytic algorithms:

1. *Rule mining*: its goal is to determine event types that seem to influence a business activity.
2. *Context aware recommendation*: its goal is to consider influent events as part of the recommendation process.

The context-aware recommendation process is required to be part of the Flink analytic suite, since it can be used internally by the business activity provider, and not by the event provider (refer to the Altic Labs use case above).

4 Design of the contextualization engine

This section introduces the design of a contextualization engine apt at matching the Event database with the statistical database. The main components are summarized in Fig. 7. We performed some preliminary testing on the approach, in a centralized setting, in order top validate its feasibility.

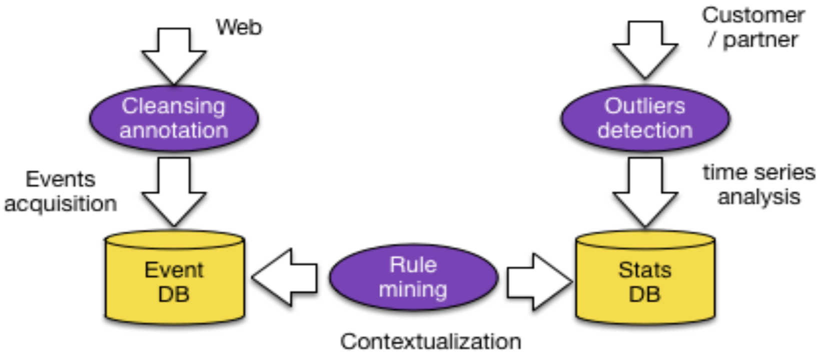

Fig. 7: Event mining component

During this phase, we worked with the customer purchasing behaviors from a retailer. The dataset is provided in the contest https://www.kaggle.com/c/acquire-valued-shoppers-challenge. It is the historical anonymized transactional data from over 300,000 shoppers from March 2012 to August 2013.

In this dataset, customer identities and product identities are masked. Data is at the point of sale level. Each transaction contains date, purchased products ID, customer ID.

The questions we want to ask are:

- Which events have impact on sales of a product?
- If the impact exists, how big is it?

4.1 Time series decomposition

We first compute the daily sale amount of each product. We then decompose the daily sale volume in components:

- **Seasonality**: A seasonal pattern exists when a series is influenced by seasonal factors (e.g., the quarter of the year, the month, or day of the week). Seasonality is always of a fixed and known period. For instance, people shop more on the weekends than on weekdays.
- **Trend**: A trend exists when there is a long-term increase or decrease in the data. It does not have to be linear. Sometimes we will refer to a trend "changing direction" when it might go from an increasing trend to a decreasing trend. For example toys are sold more before Christmas than after Christmas
- **Cyclic**: A cyclic pattern exists when data exhibit rises and falls that are not of fixed period. The duration of these fluctuations is usually of at least 2 years.
- **Remainder**: the rest which cannot be explained by seasonality and trend

Many people confuse cyclic behavior with seasonal behavior, but they are really quite different. If the fluctuations are not of fixed period then they are cyclic; if the period is unchanging and associated with some aspect of the calendar, then the pattern is seasonal. In general, the average length of cycles is longer than the length of a seasonal pattern, and the magnitude of cycles tends to be more variable than the magnitude of seasonal patterns.

The product with ID number 2122 was decomposed in three components as sown on Fig. 8:

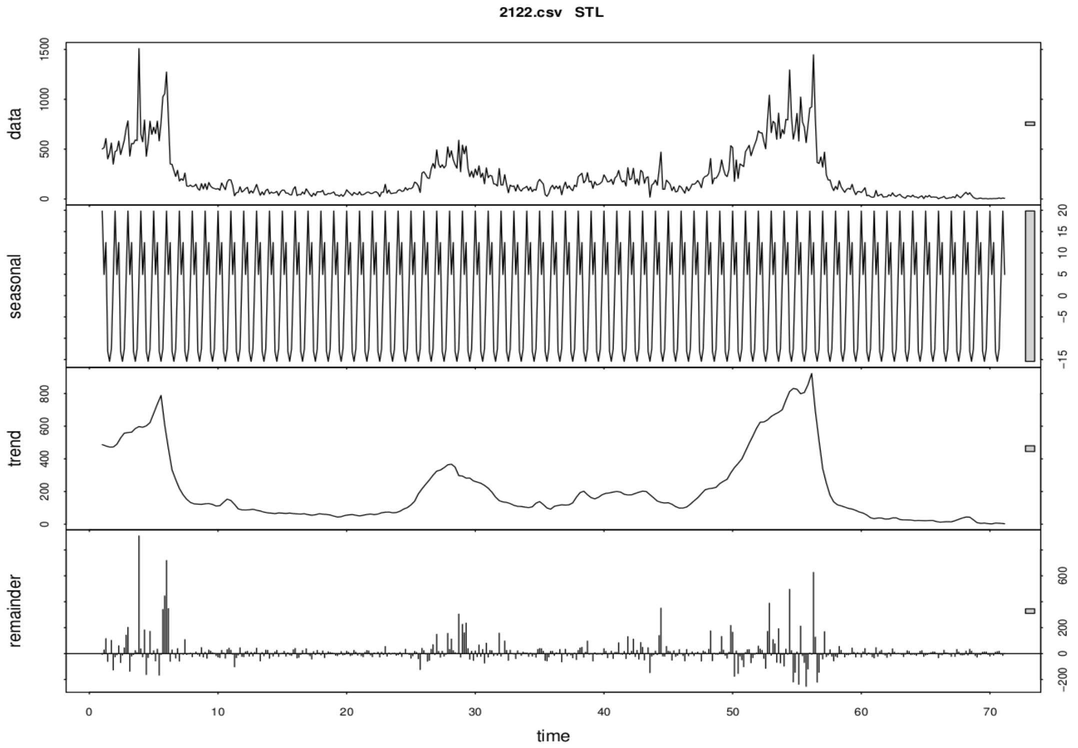

Fig. 8 : décomposition of sales for product 2122

There are two algorithms to decompose time series (daily sale amount,...):

- Classical decomposition by moving averages
- STL: Seasonal Decomposition of Time Series by LOESS

We choose STL because:

- it can handle any type of seasonality, and the seasonal component is allowed to change over time;
- it can be robust to outliers.

The external events such as football match, festivals do not have any impact on trend and a big impact on seasonality. So we are interested only in whether external events are correlated with the remainder part.

4.2 Matching events with activities via Rule Mining

In this section, we explain the process of matching events with activities using association rule mining. A rule consists of a premise X and a consequence Y, as well as some metrics describing the quality of the rule, such as support, confidence, and conviction. In our case, we mine the databases to find rules stating that an event type (the premise) impacts user activity (the consequence). Association rule mining aims at discovering dependencies between variables in a single large database. Because our interest lies in the correlation between events and user activities, this requires to join statistic outliers with events happening in the same period and same location.

We used the Classification Association Rules Mining algorithm (CMAR) to find the correlation between events and the remainder. We set the minimum support = 5 and the confidence = 75%, that mean the pair event-impact must occur at least 5 times, and the impact should be positive in 75% of cases or negative in at least 75% of cases.

Our event database is very limited at that point. So we tried to correlate sport events in US with the remainder parts of products. Our external event data consists of sport matches from 2012 to 2013 of the following competitions:

1. National Basketball Association
2. Formula One
3. American League
4. National Football League
5. National League

We base our method on an assumption. An event has impact on sale volumes only in the time window *(T - a, T + b)* where *T* is the event date, *a* and *b* are numbers of days. In our experiment, we suppose that event's impact on sales occurs no sooner than 5 days (*a* = 5) and no latter than 5 days (*b* = 5). We base our algorithm on observations:

Small values in the remainder part are likely daily variation of sales. So if an event appears around a day when the remainder value is small, we consider that this event has no impact on sales on that day. We keep only remainder values that are greater than 10% sale amount of the same day.

Since there are many events that happen on a day, we cannot say which one is the right one. An event need to occurs several times, and each time it occurs we see that there is an impact on the remainder the impact of an event on a product sale should be consistent. It should make sales increase or decrease, but not both. The impact should be before event or during event or after event exclusively. In other words, we need a way to associate event occurrences with remainder values.

Overall, this preliminary study showed that this approach, although promising at a conceptual level, requires a rich content from both the EventDB and the StatsDB, with dense datasets apt at supporting a confident extraction of rules. This prerequisite was obviously not fulfilled by the small dataset that we used so far in our experiments. Whether Streamline input was apt to match this goal is discussed in the concluding section.

5 Conclusion

5.1 Summary of objectives and results

This deliverable has presented the results of our investigations on contextualization requirements for the various use cases involved in STREAMLINE, and identified some of the potential benefits of introducing context data as side input of some machine learning algorithms

An important part of the initial effort of contextualization issues has been devoted to the study of events sources, and to experiments aiming at evaluating their quality and the potential volume of data that can be obtained. It appears that Web scraping achieves the right balance between relevancy (each use case can identify highly relevant sites), quality (the IMR wrapper delivers focused, structured data) and volume. This requires the production of wrappers, a rather costly operation since it involves, at least for the time being, annotation by experts. Improving the wrapper production towards a fully non-supervised process is a key to scale events capture.

The perspective of running context-aware data mining algorithm on streams of events and statistical time series establishes some strong requirements for the development of ML algorithms and their integration with Flink. The implementation of the contextualization engine, as it is currently designed, closely depends on the ability of our partners in Work Packages 2 and 3 to produce relevant methods.

5.2 Refactoring of W4: explanations and new objectives

After this initial phase of investigations, we proposed a re-orientation of Work package 4 that has been approved by the commission. The main reasons are exposed below, and derive from the above study.

5.2.1 Limits of the initial approach

First, is turns out that it is very hard to collect enough events to find a significant subset that matches with a user activity, given that this activity takes place in a specific location, at a specific time, and might be influenced by a bunch of factors, most of which falls out of the scope of a contextualization engine. During the use case study, we found that sources apt at providing dense coverage of relevant events are mostly highly specialized web sites, the list of which has to be supplied by the use case partners. Even though we developed during the first phase a scalable scraping technology, implementing a full crawling and extraction system dedicated to the discovery and scraping of event-related sites would have required considerable efforts.

A second, more important, issue is that matching events and user activity requires full access to the latter, including important features such as geolocation. This raises privacy issues, and restricts the deployment of a full-fledged architecture.

In the context of Streamline, after the departure of NMusic, the only remaining use case (Altice labs) were highly affected by those two issues. First, users descriptions did not feature the geolocation, which made uncertain the determination of event sources that could affect them, second Altice Labs did not want to disclose user activities. We considered that, in this situation, we would not be able to build, in the scope of the project, a contextualization system with enough confidence in the viability of the approach.

5.2.2 More information on the change of direction

On the positive side, this initial study has shown opportunities that can still be related to the contextualization concept. The main analytic technique that seems likely to be impacted by the knowledge of external events is the recommendation of content to users in online media distribution platforms. Recommendation is usually based on “internal” information obtained by tracking users activity, and on the description of the items that can be recommended. In many cases this description can be enriched from “external” sources. As soon as the item identity can be precisely obtained, we can find relevant and complementary data on the Web: reviews, user evaluation, technical documentation, etc. This is clearly the case for the “items” managed by two industrial partners: AlticeLabs with TV shows, and IMR with catalogs of products.

Therefore, the goal is now to use contextual information to support recommendation algorithms. Context here means no longer events but textual description of the recommended items, either part of the items description, or collected from external sources.

Recommendation can either use contextual information in conjunction with user profiles and activities, or rely only on contextual information to solve the cold start problem when a new user or new item enters the system. We will study the following use cases

- TV recommendation, with contextual information being TV programs, and possibly comments and reviews obtained from social media. (Altice Labs)
- Product recommendation, based on product description, and user comments extracted from web pages (IMR)

In both cases, the goal is to process textual data attached to items to infer similarity and boost the recommendation engine.