

MULTISENSOR

Mining and Understanding of multilingual content for Intelligent Sentiment
Enriched context and Social Oriented interpretation

FP7-610411

D7.6

Second Prototype

Dissemination level:	Public
Contractual date of delivery:	Month 24, October 31 th , 2015
Actual date of delivery:	Month 25, November 12 th , 2015
Workpackage:	WP7 System Development and Integration
Task:	T7.4 System development
Type:	Prototype
Approval Status:	Final Draft
Version:	1.0
Number of pages:	72
Filename:	D7.6_SecondPrototype_2015-11-12_v1.0.pdf

Abstract

This document describes the technical components and infrastructure of the advanced Second Prototype for the MULTISENSOR platform. It provides an overview of the demonstration application prototypes, the organisation and composition of the components (modules) and improvements with respect to the First Prototype (D7.4). The Second Prototype combines operational infrastructure improvements by replacing all dummy services with fully functional services.

The information in this document reflects only the author's views and the European Community is not liable for any use that may be made of the information contained therein. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.



co-funded by the European Union

History

Version	Date	Reason	Revised by
0.1	6/10/2015	Document initiation	A. Bilous (everis)
0.2	20/10/2015	Adding everis contributions	A. Bilous, E. Jamin (everis)
0.3	26/10/2015	Integration of all the contributions	A. Bilous, E. Jamin (everis) and Technical partners
0.4	27/10/2015	Internal review	A. Mas (everis)
0.5	28/10/2015	Reviewed version and feedback about the UC applications	D. Liparas (CERTH), M. Puigbo (PIMEC)
0.6	06/11/2015	Improvements of the UC applications and revision of the document	A. Bilous, E. Jamin (everis)
0.7	10/11/2015	Reviewed version and feedback about the UC applications	D. Liparas (CERTH), M. Puigbo (PIMEC), L. Blacha (PR)
1.0	12/11/2015	Final version	A. Bilous, E. Jamin (everis)

Author list

Organization	Name	Contact Information
EVERIS	Emmanuel Jamin	ejacques@everis.com
EVERIS	Andriy Bilous	andriy.bilous@everis.com
CERTH	Stefanos Vrochidis	stefanos@iti.gr
CERTH	Dimitrios Liparas	dliparas@iti.gr
CERTH	Anastasia Moumtzidou	moumtzid@iti.gr
CERTH	Ilias Gialampoukidis	heliassgj@iti.gr
PRESSRELATIONS	Leszek Blacha	leszek.blacha@pressrelations.de
ONTO	Boyan Simeonov	boyan.simeonov@ontotext.com
ONTO	Vladimir Alexiev	vladimir.alexiev@ontotext.com
LinguaTec	Reinhard Busch	r.busch@linguatec.de
EURECAT	Ioannis Arapakis	arapakis@eurecat.com
UPF	Gerard Casamayor	gerard.casamayor@upf.edu

Executive Summary

D7.6 of the MULTISENSOR platform presents the Second Prototype (SP) description combining operational infrastructure improvements by replacing all dummies with fully functional services. Our main concentration for the SP is mainly based on offline (CEP modules, crawling infrastructure and repositories) and online (Online services used for different Use Cases) modalities.

The prototype describes the main services and their integration into the User interface (UI).

This document provides a technical overview of the development and integration of the Second Prototype of the MULTISENSOR platform.

Abbreviations and Acronyms

CI	Continuous Integration
CMR	Central Multimedia Repository
CNR	Central News Repository
DB	DataBase
EBS	Elastic Block Storage
EC2	Elastic Compute Cloud
ECU	Elastic Compute Unit
FP	First Prototype
FTP	File Transfer Protocol
FTS	Full-Text Search
HTTP	HyperText Transfer Protocol
JPEG	Joint Photographic Experts Group
JSON	JavaScript Object Notation
MPEG	Moving Picture Experts Group
NER	Named Entity Recognition
OPS	OperationS repository
PPA	Personal Package Archive
OWL	Ontology Web Language
RDF	Resource Definition Framework
REST	Representational State Transfer
SIMMO	Socially Interconnected and MultiMedia-enriched Object
SOA	Service Oriented Architecture
SPARQL	SPARQL Protocol And RDF Query Language
UC	Use Case
UC(x)	Use Cases 1, 2 or 3
SP	Second Prototype
UCS	Universal Character Set
UI	User Interface
UTF	UCS Transformation Format
W3C	World Wide Web Consortium
XML	eXtensible Markup Language

Table of Contents

1	INTRODUCTION	8
2	SECOND PROTOTYPE ARCHITECTURE.....	9
2.1	Global view.....	9
2.2	Status of the First Prototype	10
2.3	Objectives of the Second Prototype.....	10
2.4	Status of the Second Prototype.....	11
3	INTEGRATION FRAMEWORK	13
3.1	Approach.....	13
3.2	High-level view	13
3.3	Offline modality	14
3.3.1	Crawlers.....	14
3.3.1.1	Media collector (PR crawler).....	14
3.3.1.2	Site crawler (EURECAT crawler).....	15
3.3.1.3	Twitter collector	17
3.3.1.4	DW API.....	18
3.3.2	Repositories	19
3.3.2.1	Central News Repository (CNR).....	19
3.3.2.2	CMR Repository.....	19
3.3.2.3	OPS Repository.....	20
3.3.2.4	RDF Repository.....	20
3.3.3	Content extraction pipeline (CEP).....	20
3.3.3.1	Language detection	21
3.3.3.2	Translation	22
3.3.3.3	Named Entities recognition.....	22
3.3.3.4	Concept extraction	23
3.3.3.5	Dependency parsing	24
3.3.3.6	Relation extraction	24
3.3.3.7	Sentiment analysis.....	25
3.3.3.8	Extractive summary.....	26
3.3.3.9	Classification	27
3.3.3.10	Context extraction.....	28
3.3.3.11	Audio extraction and ASR	28
3.3.3.12	Concept and Event detection.....	29
3.3.3.13	Indexing	30
3.3.3.14	RDF Validation.....	31
3.3.3.15	Storing RDF.....	31
3.3.4	Content Alignment Pipeline (CAP)	32
3.3.5	Social Media Analysis Pipeline (SMAP).....	33

3.3.6	Platform monitoring.....	35
3.3.7	Platform testing services	36
3.4	Online modality.....	37
3.4.1	Business Shared Services.....	37
3.4.1.1	Content delivery.....	37
3.4.1.2	Semantic search	38
3.4.1.3	Topic-Event detection.....	40
3.4.1.4	Similarity search	40
3.4.1.5	Machine Translation.....	41
3.4.1.6	Abstractive summary.....	42
3.4.1.7	Contributor analysis.....	42
3.4.2	Other Online Services.....	43
3.4.2.1	User profile	43
3.4.2.2	Reference Data.....	44
3.4.2.3	Decision support.....	46
4	PROTOTYPE APPLICATIONS	48
4.1	UC1: Journalism Use Case	48
4.2	UC2: Media Monitoring Use Case.....	54
4.3	UC3: SME internationalisation Use Case	60
5	CODE ORGANISATION	67
5.1	Source tree layout (D7.4 updates).....	67
5.2	Continuous integration environment.....	68
5.3	Packaging	68
5.3.1	Java modules	68
5.3.2	Node.js modules	69
6	INFRASTRUCTURE	70
6.1	Current farm (D7.4 updates)	70
6.1.1	Mscrawler1	70
6.1.2	Msgrinder1	70
7	DEMONSTRATOR URLS AND INFORMATION.....	71
8	SUMMARY AND CONCLUSIONS	72

1 INTRODUCTION

In D7.1, a general roadmap and technical vision for the implementation of the MULTISENSOR platform was established. The user and non-functional requirements in D8.2 and the technical vision were combined in D7.2 to define the global architecture of the system and its subsystems, workflows and interfaces.

The “walking skeleton” for the technical roadmap laid out in D7.1 is presented below:

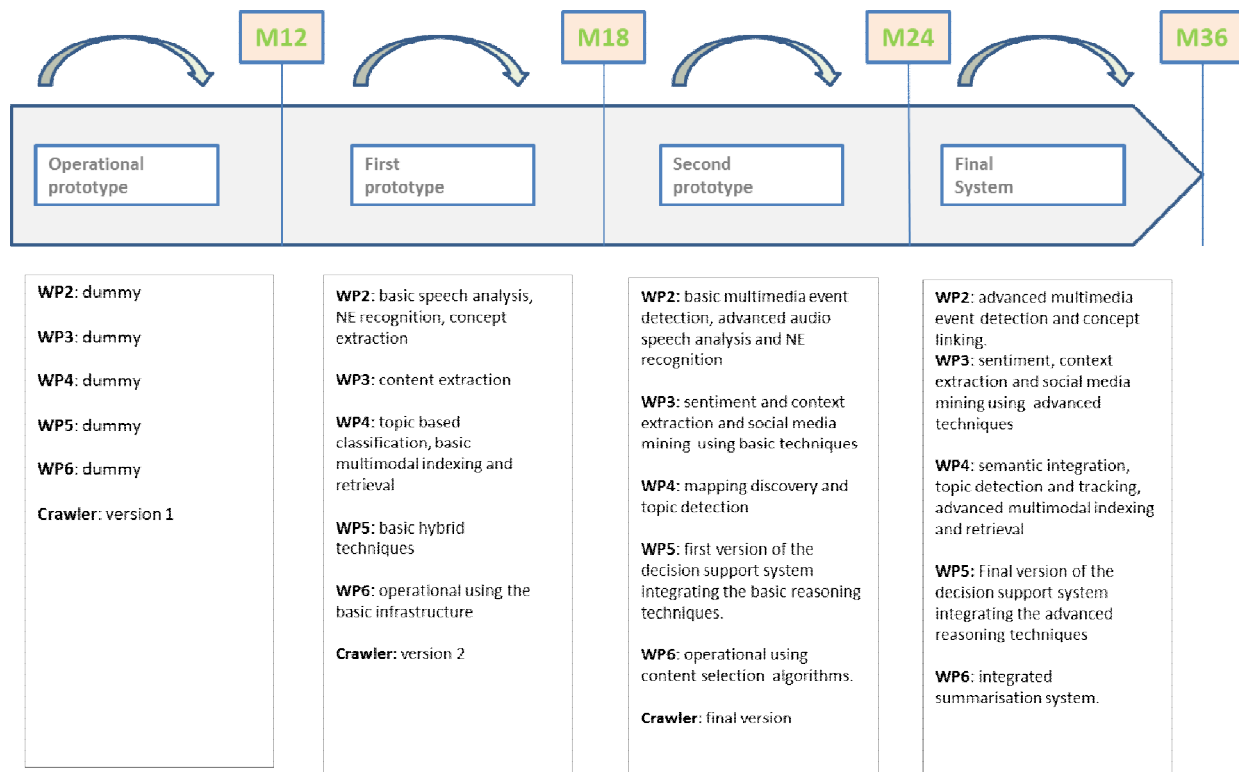


Figure 1: Technical roadmap

The purpose of this document is to provide a brief technical reference for the D7.6 deliverable, which is the third technical milestone of the project (M24). D7.6 contains the implementation description of the major services, processes, and workflows, the integration framework to connect those implementations and the update of the UI regarding the functional integration of the different services.

Section 2 contains a high-level technical overview of the prototype.

Section 3 contains a description of the integration status of the framework.

Section 4 contains a description of the online applications UC(x).

Section 5 details the technical infrastructure hosting the prototype.

Section 6 contains links and details for accessing the demonstrator application for reviewers.

Section 7 presents a brief summary and conclusions.

2 SECOND PROTOTYPE ARCHITECTURE

2.1 Global view

The global architecture for the MULTISENSOR platform has been discussed at length in D7.1, D7.2 and D7.4.

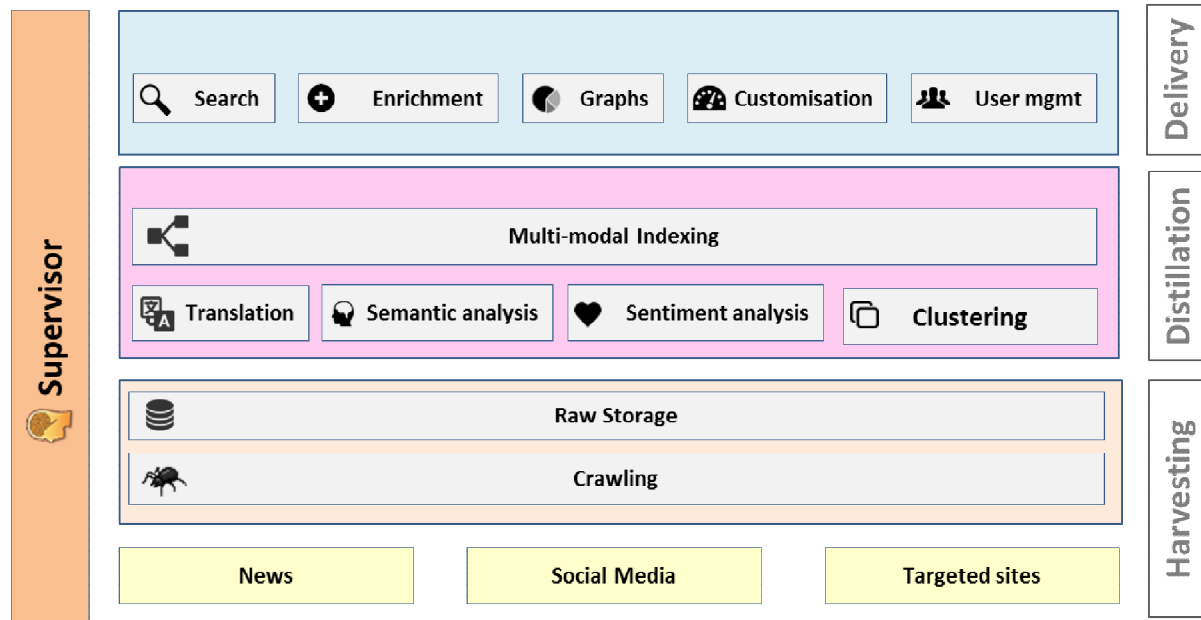


Figure 2: MULTISENSOR logical architecture

The MULTISENSOR architecture (see Figure 2) is based on a SOA approach and encompasses two discrete modalities: offline - asynchronous processing of harvested data (see D7.2, section 4.2.2), and online - synchronous retrieval, delivery and exploitation of the analytical data (see D7.2, section 4.2.3).

D7.6 explains the current status of the system in terms of repositories, services, processes and workflows of the Second Prototype.

One of the most important efforts was focused on the services integration of the Offline modality, in particular CEP services that perform analysis routine on crawled articles in order to populate the main repositories with analytics data. In this way, analytical data are delivered to the online services and the UI of the three UC applications is improved.

2.2 Status of the First Prototype

In D7.4, the status of the First Prototype was presented. A quick summary of the elements that were part of it is provided below.

- The system was migrated to a most important server and the configuration was improved.
- The four repositories of the Data Layer were in place.
- The two crawlers were implemented, however only one provided by PressRelations (PR) was deployed.
- All the main services were implemented. Only a few of them were provided in a baseline version.
- The different services were integrated into the platform, i.e. they can interact between themselves, store and collect data from the different repositories.
- The RDF repository was not populated with the extracted knowledge produced by the CEP.
- The UI for the three Use Cases was improved with the interaction of the available online services.

2.3 Objectives of the Second Prototype

The Second Prototype is an improvement of the First Prototype. In this prototype version, all Online and Offline services are expected to be delivered in their fully functional and stable versions and the knowledge base must be populated with reasonable amount of textual and multimedia SIMMOs. Here is the list of the goals to be achieved for the first SW development cycle of the project:

- Operational architecture including the full prototype versions of the individual modules:
 - Content extraction: basic multimedia event detection advanced audio speech analysis and NE recognition.
 - User-centric content extraction: sentiment and context extraction and social media mining using basic techniques.
 - Content integration and retrieval: mapping discovery and topic detection.
 - Reasoner and decision support: first version of the decision support system integrating the basic reasoning techniques.
 - Information production: operational using content selection algorithms.
- Data wrappers and crawlers (final version).
- Delivery of a report on the second round of formal and user-oriented evaluation of all modules.
- Report on dissemination, standardisation, and basic exploitation plan.

2.4 Status of the Second Prototype

The Second Prototype is the complete integration of all services and specific components like crawlers, repositories and use cases. It is composed of the online and offline modalities.

- The offline modality, as shown in Figure 3, analyses all the information that is coming from the crawlers scheduled by supervising script and is stored in the content repository (CNR).
- The Content Extraction Pipeline (CEP) is fully functional and can process multilingual textual and multimedia data from content repository (CNR) with further saving processed information into the knowledge base (RDF).
- The Content Alignment Pipeline (CAP) is implemented and integrated as a baseline version.
- The Social Media Analysis (SMAP) is analysing data, which is coming from a Twitter collector, with further saving to the knowledge base (KB) repository.

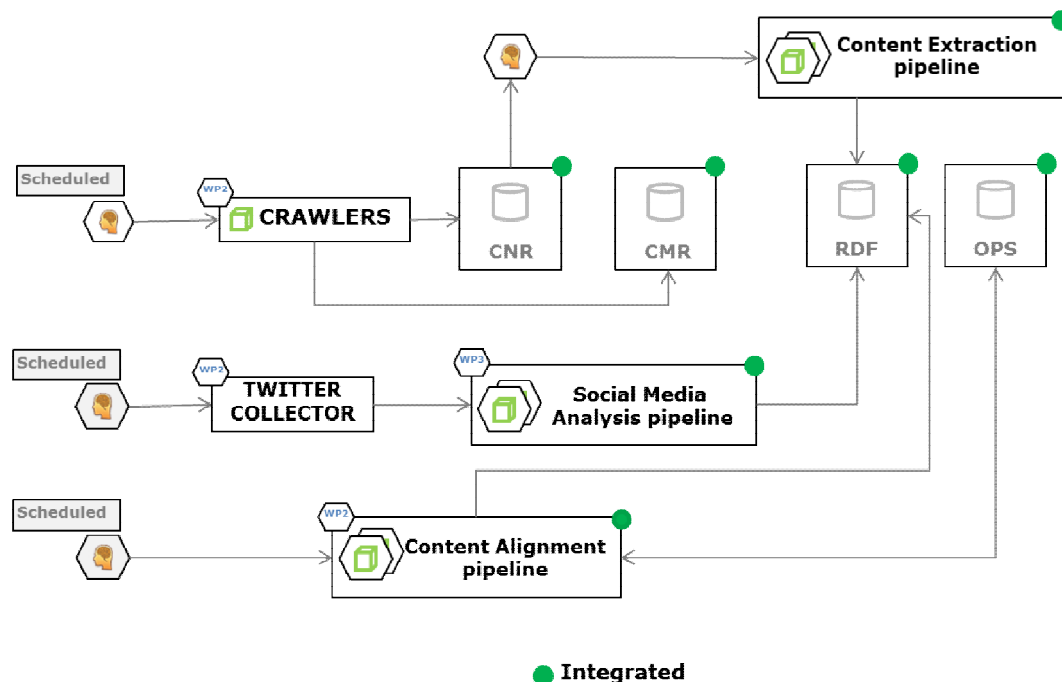


Figure 3: Second Prototype – Offline modality architecture

The online modality is the live connection between the user interfaces functionalities and the access to the knowledge available in the different repositories. As shown in Figure 4, most of the services are integrated and connected with the repositories. Now the RDF repository is populated (still in progress). Thus, the search functionality has been switched from CNR to RDF.

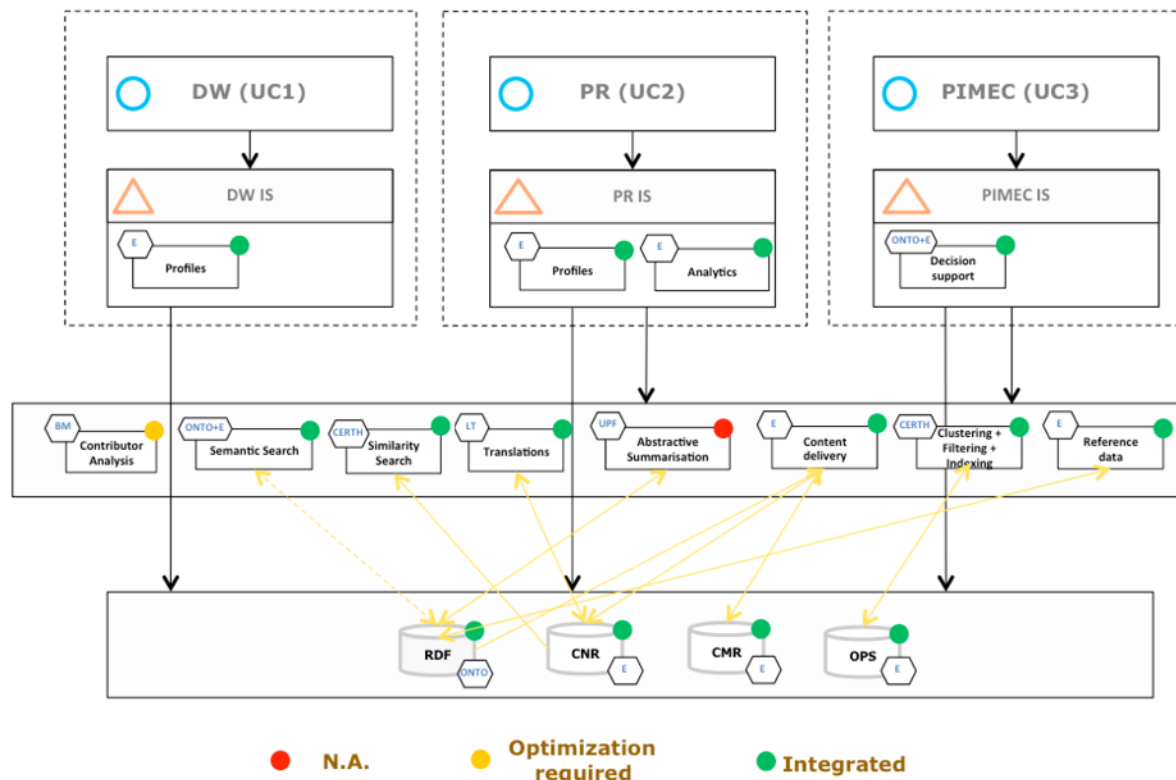


Figure 4: Second Prototype – Online modality architecture

During the development process, many challenges like integration between different technologies, component interconnection, shared usage for some functionalities, etc. were tackled. Many technical issues regarding dependencies between those services and other libraries were addressed, in order to obtain a stable and coherent integration with analytical data.

In D7.6, the result of the Second Prototype development can be summarised as follows:

- All the main services have been implemented and integrated into CEP:
 - Basic multimedia event detection
 - Audio Extraction and normalisation
 - Automated Speech Recognition (ASR)
 - NER
 - Sentiment Analysis (SA)
 - Context Extraction
 - Social media mining using basic techniques
 - Relation Extraction service performance optimised
- The Knowledge Base (KB) has been populated
- Additionally, the following services have been developed:
 - RDF validation
 - Content duplicate detection
 - Text normalisation for all data before processing
 - Introduced quality parameter for the data
- UI for the three Use Cases (UCx) are improved with the interaction of the online services

3 INTEGRATION FRAMEWORK

3.1 Approach

For the Second Prototype, all the services and the repositories are deployed, integrated and fully functioning. The new services are provided in a baseline version, and the ones that were introduced for the First Prototype were improved and provided in an advanced version.

Significant improvements in the development infrastructure were made in comparison to the First Prototype, by automating processes previously performed manually (testing, compilation, execution and deployment of the services).

Two environments for all services were developed in order to separate production and development test environments.

Jenkins (the automated build and deployment system) was configured.

3.2 High-level view

The logical layers of the MS architecture are represented in Figure 5. No modifications were realised on this architecture view for the Second Prototype.

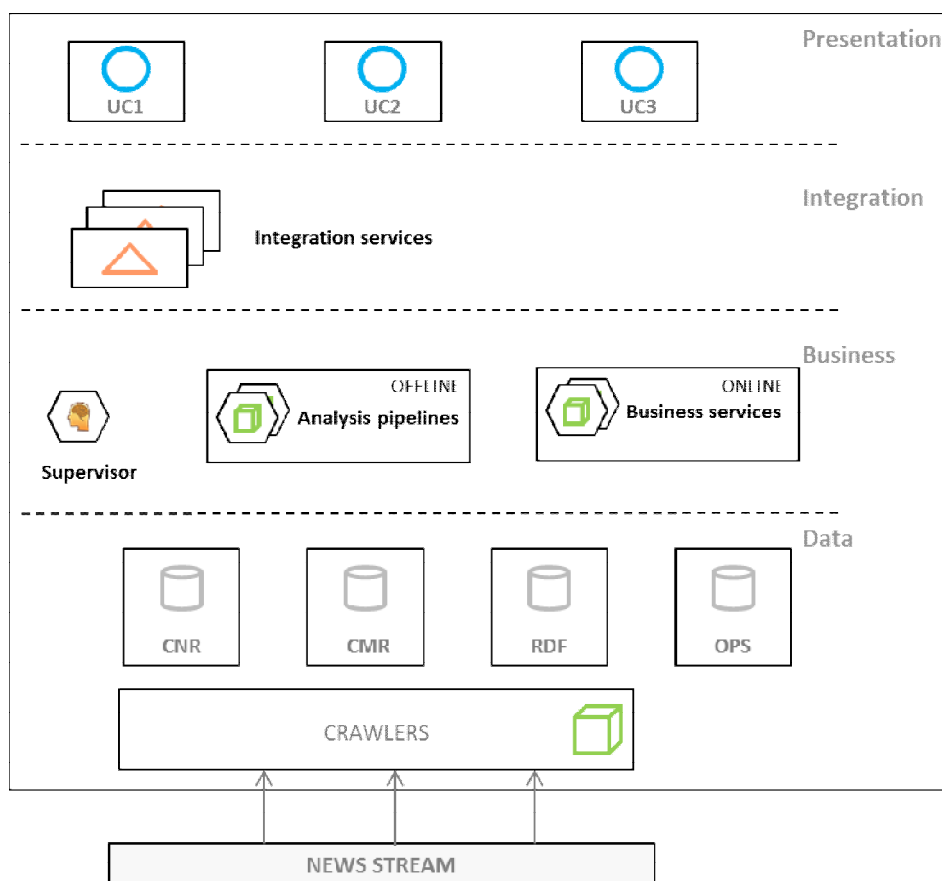


Figure 5: Second Prototype MULTISENSOR high-level view

The Second Prototype developments were focused on the improvements and integration of the Offline and Online modalities services, improving the crawling content quality, extending the crawling process on the multimedia dimension, continuing the intensive population of

the KB repository with analytic data and improving the integration of the new and already existing online services.

3.3 Offline modality

3.3.1 Crawlers

The First Prototype contained two crawlers provided by PR and EURECAT, implemented and integrated.

Crawled data provided by EURECAT and PR crawlers as a JSON endpoint, being duplicate checked, indexed and stored into CNR. However due to the CEP sensitivity to crawled contents it was decided to use data with quality information additionally performing data normalisation on CEP level. This fact limits the crawlers usage, until the required modification made by respected partners.

The PR crawler has come up with a solution to provide requested functionality with quality information. The Quality parameter is assigned manually to the source, all articles for this source were considered as verified and CEP safe. For the Second Prototype, we used articles provided by the PR crawler due to the issue described above.

The Second Prototype required multimedia content being crawled, as well as social information. This issue was solved partially with help from PR crawler and by implementing the Deutsche Welle (DW) API. Data from the abovementioned crawlers are being stored into CMR. The Twitter Collector performs the Social media crawling.

3.3.1.1 Media collector (PR crawler)

As described in D7.4, PR is crawling international news websites via their proprietary crawling technologies and provides news articles for each use case based on specific keywords. Keywords have been decided upon by the use case partners.

Additionally to textual information from news websites, PR also provides links to multimedia content. The number of websites, from which multimedia content can be extracted has been increased for the Second Prototype. But not for all websites can multimedia content be extracted, due to the heterogenous structure of the websites that are being crawled.

Modifications to the proprietary crawlers have been successfully made by PressRelations. The extraction function is now able to not only extract links to images within news articles but also links to video and audio files. These links are also provided via the PR API.

Finally, a quality parameter (as mentioned in 3.3.1) has been introduced and is provided via the PR API in order to filter content of high quality that can be processed through the complete pipeline. The quality parameter is assigned manually by the PR media team depending on the quality of the extraction results. From an initial set of news sites that provide content extracted with high quality the number of news sites has been significantly increased. Now for almost 700 news sites the content can be extracted with very good quality.

Further improvements of the Site Collector are not planned at this stage.

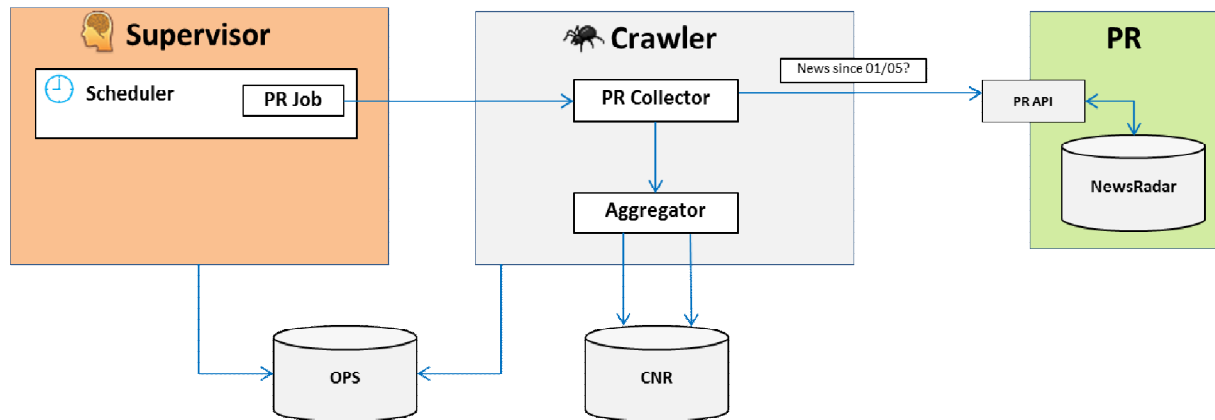


Figure 6: Site collector (PR)

3.3.1.2 Site crawler (EURECAT crawler)

Until the First Prototype, the site crawler was called Yahoo! crawler. Due to the change of the partner's name, this crawler is now called EURECAT crawler.

Discovering and downloading newly published articles in news web sites in a timely manner requires implementing a web crawling architecture that involves a number of components. To this end, the EURECAT crawler applies a modular architecture in which existing big data processing technologies (Hadoop, HBase, Nutch, Solr), whose robustness and efficiency have been previously proven, are combined in a meaningful manner for the discovery and download of news articles. Hence, it offers efficient and scalable web crawling functionality.

The crawling component, Apache Nutch¹, lies at the heart of the crawler's architecture and is responsible for three main tasks: (1) read URLs from a priority queue and download the corresponding web pages from the Web, (2) parse the content of the downloaded pages to discover new URLs, and (3) store the web pages in a data store. Nutch supports distributed crawling and it can scale to crawl millions of web pages. Crawling is performed through MapReduce² jobs running on the Apache Hadoop³ platform.

Nutch stores two types of output to HDFS (the default storage system in Hadoop): (1) extracted links, and (2) downloaded pages. HDFS provides good performance for bulk reading of the data, but it does not provide random access. Therefore, for the downloaded pages, we use a separate data store, Apache HBase⁴. HBase supports random, real-time read/write access to Big Data (e.g., billions of records with millions of fields) on clusters of commodity hardware. In our architecture, the downloaded pages are stored in HBase. After the crawling session is completed, a small Java code is executed to read all web pages and meta-data from HBase. This code writes the final output of the crawler into a text file on the local storage in JSON format. The entire crawling architecture is illustrated in Figure 7.

Previously, EURECAT developed the crawler for collecting web pages given a seed list of news provider domains. The seed list was provided by the user partners and there can be a potentially different seed list for each use case scenario. The crawler was scheduled to run

¹ <http://nutch.apache.org/>

² http://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html

³ <http://hadoop.apache.org/>

⁴ <http://hbase.apache.org/>

on a daily basis collecting news articles from the respective media sources. However, due to the CEP sensitivity to the crawled content, it was decided to use only data with a certain level of quality (quality information indicator) in addition to performing data normalisation on CEP level. At that point, the EURECAT crawler did not provide this quality indicator and that limited the crawler's usage. Moreover, given the multilingual and multi-heterogeneous character of the crawled domains, an evaluation was necessary to establish a set of domains that can be effectively crawled. Finally, the crawler was operating using the default setup, without any further optimisation of the crawling parameters and functionality.

Since the first version of the EURECAT crawler, the following modifications were applied in preparation for the Second Prototype. First, we validated all the domains that appeared in the original seed list and discarded invalid cases. We now have in total 477 domains. However, after a manual inspection, not all domains were identified as suitable for crawling due to errors produced by either dynamic content or character encoding. We further performed code modifications and added the `c_quality` indicator to the crawler output. Then we performed a quality check for a subset of web domains and selected several domains that generated a relatively clean and error-free output. Finally, we looked into different extractors of news content. One issue identified is that certain articles are not extracted fully. The reason for that is that the library we use (Boilerpipe⁵) has to identify for each page the news content. However, when the news content is filled with ads, images, and other irrelevant elements, it is not trivial to determine where the onset and offset of the news article's body. The Boilerpipe library offers several extractors⁶ and after experimenting with all of them, we concluded to the one that achieves the best possible article extraction.

⁵ <https://code.google.com/p/boilerpipe/>

⁶ <http://www.l3s.de/~kohlschuetter/publications/wsdm187-kohlschuetter.pdf>

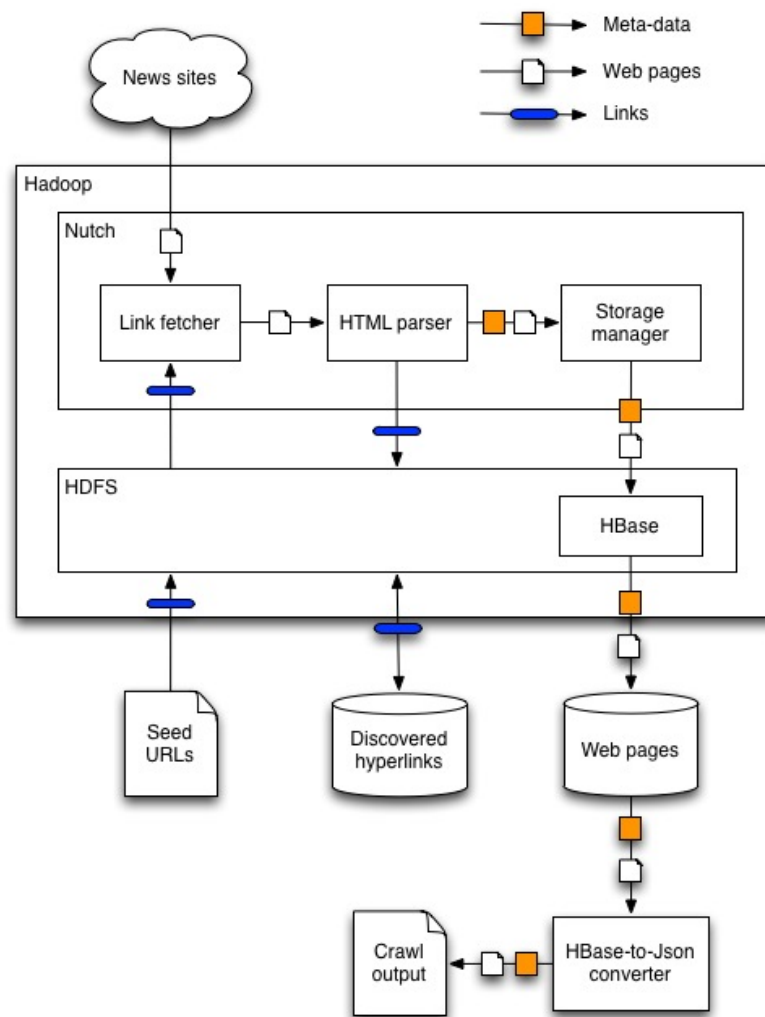


Figure 7: Crawling architecture of Yahoo crawler.

During the parsing of the output from the initial crawling cycles of sources with $c_quality = "1"$, we several memory issues that were blocking the processing of the output stored in HBase and its conversion to JSON format. After examining the produced error reports, we increased the swap space on the server from 0 to 4Gb. In addition, the HBase heap memory space was reduced since it was another potential solution to this problem. Finally, we modified the Nutch configuration (nutch-site.xml -) and changed the values of certain parameters (e.g., number of fetcher threads, number of threads per host, distribution of URLs to fetch) to optimise the crawling process. We estimate that the observed performance (pages per minute), after the above modifications, improved by at least 10% over the initial crawler setup.

3.3.1.3 Twitter collector

The Twitter collector comes from a previous project, namely SocialSensor (FP7-287975), during which Stream Manager was implemented. Stream Manager contains a number of APIs that collect incoming content relevant to a keyword, a user or a location from a set of social streams (Twitter, Facebook, Instagram, etc.). The Twitter collector gathers Twitter posts for a set of high-activity hashtags that are pre-specified for each Use Case separately. These posts, as well as information regarding the author, the associations found inside the posts (between users, between user and webpages, etc.) are then stored into a MongoDB

database. The Twitter collector runs every 30 minutes for the hashtags of each Use Case and if new posts are found they are stored inside the database. It should be noted that Twitter has a limit of 180 requests per 15 minutes.

Currently, and based on simple observation the average number of posts stored per day into MongoDB is 2000 posts for all Use Cases. However, this number is not fixed and it depends highly on the hashtags used and on how popular they are. The current total number of posts stored into MongoDB is approximately 30000. Finally, it should be noted that the Twitter posts gathered by the Twitter collector and stored into MongoDB are fed as input to the services (Influential User detection and Community detection) of the Social Media Analysis Pipeline (SMAP).

3.3.1.4 DW API

The DW API gives access to articles, video, audio and image galleries. For the Second Prototype, the DW API is used as source of video and audio content.

DW uses a REST API, the output is returned in JSON format and being saved in to CMR repository. As this is very first version currently CMR have 8 videos.

The latest list of the access point mirrors can be found on the DW website: <http://www.dw.com>. The configuration feed serves as main entry point:

<http://www.dw.com/api/config/init?product=dwapp&platform=web&version=2.1.0>

Media items can be of various types: video, audio or pictures. The URL to call is as follows: {baseApiUrl}/api/list/mediacenter/{languageId}?pageIndex={pageIndex}

This URL gives access to a list of media teasers. A teaser is a short summary on the media item that can be shown in a list. As an example, to read all recent media items in English, the URL would be <http://www.dw.com/api/list/mediacenter/2?pageIndex=1>

The returned JSON contains a list of media items where for each item, summary information is provided, such as:

- displayDate
- duration (in seconds)
- images in various resolutions
- the name (i.e. the title)
- teaser text -> a short description
- the type -> VideoTeaser, AudioTeaser, GalleryTeaser
- paginationInfo
- the reference -> the reference to the details related to the given item

The media item details are accessed through the URL provided through the teaser URL discussed above, more precisely the reference -> URL:

The reference -> URL gives access to the details for a given item:

```
▼ "reference": {
  "id": 18530364,
  "name": "Ordinary Greeks brace for more cuts",
  "type": "VideoRef",
  "url": "http://www.dw.com/api/detail/video/18530364"
},
```

So in this case, the media item's details can be found through the following URL: <http://www.dw.com/api/detail/video/18530848>. A similar approach is used for the audio content.

3.3.2 Repositories

The Second Prototype provides four different repositories:

- The CNR to store all the crawled documents
- The CMR to store the multimedia elements related to the crawled documents
- The OPS to store the Operations information like the User Profile information
- The RDF to store all the KB produced by the offline modality and some relevant Linked Open Data datasets.

All mentioned repositories were already available in the First Prototype (FP). The update of repositories is presented in the following sections (see 3.3.2.1).

In the SP, the four repositories are implemented, fully functional and accessible through the local access in the server or the REST service endpoint. For each access method, other services are able to store, update or collect data.

3.3.2.1 Central News Repository (CNR)

The Central News Repository (see D7.2, section 4.2.4.1) is the raw storage dump for the Crawlers (Site and Media collectors). The CNR implemented as an ElasticSearch instance, allowing storing “bid data” without degradations in performance.

Currently CNR populated with approximately 80.000 multilingual with high quality:

- 23.000 English articles
- 4.000 Spanish articles
- 8.000 French articles
- 8.000 German articles
- 18.000 Bulgarian articles
- 12.000 Italian articles

User Cases distribution is the following one:

- UC1 – 18.000 Articles
- UC2 – 62.000 Articles
- UC3 – 7.000 Articles

For the SP, the CNR repository has been re-indexed with additional fields, due to inefficiency of previous structure.

3.3.2.2 CMR Repository

The Central Media Repository (CMR) is the storage of the source multimedia content (video, images and audio) collected by the harvester (see D7.2, section 4.2.4.2).

For the SP, the CMR has been fully integrated into the CEP processing.

The CMR is collecting output from PR (API) crawler with *url_multimedia* fields. Inside this field, there is a list of URLs that contain the related images, videos and audio files for a specific article retrieved by the Crawler. Customised folders are created for every article and save the multimedia content in file system's directory.

An intensive population of the CMR and an integration of the EURECAT (API) crawler with the outlinks field are planned for the next deliverable.

3.3.2.3 OPS Repository

The Operations Repository (OPS) provides fast, read/write structured data storage for any systems in the platform that require it (see D7.2, section 4.2.4.4).

The OPS is implemented as an instance of Mongo DB only for user related management in all UCx. It permits to store the information like user credentials to login the web application, the user session, profiles preferences and other information related to user management.

For the Second Prototype, the following changes have been performed in OPS:

1. Relevant/Irrelevant fields added used for profiling service
2. User language field was added, which used for language interface settings

3.3.2.4 RDF Repository

The RDF repository holds all MULTISENSOR data in semantic format. This includes the ontologies, external datasets like DBpedia, Geonames and many statistical indicators from World Bank and Eurostat. The main function of the triplestore is to store the SIMMO objects. This data form the MULTISENSOR knowledge about the world. It is based on GraphDB-Enterprise which is a high-performance, clustered semantic repository created by Ontotext.

For the First Prototype, GraphDB-SE were used, but after that decision were made that the project will use GraphDB as search engine too, so this provoke the migration to GraphDB-Enterprise which is a high-performance, clustered semantic repository created by Ontotext.

At the moment, we have two main repositories. The first one is called – “multisensor-dbpedia” and hold the DBpedia, Geonames and statistical indicators data. The second one is called - “multisensor-test” and holds the SIMMO objects. The volumetric of these repositories are:

- multisensor-dbpedia:
 - number of statements – 920,942,267
- multisensor-test:
 - number of statements – 87,477,695
 - number of contexts (SIMMO objects) – 9920

After many iterations, the quality of the data was significantly improved and now we have a stable semantic data structure.

3.3.3 Content extraction pipeline (CEP)

During work on the Second Prototype, the focus was on improving CEP services, developing and performance optimisation.

Significant improvements in development infrastructure were made in comparison to the First Prototype, by automating processes previously performed manually (testing, compilation, execution and deployment of the services).

The CEP testing utilities, allowing setting custom input and access to CEP outputs, were provided.

The Content Extraction Pipeline analyses all the harvested news that are stored in the CNR.

The CEP chains execution of several language analysis services to generate intelligence from the contents of a text article or a video segment, from base syntactical analysis to sentiment analysis to automated clustering and classification of the contents. The multimedia content is separated in four modalities: text, image, video, and audio. The most important one is the textual. For the audio and video, the textual information is extracted and then the multimedia resources are described as a textual document.

In Figure 8, the CEP diagram for the Second Prototype is displayed.

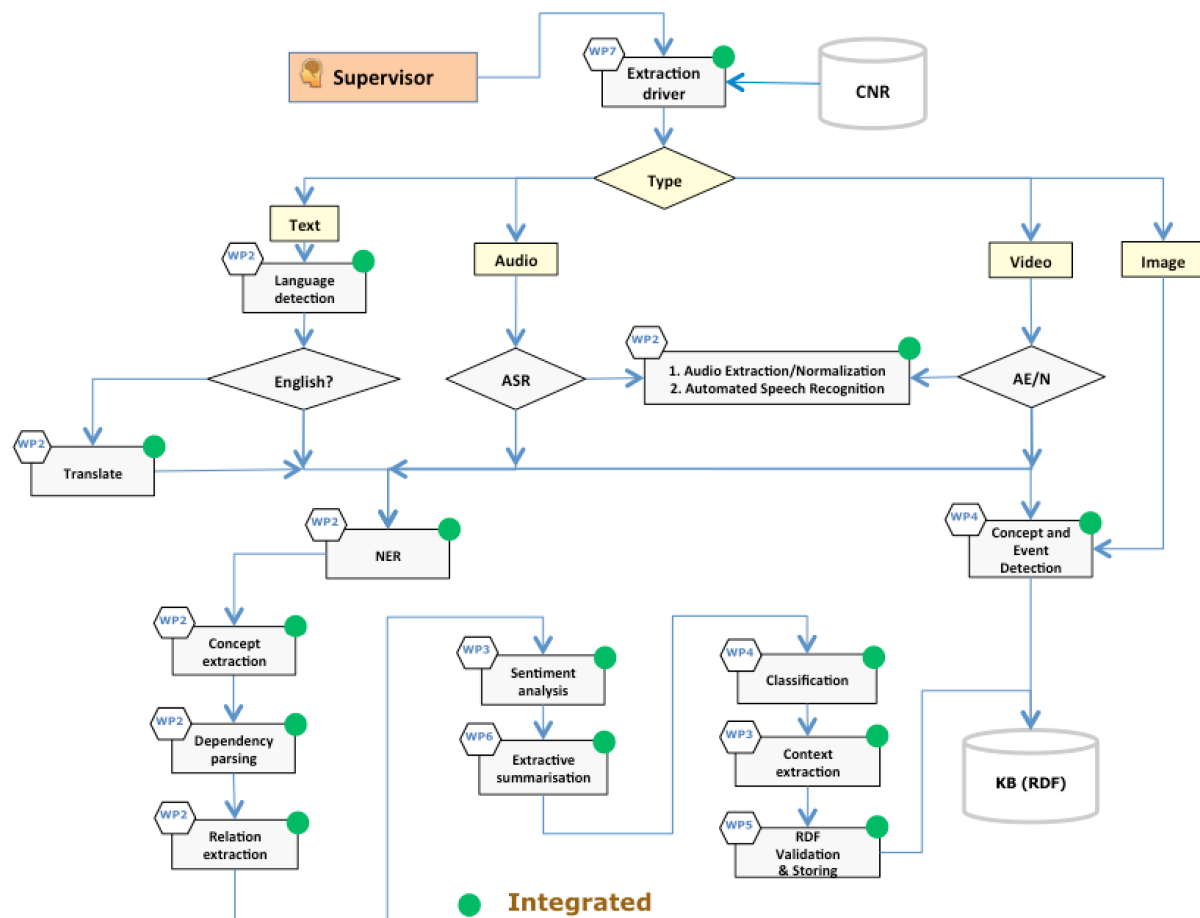


Figure 8: Content Extraction Pipeline of the Second Prototype

3.3.3.1 Language detection

The language detection service recognises language of a text.

Service name	Language Identifier
SVN artefact / remote endpoint	http://services.linguatec.org/rest/lang/identify
Functional Description	Recognises language of a text
Deployment status	Remotely deployed
P1 version	Version 1.0, fully functional
P2 version	Version 1.1, fully functional

P2 improvements	Extended to 18 additional languages for better discrimination purposes
Maximum amount of queries per second	7 queries per second
Supported languages	MULTISENSOR languages: English (en), French (fr), German (de), Spanish (es) and Bulgarian (bg). plus 18 additional languages: Arabic (ar), Chinese (zh), Czech (cz), Danish (da), Dutch (nl), Finnish (fi), Greek (el), Hebrew (he), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Pashto (ps), Polish (pl), Portuguese (pt), Russian (ru), Swedish (sv), and Turkish (tr)
Known issues	Stable performance has been reached
Next steps	None

Table 1: Integration description of the Language identifier service

3.3.3.2 Translation

The Translation service receives as input text in source language and translates it into specified target language.

Service name	Machine Translation
SVN artefact	http://services.linguattec.org/rest/lang/translate
Functional Description	Receives as input text in source language and translates it into specified target language.
Deployment status	Remotely deployed
P1 version	Version 0.8, fully functional
P2 version	Version 1.0, fully functional
P2 improvements	Improved translation quality resulting from: a) better homogenisation of training corpus b) reduction of the amounts of unknown words c) tuning of model parameters
Maximum amount of queries per second	7 queries (sentences) per second
Supported languages	English to/from French; English to/from German; English to/from Spanish; English to/from Bulgarian.
Known issues	Continued testing
Next steps	Continuation of quality and performance tuning

Table 2: Integration description of the Translation service

3.3.3.3 Named Entities recognition

Named Entities recognition is a service that recognises the names of persons, the locations, and the organisations & companies, as well as date, amount and measurements.

Service name	Named Entity Recognition
---------------------	---------------------------------

SVN artefact	http://services.linguatec.org/rest/ner/recognize
Functional description	Identifies names (named entities), i.e.: persons, locations, organisations & companies, date, amounts, measurements
Deployment status	Remotely deployed
P1 version	Version 0.8, fully functional
P2 version	Version 0.9, fully functional
P2 improvements	<ul style="list-style-type: none"> • Adaptation of lexicons towards the three use case scenarios • Extensions of grammars to a more context-sensitive approach • Output given in the CEP workbook format
Maximum amount of queries per second	7 queries per second
Supported languages	English, German, Spanish
Known issues	Continued testing
Next steps	Continued improvement of NE grammar and coverage Extension of language coverage to French and Bulgarian

Table 3: Integration description of the Name Entities Recognition service

3.3.3.4 Concept extraction

Concept extraction extends Named Entities Recognition by detecting mentions to additional entities denoted with common names rather than proper names. The concept extraction service performs term detection and term disambiguation. Term detection consists in detecting expressions used to denote concepts relevant to the domain at hand, i.e. terms. This is achieved by compiling ranked lists of potential terms resulting from the statistical analysis of use case-specific corpora. The concept extraction service uses these lists to discern between relevant terms and less informative concepts, so that only the latter are annotated. Term disambiguation tries to assign each annotated term with an entry in a lexicalised body of knowledge, i.e. a database where its entries are indexed with their lexicalisations. When multiple entries are found for the same term, the service must disambiguate between them.

Service name	Concept extraction service
SVN artefact	ms-svc-extr
Functional description	Annotates texts with mentions of concepts in BabelNet
Deployment status	Deployed
P1 version	Version 0.2, baseline version
P2 version	Version 0.3
P2 improvements	Term detection has been implemented using terminology-lists obtained by applying the TermRaider GATE plugin. NIF RDF annotations validated by Ontotext.

Maximum amount of queries per second	The current performance of the service is very fast, but future deployment of a disambiguation strategy may slow down the service.
Supported languages	English
Known issues	Current terminology list is of poor quality (see D2.3) and for texts in English only. No disambiguation.
Next steps	Obtain high-quality and (if corpora is available) multilingual terminology lists for each use case. Deploy a disambiguation mechanism against the 2.5.1 release of BabelNet being developed by UPF.

Table 4: Integration description of the Concept extraction service

3.3.3.5 Dependency parsing

The dependency parsing service annotates texts with the syntactic structure of their sentences. Two different structures are annotated, (i) a surface syntactic structure indicating language-specific grammatical relations between all words in a sentence, and a (ii) deep syntactic structure with language-independent predicate-argument relations between content words. Both structures follow the dependency syntax formalism. While the surface structure is equivalent to the output of existing dependency parsers, the deep structure is more “semantic” and hence more suitable for concept extraction.

Service name	Dependency parsing service
SVN artefact	ms-svc-dep
Functional description	Annotates texts with syntactic parses of their sentences
Deployment status	Deployed
P1 version	Version 1.0, fully functional (for English language)
P2 version	Version 2.0, fully functional (for English and Spanish languages)
P2 improvements	Added support for Spanish language. NIF RDF annotations validated by Ontotext.
Maximum amount of queries per second	Although the service is pretty fast, no more than a few sentences can be processed per second.
Supported languages	English, Spanish.
Known issues	N/A
Next steps	Add remaining languages.

Table 5: Integration description of the Dependency Parsing service

3.3.3.6 Relation extraction

The relation extraction finds and annotates n-ary relations between entities, concepts and other relations in the text. Two kind of relations are annotated, coreference relations and semantic relations indicated by linguistic predicates. Coreference relations are established between anaphoric expressions and their antecedents to indicate they denote the same

referent (i.e. a NE, a concept). Semantic relations involve multiple participants in a situation such as an event. They are derived from the deep syntactic predicates and their arguments detected by the dependency parsing service. The relation extraction service assigns to predicates semantic types from a lexical repository of relational meanings (i.e. FrameNet) and assigns role labels to each argument participating in the frame. When a predicate can take multiple frames, a disambiguation strategy is required.

Service name	Relation extraction service
SVN artefact	ms-svc-rel
Functional description	Annotates texts with n-ary semantic relations and coreference chains.
Deployment status	Deployed
P1 version	Version 0.2, baseline version
P2 version	Version 0.5
P2 improvements	Replaced the third-party library, Semafor, with a much faster deterministic tool developed by UPF. NIF RDF annotations validated by Ontotext.
Maximum amount of queries per second	The current performance of the service is very fast, but future deployment of a disambiguation strategy may slow down the service.
Supported languages	English.
Known issues	The lack of disambiguation affects the quality of the output.
Next steps	Add support for additional languages. Implement a disambiguation strategy.

Table 6: Integration description of the Relation extraction service

3.3.3.7 Sentiment analysis

The sentiment analysis service aims at detecting sentiment in English text. The sentiment is annotated for the news article and for each sentence within the news article. The implemented service is currently using the baseline SentiStrength tool. This lexicon-based tool provides for a piece of text a sentiment score, which is in a range from -1.0 to 5.0. This service provides the following set of sentiment features:

- Sentimentality: $|score_{pos}| + |score_{neg}| - 2 = score_{sent}$
- Polarity: $score_{pos} + score_{neg} = score_{pol}$
- Minimum and maximum polarity range
 - For a better understanding of the sentiment values, the sentiment score is given in a range from 0.0 to 4.0.
- Negative polarity value: negative sentiment expressed in the news article.
- Positive polarity value: positive sentiment expressed in the news article.

Improvements were made to the module to receive an object in the RDF format and return the updated object with the sentiment information in RDF format.

The next steps involve the implementation of a domain-specific sentiment lexicon using the annotated in-domain news corpus to train the algorithms. In addition, in order to detect opinion targets (entities) we aim to analyse sentiment relations that different entities share.

Service name	Sentiment Analysis
SVN artefact	ms-svc-sa
Functional description	This service aims to provide an analysis of the sentiment that is expressed in a news article. The extracted sentiment is given by SentiStrength sentiment lexicon.
Deployment status	Deployed
P1 version	Version 0.2, baseline version
P2 version	Version 1.0, fully functional extracting the sentiment features for a given news article.
P2 improvements	Developments for additional features, such as negative and positive polarity. Validation of the results at sentence and full news article level. The received and returned object is RDF (before it was implemented for a JSON object).
Maximum amount of queries per second	N/A
Supported languages	English (additional languages can be supported given the availability of machine translation).
Known issues	Requires NIF wrapper class to access the RDF part of the article object (full text of the news article or sentences).
Next steps	We propose the creation of a sentiment lexicon for domain-specific sentiment words. In addition to the features extracted by the sentiment extraction module (sentiment lexicon), we aim at providing entities that are mentioned as esteemed or disdained and capturing sentiment relations between sentiment words and entities.

Table 7: Integration description of the Sentiment analysis service

3.3.3.8 Extractive summary

Extractive summarisation refers to the generation of summaries from texts by selecting sentences from the original summaries and composing a summary from these (unchanged) sentences. The language of the summary is the same as that of the original documents. The service has been extended to incorporate metrics based on semantic features such as NE, concept and relation annotations.

Service name	Extractive summary
SVN artefact	ms-svc-summ
Functional description	Generates extractive summaries from either a single document or a whole collection.

Deployment status	Deployed
P1 version	Version 0.2, baseline version
P2 version	Version 1.0
P2 improvements	Current version incorporates first methods resulting from research in T6.4. Relevance metrics for sentences are now based on semantic features in addition to keywords.
Maximum amount of queries per second	The performance of the service is good, meaning that up to a summary can be processed per second.
Supported languages	English
Known issues	No support for multilingual summarisation due to lack of corpora.
Next steps	Improve the metrics based on semantic annotations of relations. Metrics for local coherence to be added.

Table 8: Integration description of the Extractive summary service

3.3.3.9 Classification

This activity deals with the classification of News Items retrieved from the CNR into categories by means of a supervised learning technique called Random Forests (RF). The current baseline version of the category classification service makes use of textual features that are extracted from the text body of the News Items, while the planned fully functional version of the service will utilise the multimodal features that are produced in WP2 (textual concepts, named entities) and WP3 (sentiment, polarity and contextual information).

Service name	Category classification
SVN artefact	wp4/ms-svc-categoryClassification
Functional description	The category classification service receives as input the text body of a News Item that is retrieved from the CNR, extracts a set of textual features (N-grams) and utilises a Random Forest classifier, in order to provide as output the category, to which the News Item belongs.
Deployment status	Remotely deployed
P1 version	Version 0.2, baseline version
P2 version	Version 0.8, baseline version
Supported languages	English
Known issues	R (statistical computing software) needs to be installed in the computer where the service runs, along with the packages randomForest, tm and stringr.
Next steps	The fully functional version of the service is currently under development. It will be integrated into the MULTISENSOR platform as soon as there is enough content stored into the RDF repository, in order for the classification models to be trained properly.

Table 9: Integration description of the Category classification service

3.3.3.10 Context extraction

The context extraction service requires as input the textual content and the metadata that is stored in the html source of the media item. Alike to the previous version of the context extraction module (version 1.0), the module extracts either from the text or the metadata the following information: *author* (or creator of the content item); a set of *keywords* characterising the content item; the *literary style* or *formality level* of the content of the item; and the *genre* of the item (if found in the metadata). Other features such as *date*, *location*, and *source* are provided by the crawler service.

Service name	Context extraction
SVN artefact	ms-svc-context
Functional description	Given a media item, the context extraction service extracts or collects from the output of other services contextual features, such as <i>author</i> , <i>source</i> , <i>title</i> , <i>keywords</i> , <i>genre</i> , <i>category</i> , <i>date</i> , <i>location</i> , <i>literary style</i> and <i>language</i> .
Deployment status	Deployed
P1 version	Version 1.0, fully functional extracting a subset of the contextual features
P2 version	Version 1.2, fully functional extracting a subset of the contextual features
P2 improvements	The output of this service has been validated and improvements were made to the module. At the moment, the module receives and returns a RDF object.
Maximum amount of queries per second	N/A
Supported languages	English (additional languages can be supported given the availability of machine translation).
Known issues	Requires NIF wrapper class to access the RDF part of the article object.
Next steps	The next step concerns the integration of additional features. These features focus the news articles content, and we aim to provide one or more algorithms that predict for a given news article the relevance level of a specific feature (e.g. technicality).

Table 10: Integration description of the Context extraction service

3.3.3.11 Audio extraction and ASR

This service extracts audio from video files and recognises text from audio files. The resulting text files can then be processed by the textual dimension of the CEP.

Service name	Speech recognition
SVN artefact	http://voicepro.linguatec.org/rest/documents/2075/
Functional description	Extracts audio from video files and recognises test from audio files.

Deployment status	Remotely deployed
P1 version	Version 0.8, fully functional
P2 version	Version 0.9, fully functional
P2 improvements	Extension of language coverage to German
Maximum amount of queries per second	ASR works as an asynchronous process. 4 simultaneous uploads / 100 words per minute
Supported languages	English, German
Known issues	Continued testing
Next steps	Continued improvement in recognition rate and performance

Table 11: Integration description of the Speech recognition service

3.3.3.12 Concept and Event detection

This activity involves the detection of a set of predefined concepts in multimedia files (including videos and images), by considering visual features. To this end, various procedures are involved, such as video decoding (applicable for video files only), feature extraction and supervised classification. The video decoding procedure is responsible for extracting a predefined number of frames from a video file. The feature extraction step refers to the extraction of descriptors that describe visually images by capturing either global or local information out of the images. Finally, the classification step refers to the development of models used for classifying images or video frames to the set of predefined concepts / categories.

Service name	Concept and Event detection
SVN artefact	wp2/ms-svc-conceptEventDetection
Functional description	The service receives as input a multimedia file (i.e. image or video) and computes degrees of confidence for a predefined set of concepts. In case the file is a video, the video decoding step extracts specific frames from the file. In the feature extraction step, strictly local features are considered, namely the SIFT and SURF descriptors and their variations (RGB-SIFT, RootSIFT, opponent-SURF, etc.). Finally, all the concept detection models have been developed and trained by means of the Support Vector Machines (SVM) classification algorithm. The models provide confidence scores indicating the belief of each model that the corresponding concept appears in the image or video file.
Deployment status	Remotely deployed
P1 version	Version 1.0, fully functional version
P2 version	Version 1.1, fully functional version
P2 improvements	Concept detection models for approximately 30 new concepts have been developed and integrated into the service. Regarding the feature extraction procedure and for a specific set of concepts, a procedure called hierarchical saliency detection (aims at finding

	and isolating the most important information of the image) was applied to the training image datasets, in order to pre-process them before the extraction of the features and the training of the concept detection models. Finally, in the classification step, an improved late fusion strategy (compared to the corresponding strategy used in the P1 version) was utilised for fusing the prediction results of the concept detection models.
Maximum amount of queries per second	N/A
Supported languages	N/A
Known issues	Dependencies: OpenCV and vlfeat libraries, ffmpeg, ffprobe.
Next steps	Implementation of a version that apart from concepts, will additionally offer object and event detection functionalities.

Table 12: Integration description of the Concept and Event detection service

3.3.3.13 Indexing

In the Indexing service, a multimedia data representation framework that allows for the efficient storage and retrieval of SIMMO objects is developed. The service stores the News Items of CNR into MongoDB and allows the user to send a simple or more complicated query.

The current status of the service remains unchanged compared to the previous version. This is due to the adjustments that should be realised in the SIMMO model, in order to be able to hold efficiently all the required information (i.e. named entities, concepts, sentiments etc.) and thus perform more complicated questions to the Mongo database.

Service name	Indexation
SVN artefact	wp4/ms-svc-multimediaStructure
Functional description	The service receives as input a set of parameters and their values, which are used for performing queries to the MongoDB and returns the retrieved results.
Deployment status	Deployed
P1 version	Version 1.0, fully functional version
P2 version	Version 1.0, fully functional version
P2 improvements	N/A
Maximum amount of queries per second	Normally, the amount of queries that can be handled by MongoDB per second is in the order of 10000+.
Supported languages	N/A
Known issues	None
Next steps	Update of the SIMMO model, in order to hold all the necessary information found inside CNR and the data produced from the pipeline services.

	Update of the type of queries that can be handled by the service. Currently the service retrieves exclusively SIMMO objects based on a key value. It will be updated, in order to handle more complicated queries.
--	--

Table 13: Integration description of the Indexing service

3.3.3.14 RDF Validation

To improve the RDF data quality Ontotext integrated validation tool named RDFUnit. It is an open source test driven data-debugging framework that can run automatically or manually generated tests against SPARQL endpoint or file in one of the recommended by W3C data formats. In MULTISENSOR is used adapted version for NIF format. This helps all the partners to align their RDF output with the standard. There are two ways to use the RDF validator:

- By command line
- By user interface

History of 500 validated files in two different formats – turtle and HTML is kept on the server. This provides the users with easy way to check the results of the validated files.

If there are modifications on the data structure for the final prototype, we should consider do the validation tool going to cover these changes. The answer of this question will lead to some modifications on the test cases or on the tool.

Service name	Storing RDF
SVN artefact	http://multisensor.ontotext.com/cep
Functional description	Validate the RDF content produced by CEP and each service separately. Produce results in two different formats – turtle and HTML and store these results on the server. Provide the users with easy access to these results.
Deployment status	Remotely deployed
P1 version	Implemented in for P2
P2 version	Version 1.0, fully functional
Maximum amount of queries per second	20 files per minute
Supported languages	The validation service validate the schema of the data, so it does not depend on particular language. It supports all recommended RDF formats by W3C.
Known issues	At the moment there are no known issues.

Table 14: Integration description of the Storing RDF service

3.3.3.15 Storing RDF

The RDF Storing service is designed to handle input in the form of SIMMO JSON objects, parse this input and store it in the knowledge base where each SIMMO is stored in different context. The supported request methods are PUT and POST. For the First Prototype, this service was designed to extract particular fields from the SIMMO, extract also the json-Id part of the object and store them both in the repository. For the Second Prototype, the RDF

validation service has been integrated, so each SIMMO object first is validated and if the validation is successful, the data is stored in GraphDB. This service is fully functional and implemented. If there are changes in the SIMMO model for the last prototype, this will lead to changes in the storing service too.

Service name	Storing RDF
SVN artefact	http://multisensor.ontotext.com/cep
Functional description	RDF storing service is designed to handle SIMMO JSON objects, parse, validate and store then in the semantic repository.
Deployment status	Remotely deployed
P1 version	Version 1.0, fully functional
P2 version	Version 2.0, fully functional
P2 improvements	For the Second Prototype the storing service is integrated to work with the RDF validation service. This produce better data quality.
Supported languages	The RDF storing service does not depend on any particular language but on the data format.
Known issues	Some modifications should be made to parse and store the original text from the SIMMO in the semantic repository.
Next steps	As next steps we should try to improve the inserting speed.

Table 15: Integration description of the Storing RDF service

3.3.4 Content Alignment Pipeline (CAP)

The Content Alignment Pipeline includes the Content Alignment service, which receives as input a document ID and returns as output a list of IDs regarding documents with similar content to the input document. It uses the RDF knowledge repository to extract the documents' content and identify similarities between content. The CAP is initiated by the user when he/she selects a document from the user interface. The current service implements a baseline version, which makes use of a SPARQL query-based method for retrieving similar content. Content similarity is calculated based on the number of common named entities between articles. Currently, the considered named entities are Persons, Locations and Organizations. The score value of an article is computed as:

$$Score_{i,j} = \sum_k w_k * NE_{i,j}^k$$

where i is the article, whose ID is given as input and j is the article under consideration, w_k is the weight that is assigned for a Named Entity type with $k \in \langle Person, Location, Organization \rangle$ and $0 \leq w_k \leq 1$, $NE_{i,j}^k$ is the common Named Entity count between i and j .

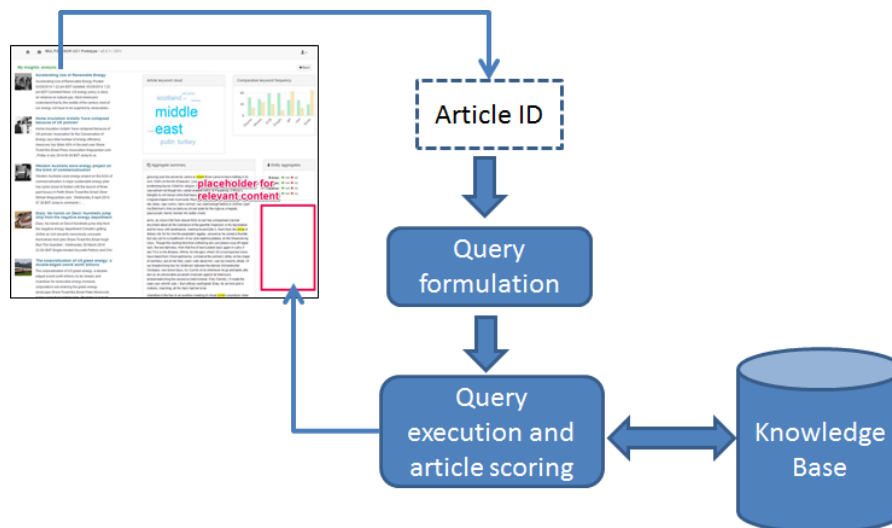


Figure 9: Content Alignment service

Service name	Content Alignment
SVN artefact	wp4/ms-svc-contentAlignment
Functional description	The service receives as input an article ID in String format and provides as output an ordered list of article IDs. The ordering is based on relevance to the input articles. The relevance score is calculated as a weighted sum of common references to named entities.
Deployment status	Deployed
P1 version	Version 0.2, baseline version
P2 version	Version 0.5, extended baseline version
P2 improvements	The current version works on the actual KB data. An improved relevance score calculation method has been implemented. The service is developed as a REST service and produces JSON output.
Maximum amount of queries per second	Depends on GraphDB capabilities. The average time for a query response is 918.1ms.
Supported languages	N/A
Known issues	None
Next steps	The advanced version of CAP for identifying contradiction and hidden relations between content will be implemented.

Table 16: Integration description of the Content alignment service

3.3.5 Social Media Analysis Pipeline (SMAP)

The Social Media Analysis pipeline (SMAP) is a set of processes related to analysis of social network data stored into the MULTISENSOR repositories. It is executed periodically in the background by the Supervisor. The SMAP pipeline performs social influence and interaction analysis on previously crawled Twitter data. The data is collected using the Twitter collector (see Section 3.3.1.3). The collector makes use of Twitter's streaming API, in order to produce

JSON-encoded data containing the set of posts relevant to a given set of hashtags, together with information about the profiles of the posters and the associations among them.

Given this data, the Graph Extraction service builds a topic-dependent network of contributors based on the mentions in the set of monitored tweets. It also computes retweet probabilities between users in this network, and finally the Influential User Detection module outputs two ranked lists of users, one by decreasing order of Pagerank and another one by decreasing order of influence in the Independent Cascade model (see deliverable D3.3 – Section 6.2.2). The Community Detection module, on the other hand, makes use of the Twitter posts collected by the Twitter collector, in order to detect online dynamic communities by means of an appropriate community detection algorithm, which is applied to each graph snapshot defined by the user network of mentions. The flowchart of SMAP is depicted in Figure 10.

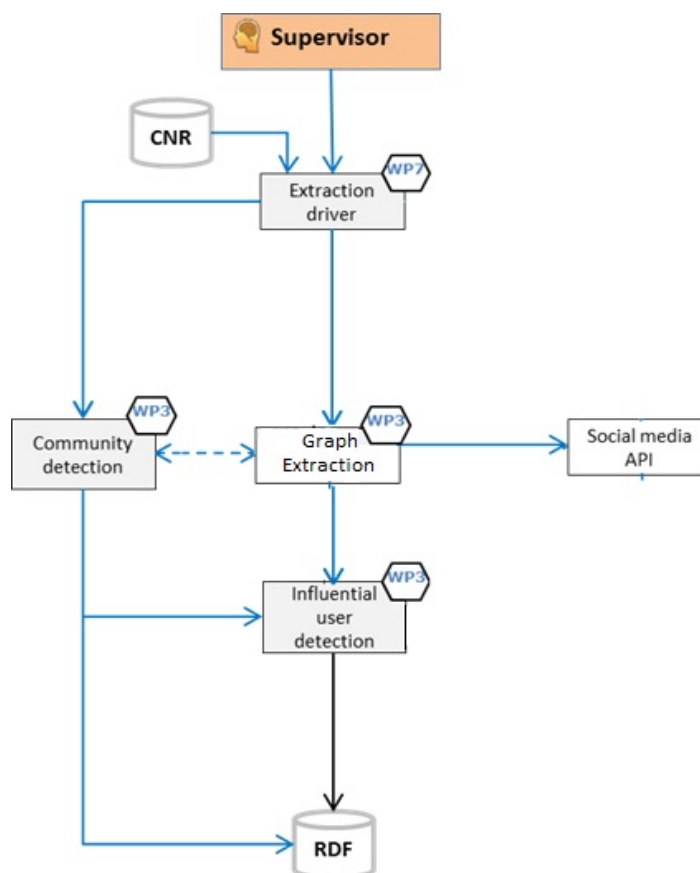


Figure 10: SMAP execution

Service name	Social Media Analysis Pipeline (SMAP)
SVN artefact	Influential User detection: wp3/ms-svc-socialMediaAnalysis Community detection: wp3/ms-svc-communityDetection
Functional description	Influential User detection: The Influential User detection service receives as input Twitter posts retrieved by the Twitter collector for a given set of hashtags concerning each Use Case separately and a list containing the 10 most influential users in descending order for each Use Case is provided as output. Community detection: The community detection service receives

	as input tweets retrieved by the Twitter collector for a given set of hashtags concerning all 3 Use Cases, utilises the Infomap method, in order to process the tweets posted in a given date that is added as an input parameter and provides as output user communities based on the associations that arise when a user mentions another user. The service displays these communities in descending order, according to the number of users that exist in each community.
Deployment status	Remotely deployed
P1 version	---
P2 version	Version 1.0, fully functional version
P2 improvements	N/A
Maximum amount of queries per second	N/A
Supported languages	Multilingual
Known issues	<p>Influential User detection: Python needs to be installed in the computer where the service runs, along with the libraries JSON and networkx.</p> <p>Community detection: R (statistical computing software) needs to be installed in the computer where the service runs, along with the packages igraph and rjson.</p> <p>Additionally, a MongoDB database, along with the Twitter collector, need to be running on the computer where both services run.</p>
Next steps	In the next steps regarding the SMAP, a faster approximate computation of reachability sizes via bottom-k sketches will be realised. Moreover, the modelling of information propagation will be improved. Finally, it should be noted that the concluding set of implemented services has been identified by the user partners as the most important. Therefore, no further modules are foreseen for the SMAP.

Table 17: Integration description of the Social Media Analysis Pipeline (SMAP)

3.3.6 Platform monitoring

The servers will be monitored with tools MRTG⁷ and Nagios⁸, which will allow controlling aspects regarding the performance and reliability of the technical infrastructure of the project. The results will be reported in the future deliverable.

⁷ <http://oss.oetiker.ch/mrtg/doc/mrtg.en.html>

⁸ <https://www.nagios.org/>

3.3.7 Platform testing services

For the Second Prototype, the separation between production and development environments has been performed. The solution for testing CEP services was developed and configured to run on development environment. This is online-based testing tool assessable via <http://grinder1.multisensorproject.eu/cepTesting>.

With the help of the CEP tool testing the partners can:

- Test services individually and independently on other CEP services
- Execute entire CEP pipeline
- Obtain output of the services on the same page or download output stored in JSON file
- Access/Download logs

In Figure 11 the initial view of the tool is displayed. On left side menu, services buttons are marked with green colour, by clicking on each button the respective service will be triggered, and the output of the service will be displayed on the right part of the CEP testing tool (see Figure 12). Output represents a series of containers RDF content produced by the service.

CONTENT EXTRACTION PIPELINE

Custom input

1. LANGDETECT

Text to be analyzed

2. NER

Simmo's URL

Place your text here

JSON file input

Choose File

No file chosen

1. CONCEPT EXTRACTION

2. DEPENDENCY PARSING

3. RELATION EXTRACTION

4. SENTIMENT ANALYSIS

5. CONTEXT EXTRACTION

6. EXT. SUMMARIZATION

7. CAT. CLASSIFICATION

RUN CEP completely

RUN CEP from ES article

6a0c68d675fbc078b2e8f8ca3d66

CEP LOGS

CEP OUTPUTS OF ES ARTICLES

Output execution of single service

Clear output

Output console

Clear console

Figure 11: CEP testing tool initial view

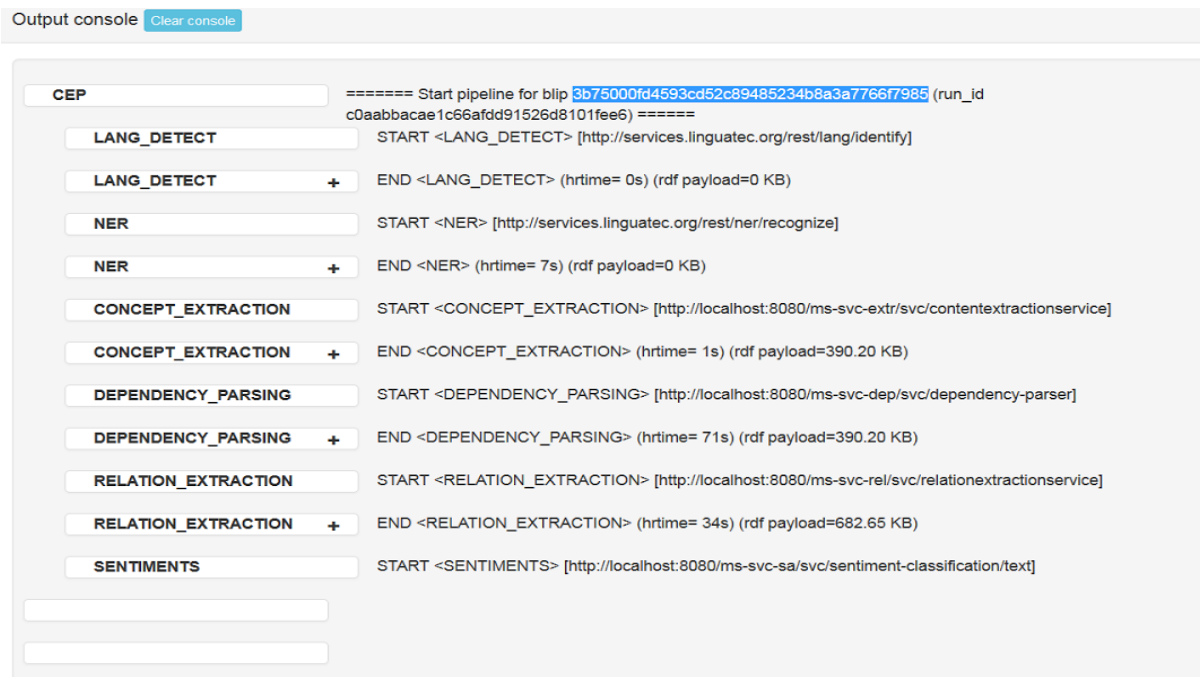


Figure 12: CEP testing tool output

3.4 Online modality

3.4.1 Business Shared Services

3.4.1.1 Content delivery

The Content delivery service gives access to the information from KB. In other words, it provides an access to information that available internally via REST API.

For the First Prototype, the service was not useful due to not populated KB. Many different entities were not available.

For the Second Prototype, the content delivery service was extended. Here is the full list of online services with description:

Description	Endpoint	Output	Status
	<a href="https://grinder1.multisenso
rproject.eu/onlineapi/news">https://grinder1.multisenso rproject.eu/onlineapi/news		
Retrieves the original SIMMO article, including title, language, etc	/ID/raw	JSON object. Additional fields, all the content placed within _source field.	DEPLOYED
Retrieves all the multimedia assets of an item	/ID/media	Array of URLs	DEPLOYED
Retrieves summary text created by PR services and stored as intel	/ID/summary	Plain text	DEPLOYED

Retrieves content of entities field (persons, locations)	/ID/entities	JSON object	Temporarily discarded Please use /all
Retrieves content of sentiments fields	/ID/sentiments	JSON object	Temporarily discarded Please use /all
Retrieves the original text of the article and the additional translation performed	/ID/trans/LANG	Plain text	DEPLOYED
Retrieves all the intel available on GraphDB	/ID/all	JSON object	DEPLOYED

Table 18: List of the online services

Service name	Content delivery
Functional description	Service provides API for online services
Deployment status	Deployed
P1 version	Version 0.8
Maximum amount of queries per second	Up to 20
Known issues	NA
Next steps	Following functionalities will be implemented: Content retrieval for entities and sentiment

Table 19: Integration description of the Content delivery service

3.4.1.2 Semantic search

The Semantic search is implemented in two different ways:

- CNR (ElasticSearch instance)
- RDF (KB implemented as a GraphDB instance)

The semantic search is currently a hybrid service. It calls the original content from CNR and the semantic data from the RDF repository. This approach has been chosen as optimal system performance. The calls to CNR are several times faster than the calls to KB.

In the future, it will be switched to GB for all routine to have centralised repository, even if this could affect productivity of the system.

Following changes were made in comparison to the First Prototype:

- Removed Search-type custom header
- Removed facetNames

- Fixed minor issues when searching for different fields at the same time
- Mandatory use of Content-type header which was not clear in 1st version
- Search operator (AND & OR) can be set when searching for multiple words (e.g., querying for energy AND agency will provide different output than energy OR agency)

The endpoint for the service is the following:

<http://grinder1.multisensorproject.eu/onlineapi/search/rdf>

The method for querying against it is always POST, so the different fields will be specified on the request's body.

Fields that currently available for search:

- subject
- title
- source
- language
- country

Request parameters (**mandatory** parameters in bold)

- **queryFields**: fields to be used.
- **queryWords**: words to search on the different queryFields.
- offset: from where to start querying.
- limit: amount of articles to retrieve at the same time.
- use_case: filter by use_case field from PR API.
- pr_feed: filter by pr_feed field from PR API.

Important: queryFields and queryWords must have the same length, since they are mapped together. If you use more than one field, specify the different fields using a CSV format (queryFields=title,country & queryWords=energy,en).

For faceted values retrieval:

- **facets**: set to **true**, and
- **facetFields**: fields to be used as facets [example: facetFields=title,subject,country]

More fields will be added soon, so almost all the fields from the SIMMOs will be searchable.

Service name	Semantic search
Functional description	Service provides API for semantic search
Deployment status	Deployed
P1 version	N/A
P2 version	Version 0.8
Maximum amount of queries per second	Up to 20
Next steps	Next delivery service will be switched to GB

Table 20: Integration description of the Semantic search service

3.4.1.3 Topic-Event detection

Topic-event detection is tackled as a clustering problem on the space of concepts and named entities. The goal of this activity is to provide a grouping for a list of News Items without a priori knowledge of the number of topics. The current version of the topic-event detection service makes use of the named entities and concepts, extracted offline as described in Sections 3.3.3.3 (Named Entities recognition) and 3.3.3.4 (Concept extraction). Each detected topic is presented as a list of article IDs, ordered from the most topic-relevant to the less topic-relevant. The irrelevant news items, if any, are presented as a topic, namely “noise”, after the last detected topic.

Service name	Topic detection
SVN artefact	wp4/ms-svc-topicDetection
Functional description	The service receives as input the concepts and named entities of a list of News Items retrieved from the CNR. The topic detection service estimates the number of topics using one realisation of the DBSCAN-Martingale (i.e. a density-based clustering algorithm), extracts irrelevant News Items as noise and re-orders the News Items in each topic-event, using Latent Dirichlet Allocation.
Deployment status	Remotely deployed
P1 version	Version 0.2, baseline version
P2 version	Version 1.0, fully functional version
P2 improvements	Concepts and Named Entities are employed, so the monolingual previous baseline version became multilingual.
Query list size	Minimum: 100 News Items (at least 2 topics have to be detected) Maximum: 2000 News Items (for real-time results)
Supported languages	Multilingual
Known issues	R (statistical computing software) needs to be installed in the computer where the service runs, along with the packages tm, RWeka, fpc, topicmodels and rjson.
Next steps	The next version of the service will be faster and scalable to very large databases. Additionally, the detected topics-events will be labelled.

Table 21: Integration description of the Topic detection service

3.4.1.4 Similarity search

The Similarity search service involves the retrieval of similar articles/documents given a query. Depending on the query, which can be an image, video or finally article that may include images or videos, similarity based on a single or multiple modalities is realised. The service involves the creation and update of indexing structures for every modality (i.e. visual features, visual concepts, textual concepts, and named entities). These structures use different monomedia similarity measures. It should be noted that the indexing structures are updated regularly, that is every time a new item is stored into MongoDB holding the SIMMO

objects. Finally, the similarities between the query item and the indexed objects are calculated and the top k results are returned.

The current status of the service remains unchanged compared to the previous version. This is due to the adjustments related to the SIMMO implementation in order to be able to hold efficiently all the required information (i.e. named entities, concepts, sentiments etc.).

Service name	Similarity search
SVN artefact	wp4/ms-svc-similaritySearch
Functional description	The service receives as input the ID (Elastic Search ID) of an article and outputs a list of 250 randomly selected article IDs.
Deployment status	Deployed
P1 version	Version 0.1, dummy version
P2 version	Version 0.1, dummy version
Known issues	None
Next steps	The fully functional version of the service that will output a list of the most similar articles (IDs) to the input article (ID) is currently under development. To this end, a suitable ranking / retrieval multimodal similarity algorithm is being developed and will be integrated into the service.

Table 22: Integration description of the Similarity search service

3.4.1.5 Machine Translation

The Translation service receives as input text in source language and translates it into specified target language.

Service name	Machine Translation
SVN artefact	http://services.linguec.org/rest/lang/translate
Functional Description	Receives as input text in source language and translates it into specified target language
Deployment status	Remotely deployed
P1 version	Version 0.8, fully functional
P2 version	Version 1.0, fully functional
P2 improvements	Improved translation quality resulting from: a) better homogenisation of training corpus b) reduction of the amounts of unknown words c) tuning of model parameters
Maximum amount of queries per second	7 queries (sentences) per second
Supported languages	English to/from French; English to/from German; English to/from Spanish; English to/from Bulgarian.

Known issues	Continued testing
Next steps	Continuation of quality and performance tuning

Table 23: Integration description of the Machine translation service

3.4.1.6 Abstractive summary

Abstractive summarisation refers to the generation of multilingual summaries from data using natural language generation methods. The abstractive summarisation service can generate text-based summaries about one or more entities in the semantic repository following a user query requesting information about those entities. The summarisation process consists of a text planning stage, where the most relevant facts about the entities are selected from the semantic repository and sorted in the order in which they will appear in the text, and second surface generation stage where the contents are rendered in natural language.

Due to delays in the population of the semantic repository, the text planning strategy for the current prototype is template-based and implemented as a set of SPARQL queries. The surface generation, on the other hand, is rule-based as foreseen in the DoW.

Service name	Abstractive Summarisation service
SVN artefact	ms-svc-summ
Functional description	Produces abstractive summaries in response to keyword-based queries.
Deployment status	Deployed
P1 version	Version 0.1, dummy
P2 version	Version 0.5, partly functional
P2 improvements	Template-based text planning and rule-based surface generation modules have been deployed for the production of summaries in English.
Maximum amount of queries per second	The module can produce a short summary in no less than a few seconds. Processing time depends on the length of the summary.
Supported languages	English
Known issues	Text planning (selection and ordering of contents) follows a very simple and rigid approach, affecting the relevance and coherence of the resulting summaries.
Next steps	Obtain a model of relevance and local coherence of contents and implement a learning mechanism. Deploy statistical generation. Support for additional languages.

Table 24: Integration description of the Abstractive summarisation service

3.4.1.7 Contributor analysis

The contributor analysis module receives as input a twitter handle (e.g., "@barackobama"), and then queries the Twitter API for information about the user and his immediate

connections, including measures of the user's authority. The authority scores are based on three criteria: reach (number of followers and size of the ego network), relevance to a given set of keywords and retweet influence score (average fraction of followers that retweet a random post by the user).

Specifically, the service allows a user with a legitimate Twitter application and user authentication keys to crawl the profile of particular users and compute basic statistics on network and retweeting influence. Instead of giving as input a specific twitter handle, the service can work alternatively given a specific search key as input, e.g., "Barack Obama". Given this search key, the service retrieves the top 10 relevant Twitter accounts with this string and proceeds as before with each of them. The service has been fully developed and deployed. Since this an online service operating given a specific user input (i.e., a specific Twitter handle or a search key for retrieving relevant twitter handles), the service is integrated but with specific limitations. Depending of the Twitter API, the users of this service are allowed to check 3 accounts per hour.

The Contributor Analysis module was fully functional in the First Prototype. Hence, no further changes were required.

Service name	Contributor analysis
SVN artefact	ms-svc-contributorAnalysis
Functional description	Retrieve information about a Twitter user and compute local authority scores
Deployment status	Deployed
P1 version	Version 1.0, fully functional
Known issues	Limitations of the Twitter API (3 accounts per hour)

Table 25: Integration description of the Contributor analysis service

3.4.2 Other Online Services

3.4.2.1 User profile

The User Profile service was designed in order to control user accounts over the different UCx. Service supported by all use cases and allows registration of the new record, authorise users and profile edition.

Profile information data is stored in the OPS repository.

In comparison to the First Prototype current version was significantly extended, currently API consists of the following functionalities:

PATH	METH	DESCRIPTION	PARAMETERS	RESPONSE
/register	- GET - POST	Renders registration view with a form (once integrated in the portals) to post credentials (username, email and password) to be saved in DB (MongoDB). GET for retrieving the register page and POST for registering a new user into the system. Creates a new user in OPS DB (Mongo).	- email - username - password	Response redirects to login page.

/login	- GET - POST	Insert user's details on database. MongoDB's methods to save and insert new users and their fields. For UC2 decide redirect view (home for example). GET for retrieving the login page and POST for logging into the corresponding portal.	- email - password	JSON Object (user) + Redirect to profile page on UC portals
/home	- GET	Renders home page. For the UC portals login and registration links will be displayed.	---	Redirect to the login page could be the response or simply show the login form in the same view.
/profile	-GET -POST -PUT -DELETE	Renders users dashboard. Allows to save and manage multiple search profiles associated to that user. GET for retrieving the profile info. POST for saving search profile data. PUT for updating the user's information. DELETE for removing specific content.	-user_language -email -profile_name -keywords -country -language -media_source -relevant -irrelevant	JSON Object
/profile/profile_name_to_be_changed	-PUT	Possibility to modify a particular profile_name stored for a set of filters. PUT for updating that profile's name. All the other filters stored remain unchanged.	-profile_name	JSON Object
/logout	-GET	Sign out path to end the session and redirect to initial home page. Logs out the existing user from the system.	---	Redirect to home page.

Table 26: Methods of the User Profile service

The Profile service model is implemented with the npm module a part of node.js.

Service name	User Profile
SVN artefact	ms-svc-profile
Functional description	Service provides API for profiling services
Deployment status	Deployed
P1 version	Version 1.0, fully functional
P2 version	Version 1.1, fully functional
P2 improvements	Improvements to store the relevant articles in the user folder
Maximum amount of queries per second	Up to 20

Table 27: Integration description of the User Profile service

3.4.2.2 Reference Data

The Reference Data is a service that permits to collect many indicators about the countries. Those indicators were selected by PIMEC to help the SMEs to understand what are the

relevant internationalisation factors and conditions. Those indicators are organised by categories and the following Table depicts all the indicators:

Category	Sub-category	Indicators
Economic indicators	GDP	* GDP growth
		Real GDP growth rate – volume (tec00115)
		GDP per capita in PPS (tec00114)
		GDP per capita – quarterly Data (namq_aux_gph)
		Exports of goods and services in % of GDP (tet00003)
		Imports of goods and services in % of GDP (tet00004)
		Export to import ratio (tet00011)
		Inward FDI stocks in % of GDP (tec00105)
	Importation / exportation	Customs and tariffs
		Structure of taxes by economic function (gov_a_tax_str)
		Export and Import
		Current account – quarterly data (ei_bpca_q)
		Harmonised indices – monthly data (ei_cphi_m)
		Foreign Direct Investment
Political indicators	---	Government type
		Political instability index
		Corruption perception index
		General government deficit (-) and surplus (+) – quarterly data (ei_nagd_q_r2)
Social indicators	Population	Life table (demo_mlifetable)
		Human Development Index
		Population with tertiary education attainment by sex and age (edat_lfse_07)
	Work	Unemployment rate
		Harmonised unemployment rates (%) – monthly data (ei_lmhr_m)
	Health	Life expectancy
		Life expectancy by age and sex (demo_mlexpec)
		Population distribution
Cultural indicators	Urbanisation	Distribution of population by degree of 45 urbanization, dwelling type and income group (source: SILC) (ilc_lvho01)
	Consumption habits	Economic sentiment indicator (teibs010)
		Households having access to the internet at home (isoc_pibi_hiac)
		Easiness of doing business

Table 28: List of the indicators

Most of the indicators are provided by EuroStat⁹ and WorldBank¹⁰. Those organisms publish the data as Linked Open Data in RDF format. So the EuroStat dataset¹¹ is stored in the knowledge base (GraphDB) and the indicators values are collecting by SPARQL queries.

For each indicator, SPARQL queries have been created to collect the values. To be completely adaptive to the user browsing, those queries are formalised as template to take

⁹ <http://ec.europa.eu/eurostat/web/main/home>

¹⁰ <http://databank.worldbank.org/data/home.aspx>

¹¹ <http://datahub.io/es/dataset/eurostat-rdf>

into account specific parameters (country, time frame, etc.). In the next Table, an example of a SPARQL query is presented:

SPARQL Query Template to get the Economic indicator called Inward FDI stocks in % of GDP

```
# Inward FDI stocks in % of GDP (tec00105)
PREFIX qb: <http://purl.org/linked-data/cube#>
PREFIX eudata: <http://eurostat.linked-statistics.org/data/>
PREFIX prop: <http://eurostat.linked-statistics.org/property#>
PREFIX eugeo: <http://eurostat.linked-statistics.org/dic/geo#>
PREFIX sdmx-dimension: <http://purl.org/linked-data/sdmx/2009/dimension#>
PREFIX sdmx-measure: <http://purl.org/linked-data/sdmx/2009/measure#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?datePretty ?value {
  ?s qb:dataSet eudata:tec00105;
    prop:geo ?country;
    sdmx-dimension:timePeriod ?date;
    sdmx-measure:obsValue ?value;
  FILTER(?country = eugeo:ES)
  BIND(((substr(str(?date),1,4+1+2))) as ?datePretty) # Monthly returns YYYY-MM
}
```

Table 29: SPARQL Query Template to get the Inward FDI stocks in % of GDP

Service name	Reference data
SVN artefact	ms-svc-refdata
Functional description	This service permits to collect indicators on Linked Open Data datasets through SPARQL queries.
Deployment status	Deployed
P1 version	Version 0.5, baseline version
P2 version	Version 1.0
P2 improvements	More indicators have been identified and integrated in UC3.
Next steps	More indicators will be integrated.

Table 30: Integration description of the Reference data service

3.4.2.3 Decision support

The decision support system is part of Task 5.3. To handle this task we decided to use two different approaches. The first one is based on loading and using statistical indicators retrieved from World Bank and Eurostat. These indicators are from different areas like – social, economic, political, sector and products. They will be used to compare all this parameters between the different countries. Based on that information, users will be able to make better decision and build their companies strategy.

As addition to the indicators, we are going to implement and integrate semantic recommendation engine. This engine will provide the users with highly relevant recommended articles. This will help journalists and press clipping agents to do their research, drill down on specific topic and find relevant content. In our existing solution, to come up with a single recommendation list we are combining a variety of factors: the relevance of the article to the currently opened one, the relevance with respect to the

previous reads of the user, popularity of the article among readers, co-visitation and freshness.

Service name	Decision support
SVN artefact	ms-svc-decsupport
Functional description	The semantic recommender will provide the users with highly relevant recommended articles. This will help journalists and press clipping agents to do their research, drill down on specific topic and find relevant content.
Deployment status	Deployed
P1 version	---
P2 version	Version 0.5, baseline version
Supported languages	For the first version of the recommender, we will support only English language.
Next steps	Allowing the user to adjust the influence of different recommendation factors is also an effective way to tackle the filtering bubble phenomenon - the user can control (and hence strongly decrease) the influence of the popularity/collaborative components thus reducing the influence of very popular stories.

Table 31: Integration description of the Decision support service

4 PROTOTYPE APPLICATIONS

4.1 UC1: Journalism Use Case

The UC1 application (see D7.1, section 4.1) is an application that should support media professionals (e.g. journalist, media expert) to find relevant information in different formats, coming from different sources, and according the social activities that were produced around.

This description is an update of the UC3 description provided in the D7.4 deliverable (see D7.4, section 4.1). The main improvement of the UC1 is the switch of the search engine (CNR) to the RDF repository. This is provided by the ElasticSearch connector which enables search functionalities on top of the knowledge base (GraphDB) provided by OntoText.



This means that the index used for the retrieval algorithm is now taking into account the intel extracted during the CEP process. Moreover improving the retrieval relevancy, the search functionalities are extended. In addition to the textual search that was available in the First Prototype, the semantic search and the multimedia search are provided in the Second Prototype.


This important improvement has an impact on the UI. The design did not change so much but now, all the displayed information is real data collected from the MultiSensor repositories through the Online Services.

As implemented in the First Prototype, the access to the application is managed by a user profile service, which controls the user account (credential and preferences). The user has to login with his/her credentials to the UC1 application. When the user is logged in the system, he can access his folder that contains his/her favourite documents.

Thanks to the new index repository (GraphDB), the search functionality has been improved. By selecting some keywords, the user can still realise a textual search but now he can realise an advanced search, which means a multimedia query or a semantic query:

- The multimedia search consists to retrieve all the articles that contain at least one multimedia element (image, audio or video).
- The semantic search: by selecting a semantic entity (i.e. Person, Organisation, Location), the retrieval engine will search specifically for the articles that contain semantic entities.



MULTISENSOR UC1 Prototype / v2.0 / GRINDER

Welcome foo@foo.com
 Sign out

Search

energy policy

Filter by entity: Clear

☐ Person
☐ Organisation
☐ Location


Multimedia Analysis:

☐ Show articles with analysed content

Filter by language:

☒ English
☐ French
☐ German
☐ Spanish

Filter by date: Clear



Submit










Title	Country	Source	Date
No change to dividend policy expected-Electrolux CEO Home appliances maker Electrolux is to stick to its current dividend policy, the company's top executive said on Tuesday after announcing a new round of cost saving measures. "Dividend policy is of course a board decision, but we have no changes in the dividend policy and I don't have any indicat 	 US	Reuters (USA)	15/11/2011
ECB Nowotny says new measures needed to boost inflation ECB Nowotny says new measures needed to boost inflation WARSAW European Central Bank (ECB) Governing Council member Ewald Nowotny addresses a news conference in Vienna June 6, 2014. Reuters/Heinz-Peter Bader WARSAW New efforts are needed to boost price growth in the euro zone as inflation, 	 GB	Reuters (GB)	15/10/2015
Baytex Energy Corp. 30/07/2015 11:30 Baytex Reports Q2 2015 Results... Baytex Energy Corp. 30/07/2015 11:30 Baytex Reports Q2 2015 Results... CALGARY, ALBERTA -- (Marketwired) -- 07/30/15 -- Baytex Energy Corp. ("Baytex") (TSX: BTE)(NYSE: BTE) reports its operating and financial results for the three and six months ended June 30, 2015 (all amounts are in Canadian do 	 FR	Le Figaro	30/07/2015
IMF urges Japan to proceed with second sales tax hike IMF urges Japan to proceed with second sales tax hike LIMA, Peru By Leika Kihara The International Monetary Fund (IMF) logo is seen at the IMF headquarters building during the 2013 Spring Meeting of the International Monetary Fund and World Bank in Washington, April 18, 2013. Reuters/Yuri Gri 	 GB	Reuters (GB)	12/10/2015
Younicos 03/08/2015 14:28 Younicos Applauds U.S. Senate Energy Committee Passage of Energy Policy M	 FR	Le Figaro	03/08/2015

Figure 13: Widget of the advanced search

In this version, the extracted knowledge of the articles is used by the retrieval engine to improve the ranking of the search results. When the user executes a query (textual, multimedia or semantic), the system retrieves a list of relevant documents (Figure 13). Then, the user can browse the results' list and select one article to view its intel information. Figure 14 presents the display of the knowledge extracted during the CEP process. The list of the knowledge related to the article is the following:

- **Context information:** this information is provided by the Context Extraction service. It provides the author's name, the date of publication, the source from which is coming from the article and the language of the article.
- **Overall sentimentality:** this information is provided by the Sentiment Analysis service. It provides the level of sentimentally related to the whole content of the article: numeric values between 0 and 4 are displayed in order to express the intensity of the sentimentality.
- **Category:** the list of the categories is provided by the Classification service.
- **Semantic information:** the semantic information is a link to display the whole article. In this page, the text is enriched with the named entities and the identified concepts.
- **Summarisation:** the summary is provided by the Extractive Summarisation service. By default, the text compression is defined as a 30% ratio of the original size.
- **Translation:** this information is provided by the Machine Translation service. The user can select one of the five available languages (English, French, Spanish, German and Bulgarian) to trigger the translation the summary in this language.
- **Tag Cloud:** the information is provided by specific library (Gramophone¹²) integrated in the platform. This library permits to analysis the text of the articles and identify the

¹² <https://www.npmjs.com/package/gramophone>

terms that have the highest frequency. Then, the most frequent terms are displayed in the tag cloud.

- **Named Entities:** the information is provided by the Named Entity Recognition service. The different entities recognised in the article's content are listed according to their type (i.e. Person, Organisation, Location and Time).

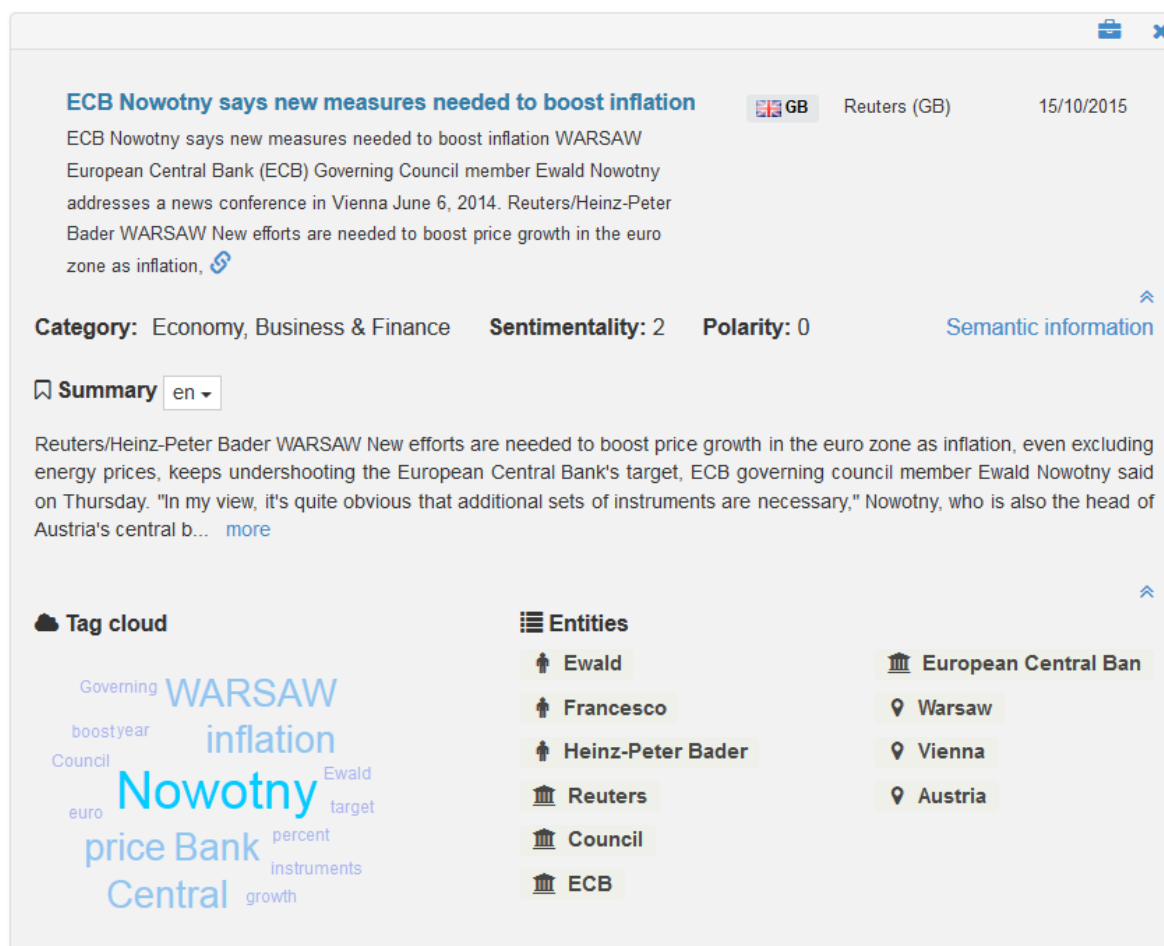


Figure 14: Panel with the Intel information of the selected article

As explained before, a link called “Semantic information” permits to visualise all the content of the article. As depicted in Figure 15, on the left of the page, different blocks of information are present:

- **Overall Sentimentality:** the degree of sentimentality of the whole text is depicted by numeric values between 0 and 4.
- **Named entities:** the list of the recognised Named Entities displayed. If one is selected in this block, all the instances of this entity are highlighted in colour in the text.
- **Related articles:** the list of the related articles is presented in this block.

The knowledge extracted by the CEP process is associated to the content in a structured representation (RDF format). Effectively, the content is annotated with the semantic metadata, which refer to the semantic dimensions of the articles (context, named entities, concepts, sentimentality, etc.). This knowledge representation permits to highlight directly in the text the knowledge that was identified (e.g. named entities and concepts). In the next version of the prototype, more knowledge regarding the article will be displayed in this page and more interactive functionalities to browse this knowledge will be provided to the users.



Figure 15: Display of the Semantic information of the article's intel (SIMMO)

And for an article which contains a multimedia element (image, audio or video), the concepts extracted from the video content by the Concept and Event detection are represented in a tag cloud (Cf. Figure 16).



Figure 16: Display of the multimedia information of the article's intel (SIMMO)

During the browsing of the search results, the journalist can see the synthetic view of the article and if he detects some relevant ones, he can add them to his folder. Then, after selecting several relevant articles, the journalist can go to his/her folder clicking on the "My Findings" button and trigger the analysis of this set of documents.

Then, the systems execute a multi-documents analysis. At this stage, three different multi-documents functionalities have been implemented in UC1 (Cf. Figure 17: Graphical view to represent the information of a set of documents):

- The tag cloud: the system analyses the term frequency over all the texts of the selected articles in the folder. Then, the tag cloud provides a graphical summary of the folder content.
- Entity aggregates: the analysis services retrieve all the entities that are present in the folder's documents.
- List of the related articles: based on the folder's articles, the system calculates the list of the related articles.

In the next version of the prototype, the abstractive summary will provide a conceptual summarisation of the documents set.

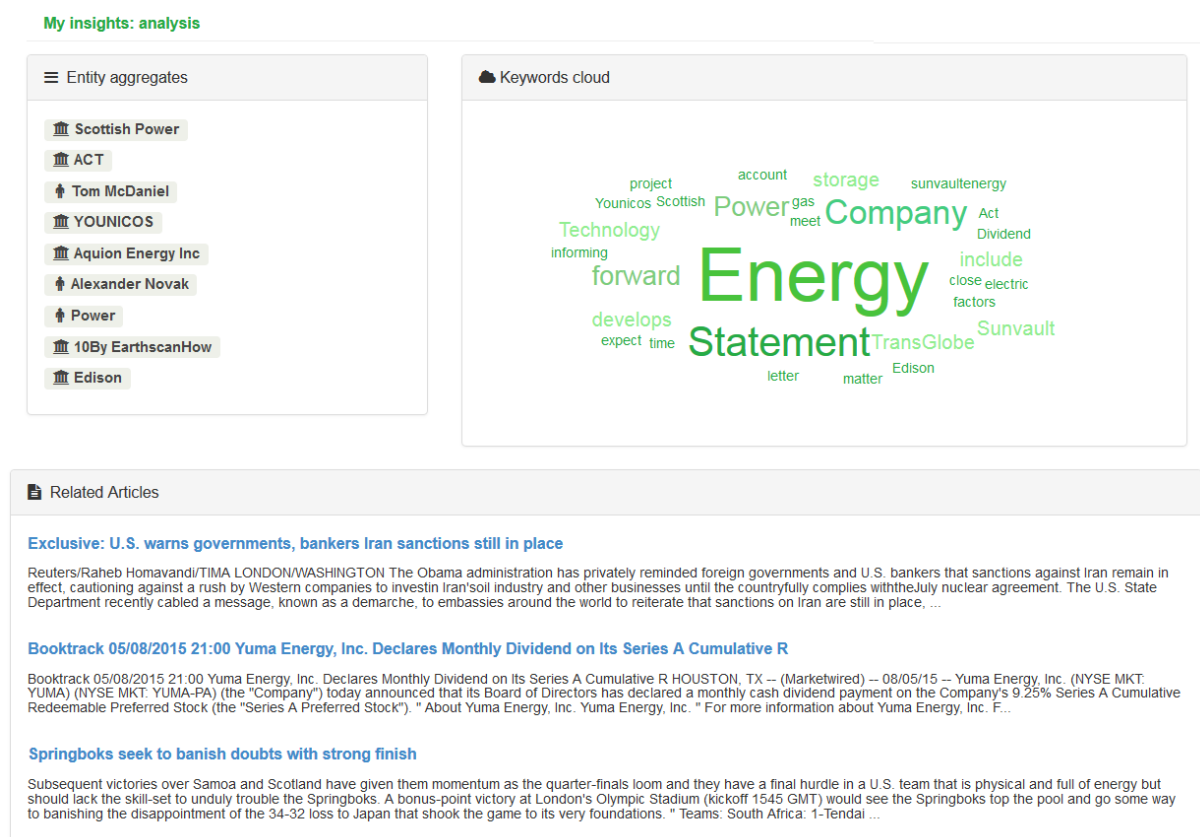


Figure 17: Graphical view to represent the information of a set of documents

All the Online services are integrated with the knowledge base. So their integration in the UC1 application is a really important step further, especially for the search functionalities. The Online Services are now able to exploit the knowledge available in GraphDB and most of those services can provide a specific display of this knowledge.

Here is the summary of the integration of the Online Services within the UC1 application and how the extracted knowledge is presented to the end user:

- **Profile service**: this service is fully integrated. It supports the login functionality and the storage of the relevant articles in the user's folder.
- **Semantic search service**: the textual, multimedia and semantic search functionalities are integrated with the knowledge base (with the ElasticSearch connector on top of

GraphDB). The different facets (categories, sources, countries, and most relevant entities) are fully implemented. Thanks to the new index, the results ranking is more precise. In the next prototype, the hybrid search will be implemented.

- Content delivery service: this service is fully integrated to access the whole content of the SIMMO or only some specific fields.
- Similarity search service: this service is fully integrated. It provides a list of related articles in the semantic view and the analysis view of multi-documents.
- Translations service: this service is fully integrated. It is available to translate the summary in 5 different languages.
- Summarisation service: Extractive summarisation is fully integrated.
- Abstractive Summary service: provided as a baseline version. It is implemented as a tag cloud to display the main keywords of each article.
- Clustering + Filtering service: Filtering service is implemented partially. This service is not required in UC1 application.

4.2 UC2: Media Monitoring Use Case

As already described in D7.2 and D7.4, the use case 2 application (Media Monitoring) will replicate the workflow of a media monitoring professional to execute an analysis for a client.

This includes checking articles for relevance by various indicators and saving the relevant articles for a client's profile. The relevant articles will then be analysed so that conclusions can be drawn from this analysis.

For the First Prototype, user-specific functionalities were implemented, in order to be able to register and login into the application. This has been described at length in D7.4.

When the user is logged in, he can either search for any keyword with additional filters or select a profile if it has been created.

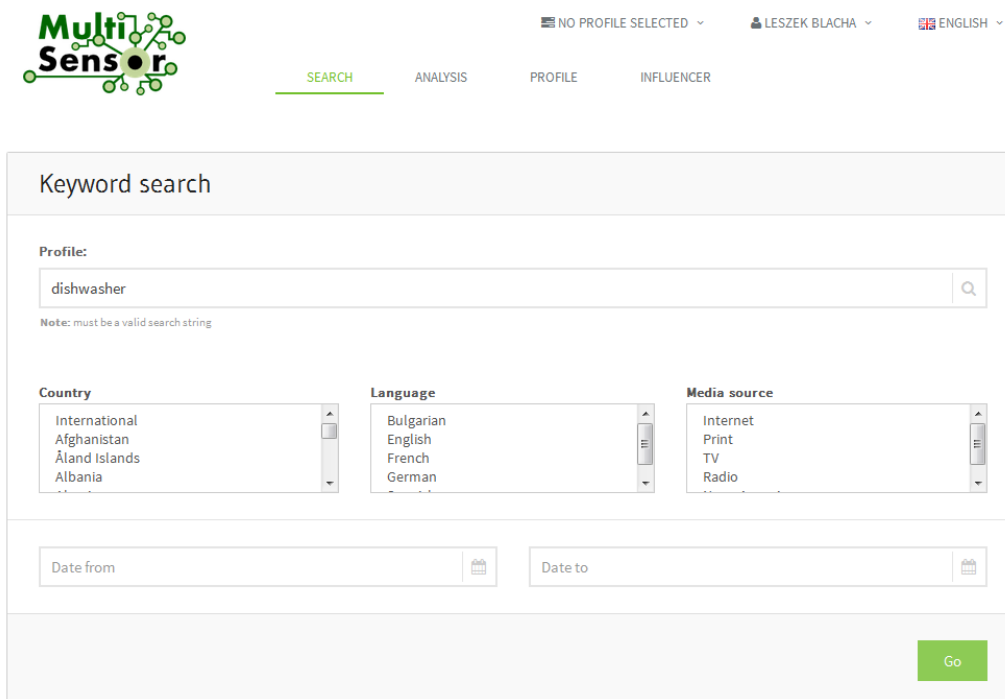


Figure 18: UC2 search page

After querying a search, the results are displayed grouped by topic to provide an easy way to mark articles as relevant/irrelevant.

Please choose

topics

ON

single results

OFF

Economy, Business & Finance

▶ 83	Accept	✓	Reject	✗
▼ Economy, Business & Finance 17	Accept	✓	Reject	✗
▶ Stockmarket 4	Accept	✓	Reject	✗
▶ Finance 4	Accept	✓	Reject	✗
▶ Currency 6	Accept	✓	Reject	✗
▶ Business 2	Accept	✓	Reject	✗
▶ Economy 1	Accept	✓	Reject	✗

Figure 19: Selection of the relevant articles

Moreover, all articles can be viewed in single article view with additional information like media source, date, language and country.

Within the single article view, the Translation service and the Summarisation service can be called to retrieve the translation and/or a summary of the article.

« Previous
Next »

accept all

✓

reject all

✗

reset

↺



What's new in floor care products

Accept

✓


Reject

✗


Source: IER Retail | Date: | Language:  | Country: 

What's New in Floorcare Products Dyson Digital Slim Revolutionises Irish Floorcare Market The Dyson Digital Slim, a lightweight Cordless vacuum cleaner engineered for nimble, highperformance cleaning, has revolutionised the floorcare market in Ireland. Consumer demand for this product has been "phenomenal", according to Dyson, and CFI(data for 2012 shows that the category, traditionally known as ...

Summary



get more Data





Do You Have an Innovation Mindset?

Accept

✓


Reject

✗

Source: socialmediatoday.com | Date: | Language:  | Country: 

Do You Have an Innovation Mindset? President, The Myndset Company Posted on September 2nd 2014 Innovation is a word that is used very frequently in business meetings and written with great conviction in annual reports. Innovation is a lofty ambition that can inspire. And it is considered by many as the single biggest driver of value. As Thierry Wellhoff wrote in his book, " ", innovation is a ...

Summary



get more Data




Figure 20: Summarisation and translation of the article's content

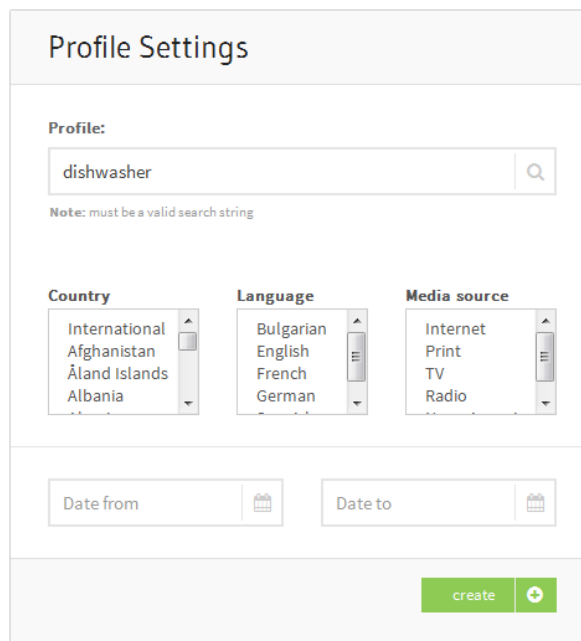
In the analysis section, previously included basic charts for media, country and language have been discarded. They are replaced by charts on facets provided by the Semantic Search service. This will be explained in more detail in the next section “Improvements since First Prototype”.

The selection of the most current articles has also been omitted.

Improvements since First Prototype:

In the Profile service, the user language can now be saved. Additionally the IDs of articles that are marked as relevant/irrelevant can be stored which is important for the analysis section of the UC2 application.

Editing and saving of profiles cannot be done from within the search section as in the previous prototype. It has been omitted for usability reasons. Creating, editing, storing and deleting of profiles can only be done within the profile section.



The screenshot shows a 'Profile Settings' form. At the top, there is a 'Profile:' label followed by a text input field containing 'dishwasher' and a search icon. Below this is a note: 'Note: must be a valid search string'. The form is divided into three columns: 'Country', 'Language', and 'Media source'. Each column has a list of options with a scrollbar. The 'Country' list includes International, Afghanistan, Åland Islands, and Albania. The 'Language' list includes Bulgarian, English, French, and German. The 'Media source' list includes Internet, Print, TV, and Radio. At the bottom of the form, there are two date input fields labeled 'Date from' and 'Date to', each with a calendar icon. A green 'create' button with a plus icon is located at the bottom right of the form.

Figure 21: Management of the user’s profile

The improved Semantic Search service now provides categories for all articles that can be used for grouping in the search section. Previously topics had been faked. Now real data is available.

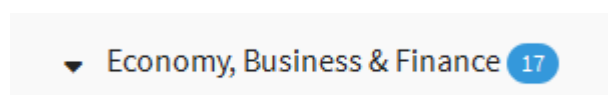


Figure 22: Categories to group the articles

For a set of articles grouped by category, a tag cloud can be displayed, in order to quickly get the main keywords and their importance.



Figure 23: Tag cloud of the articles' set

In the single article view, additional information is displayed for each article. Previously only media source, date, country and language could be displayed. Now the overall sentiment of the article and the category are shown as well. Furthermore, entities can be displayed by clicking a button.

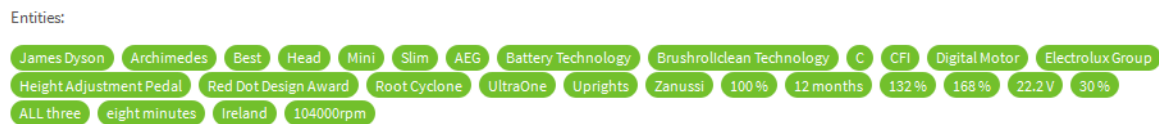


Figure 24: UC2 search page

Functionalities for translation and summary have been kept as they were working as expected.

The retrieval of only relevant articles for the analysis section has been implemented. Now only for relevant articles charts are displayed.

The following charts are available:

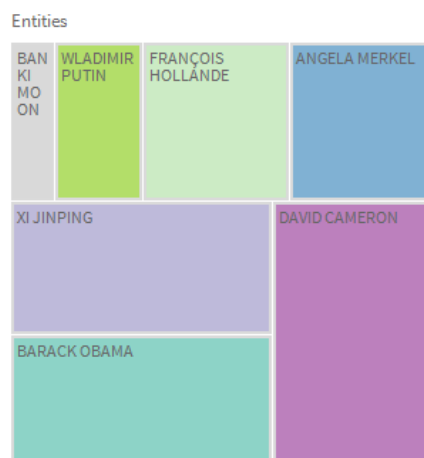


Figure 25: Tree map of the most important entities

Articles per country

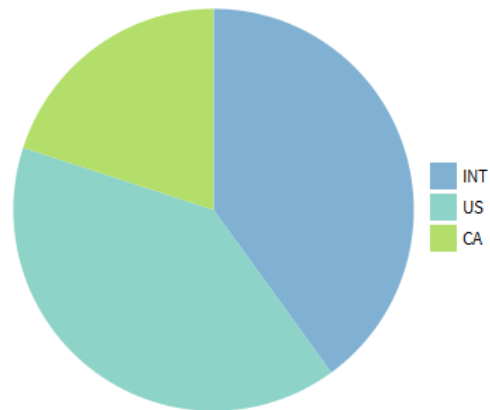


Figure 26: Pie chart for articles per country

SUBJECTS



Figure 27: Bar chart for categories

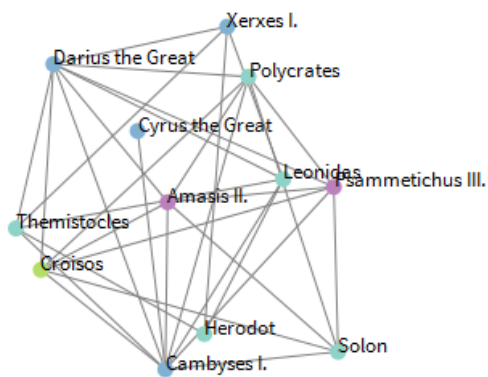


Figure 28: Network analysis chart (currently faked)

A new section “Influencer” has been introduced where the top 10 Twitter users for UC2 are displayed.

Twitter Influencer for UC2 – Household Appliances



Figure 29: Chart of the Twitter influencers

List of the integrated services:

- Profile service: Fully implemented.
- Semantic Search service: Provides sentiment, categories and entities for each article. Facets for entities and categories are also available for the analysis section.
- Translation service: Fully implemented.
- Extractive Summary service: Fully implemented.
- Abstractive Summary service: Implemented as an entity tag cloud for multiple documents.
- Sentiment Analysis service: Overall sentiment per article is implemented.
- Filtering service: Implemented for relevant/irrelevant articles. Also filtering of countries, languages etc. had been implemented in the previous prototype.
- Social Media analysis: For pre-defined hashtags, the most influential Twitter users are retrieved.

4.3 UC3: SME internationalisation Use Case

The UC3 application (see D7.1, section 4.2) is an application that should support SMEs, in order to start a process of internationalisation with any kind of products. Relevant information related to the countries, the economic situation of the market, the legal information, and the exportation/importation conditions should be retrieved easily to support decision making.

As the UC1 application, the description of the user interfaces have been provided in the D7.3 deliverable (see D7.4, section 4.3). In this version, the search system has been also switched to the knowledge repository instead of the CNR. In addition, the knowledge base has been improved with new indicators' datasets.

Since the First Prototype, the scenario of the SME internationalisation has been extended with new sectors and new products. Also, the numbers of indicators that should help the user to take a decision on which country it could be interesting to export his/her products has been extended. For this reason, the global design was adapted to reflect the new content structure and to facilitate the readability of many indicators.

Based on the NACE taxonomy¹³, 3 sectors were selected and for each sector, a list of products has been defined. The list of the sectors and products is structured in the following table:

Sector category	Sectors	Products
C - Manufacturing	C10 - Manufacture of beverages	Tea Coffee Beer Soft drinks Juice
	C11 - Manufacture of food products	Dairy products Cheese Meat Ice-cream Olive oil Bakery Vegetables Sugar Chocolate
	C13 - Manufacture of textiles	Animal (wool, silk), Plant (cotton, flax, jute), Mineral (asbestos, glass fibre), Synthetic (nylon, polyester, acrylic)

Table 32: List of the sectors and the corresponding products

Now, the user can select one of them and search for specific information.

¹³ http://ec.europa.eu/competition/mergers/cases/index/nace_all.html

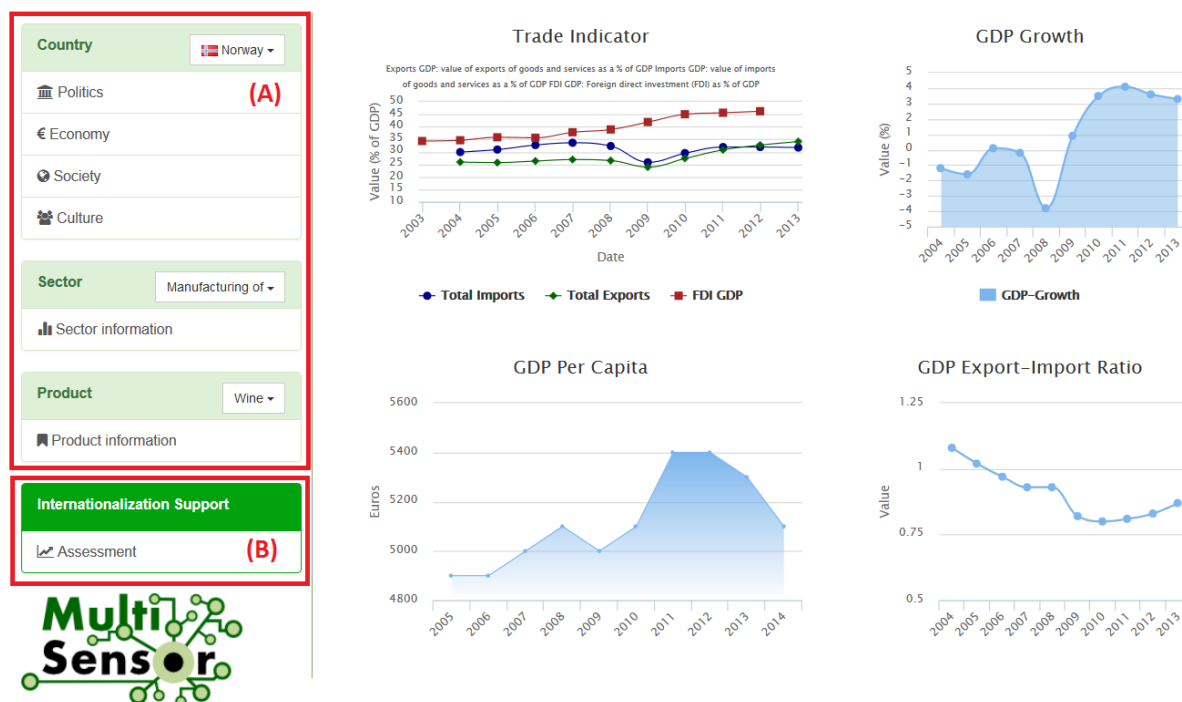


Figure 30: Overview of the UC3 application

The web application is now separated in two different kinds of activities (see Figure 29):

- (A) browse specific information to one country: for this activity, three main blocks of information are provided (Country's indicators, Sector information and Product information).
- (B) compare several indicators between two countries: this activity is related to the internationalisation support.

For the Second Prototype, the list of the indicators has been extended to provide more valuable information for the internationalisation support. Based on the deliverable D3.2 (Indicators for media monitoring and internationalisation), the indicators have been selected and organised by categories to depict the relevant information related to the target country: Politics, Economy, Society and Culture. All the categories and the corresponding indicators are presented in the following Table:

Category	Sub-category	Indicators	Graphical representation
Economic indicators	GDP	GDP growth	Line chart
		Real GDP growth rate – volume (tec00115)	Line chart
		GDP per capita in PPS (tec00114)	Line chart
		GDP per capita – quarterly Data (namq_aux_gph)	Line chart
		Exports of goods and services in % of GDP (tet00003)	Line chart
		Imports of goods and services in % of GDP (tet00004)	Line chart

		Export to import ratio (tet00011)	Line chart
		Inward FDI stocks in % of GDP (tec00105)	Line chart
	Importation / exportation	Customs and tariffs	Multidimensional lines chart
		Structure of taxes by economic function (gov_a_tax_str)	Multidimensional lines chart
		Export and Import	Multidimensional lines chart
		Current account – quarterly data (ei_bpca_q)	Line chart
		Harmonised indices – monthly data (ei_cphi_m)	Line chart
		Foreign Direct Investment	Line chart
Political indicators	---	Government type	Bar chart
		Political instability index	Bar chart
		Corruption perception index	Bar chart
		General government deficit (-) and surplus (+) – quarterly data (ei_nagd_q_r2)	Bar chart
Social indicators	Population	Life table (demo_mlifetable)	Bar chart vertical
		Human Development Index	Line chart
		Population with tertiary education attainment by sex and age (edat_lfse_07)	Bar chart with age groups
	Work	Unemployment rate	Line chart
		Harmonised unemployment rates (%) – monthly data (ei_lmhr_m)	Line chart
	Health	Life expectancy	Bar chart with age groups
		Life expectancy by age and sex (demo_mlexpec)	Bar chart with age groups
		Population distribution	Line chart
Cultural indicators	Urbanisation	Distribution of population by degree of urbanisation, dwelling type and income group (source: SILC) (ilc_lvho01)	Bar chart
	Consumption habits	Economic sentiment indicator (teibs010)	Line chart
		Households having access to the internet at home (isoc_pibi_hiac)	Histogram
		Easiness of doing business	Bar chart

Table 33: List of the indicators displayed per category

In the **Political category**, one interesting indicator is the Corruption Perception. The value of the selected country (such as Spain in Figure 30) is highlighted in a different colour and can be compared with all the other European countries.

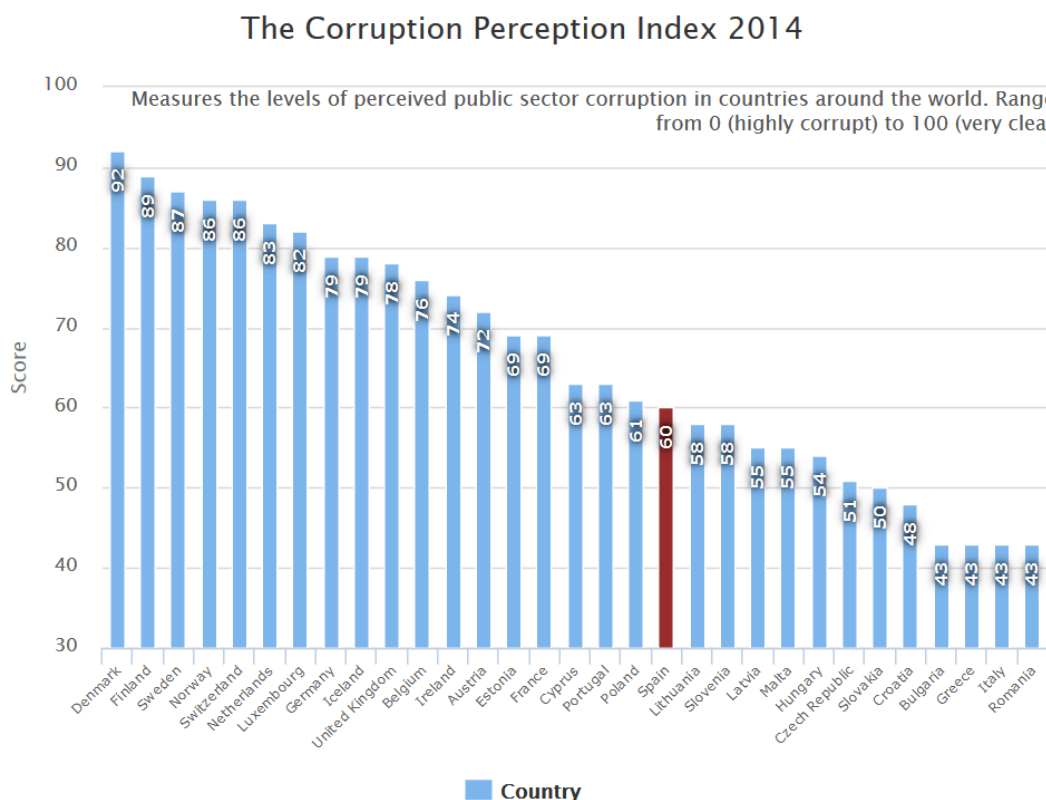


Figure 31: Political category: the corruption perception index (2014)

In the **Economic category**, the comparison between the GDP imports and GDP exports of the Goods and Services is displayed in a chart with a multidimensional representation: year by year, the user can evaluate the evolution of indicators' values.

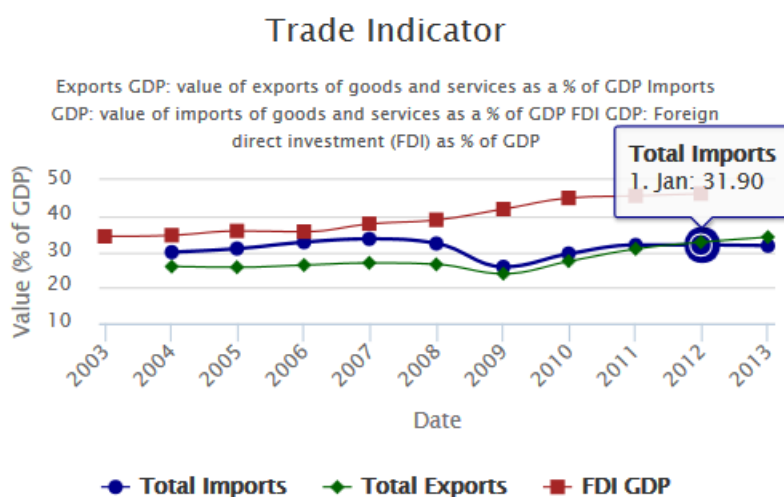


Figure 32: Economic category: Trade indicator

For the **Social category**, several indicators are provided. In Figure 32, two examples are provided. The first one is related to the life expectancy. This indicator is represented as an horizontal bar chart what age can be expected by a citizen according to his current age.

The second example is representing the evolution of the unemployment rates during a ten years period for a predefined country.

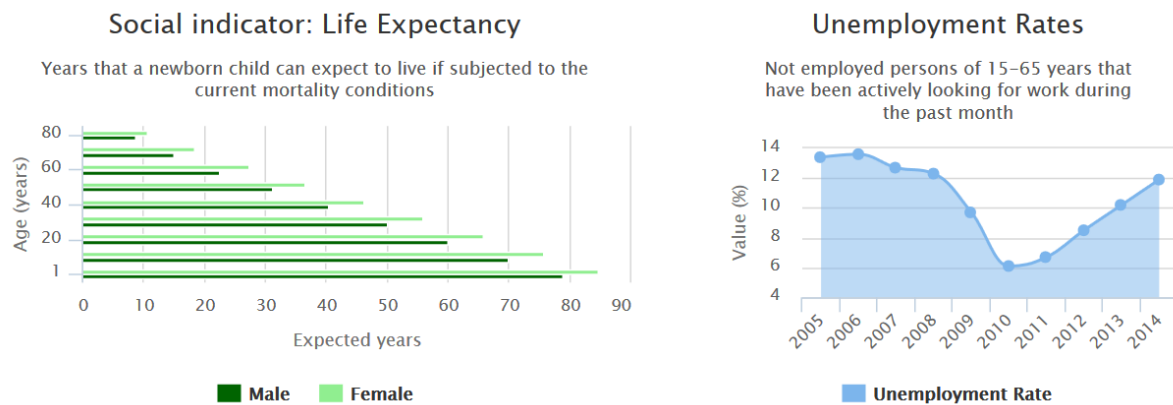


Figure 33: Social category:

For the **Cultural indicator**, the multi-bars chart represents the levels of the urban distribution and of the Internet access year by year. In order to understand the evolution of these values, all the average values are linked through a line (see Figure 33).

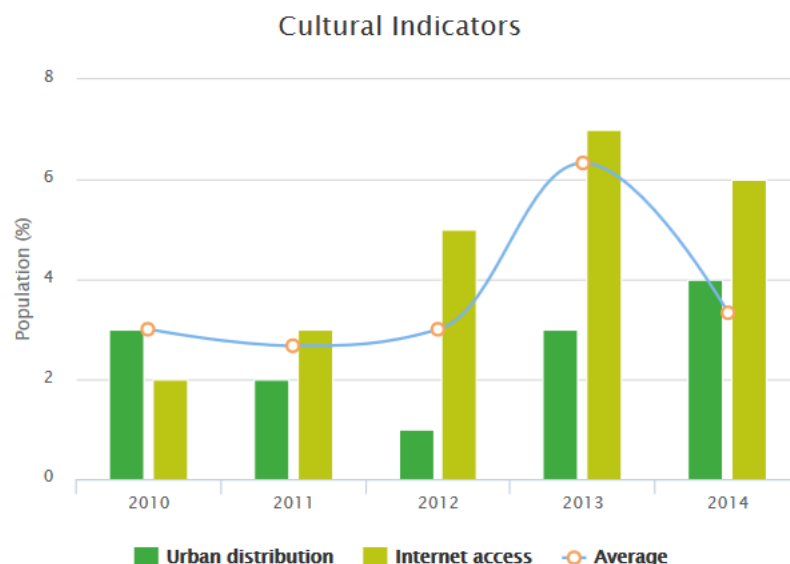


Figure 34: Cultural category: Urban distribution and Internet access

The second important functionality of the UC3 application is the indicators' assessment to support the Internationalisation activity. The SME people could be interested to target specific countries to establish new commercial activities. For this, the assessment panel permit them to compare the indicators between the two targeted countries.

Category	Indicator to be compared	Comparison conclusion
Economic	GDP Growth	Country that has the best value
	GDP PPS	Country that has the best value C2 flag
	Balance of trade	Country that has the best value
	Days to import	Country that has the best value
	Days to export	Country that has the best value
Social	Unemployment	Country that has the best value

	Urbanisation	Neutral indicator
	Internet households	Country that has the best value
	Total Population	Country that has the best value
Political	Type of Government	Neutral indicator
	Government Composition	Neutral indicator
	Years to Elections	Neutral indicator
Economic Culture	Consumption Confidence	Country that has the best value
	Ease of doing business	It's a ranking, the lower the number the better.

Table 34: List of the indicators that are compared in the assessment

In the web application, this indicators comparison is represented in a table in which every indicator's value is displayed for both countries. If the indicator is not a neutral one, a conclusion about which country has the best score is provided to the user. Finally, all the scores are summed and based on it, which country is the most suitable one for the internationalisation is provided as a conclusion (see Figure 34).




Table of indicators			
The selection of the correct country for the international investment depends on a number of indicators. These indicators are very important in order to make a first analysis of the different options that can be presented in a global market.			
#	Indicator	 Spain ▼	 Ucraina ▼
1	Merchandise Imports ⓘ	3.17 (2011)	15.39 (2011)
2	Clear imports ⓘ	5.5 (2005)	5.9 (2008)
3	Average days to import goods ⓘ	10 (2011)	33 (2011)
4	GDP Growth ⓘ	1,5 %	1,2 %
5	GDP-PPS ⓘ	45	35
6	Unemployment ⓘ	14 %	10 %
7	Internet households ⓘ	80 %	75 %
8	Total number of inhabitants ⓘ	10 M	42 M
9	Ease of doing business ⓘ	12	6
10	Balance of trade ⓘ	2002,6	4034,5
Decision Support			
Based on the above information, the most suitable country is: 			

Figure 35: New version of the Decision support's panel

The last important functionality of the UC3 application is the information search about the sectors or the products. The MultiSensor search engine can retrieve specific information about the UC3 topics. Then, the user is able to search for any keywords to retrieve the list of the relevant articles.

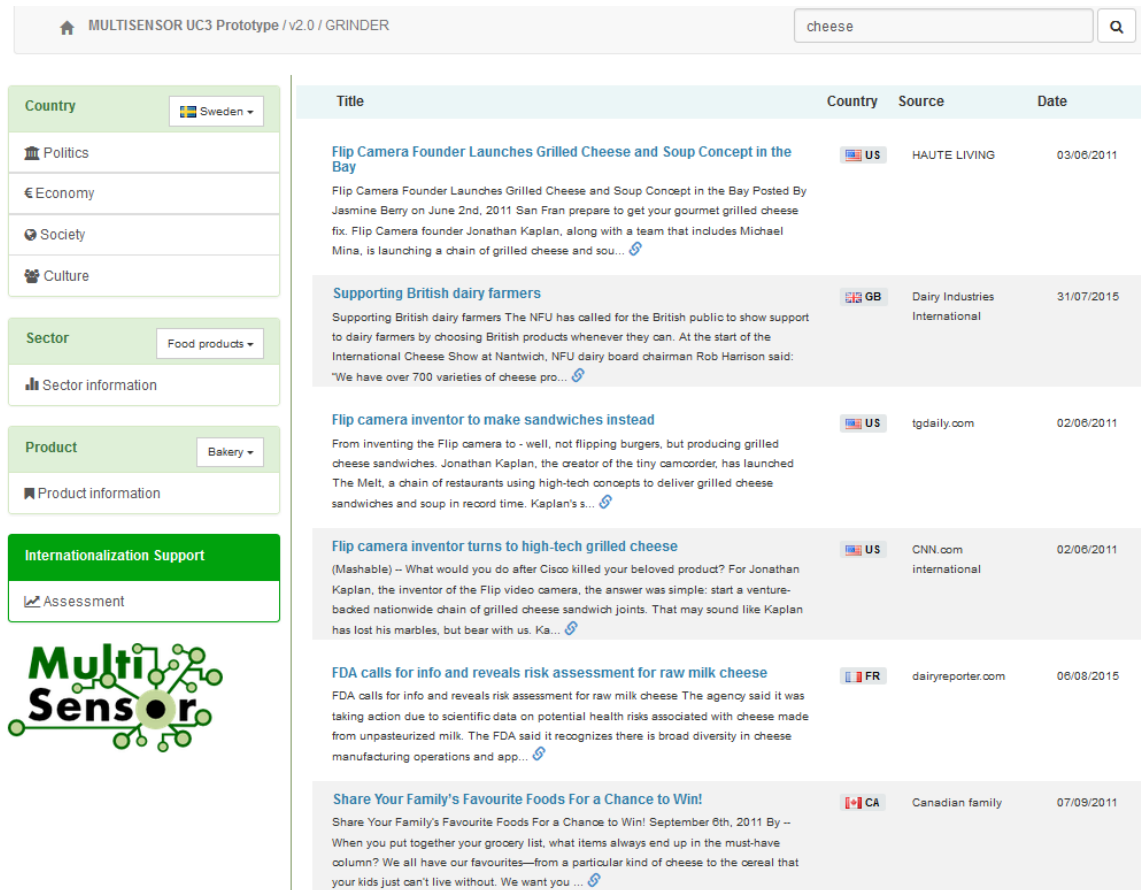


Figure 36: Page to search information about a product

Here is the summary of the integration of the Online Services with the UC3 application.

- Profile service: Not required for UC3.
- Contributor analysis service: implemented as an API that provides the detailed information of a Twitter handle, but it is limited in 3 requests per minute. This limitation does not permit a functional integration.
- Semantic search service: The search functionality is provided by ElasticSearch (CNR).
- Similarity search service: Not required for UC3.
- Translations service: Not required for UC3.
- Summaries service: Not required for UC3.
- Content delivery service: Fully integrated.
- Clustering + Filtering service: Not required for UC3.
- Reference data service: integrated as a data wrapper that collects the data from the Linked Open Data datasets uploaded in GraphDB. Results are obtained by querying the SPARQL endpoint with query templates. The list of indicators has been extended.
- Decision support service: integrated as a SPARQL query to compare specific indicators between two countries. The list of indicators has been extended.
- Community Detection: For pre-defined hashtags, the list of the detected communities is retrieved.
- Social Media analysis: For pre-defined hashtags, the most influential Twitter users are retrieved.

5 CODE ORGANISATION

5.1 Source tree layout (D7.4 updates)

All the MULTISENSOR code and related artefacts are kept in a Subversion¹⁴ repository in EVERIS premises and organised on a per-Work Package basis. The root of the source tree is located at <https://quark.everis.com/svn/MULTISENSOR/trunk>.

A breakdown of the layout of the repository follows.

- **wp1:** WP1 artefacts
- **wp2:** WP2 artefacts
 - **ms-svc-dep:** Dependency Parsing Service, Maven package (see section 3.3.3.5).
 - **ms-svc-extr:** Concept Extraction Service (see section 3.3.3.4).
 - **ms-svc-rel:** Relation Extraction Service (see section 3.3.3.6).
 - **nifutils:** Library to manage NIF formats, Maven package (complementary tool for the Concept extraction, Dependency parsing and the Relation extraction).
 - **ms-svc-conceptEventDetection:** : Concept and Event Detection Service, Maven package (see section 3.3.3.12)
 - **ms-svc-ner:** NER (see section 3.3.3.3)
- **wp3:** WP3 artefacts
 - **ms-svc-context:** Context Extraction Service (see section 3.3.3.4).
 - **ms-svc-contributorAnalysis:** Social Graph Service (see D7.2, pp. 42).
 - **ms-svc-sa:** Sentiment Analysis service (see section 3.3.3.7).
 - **ms-svc-communityDetection:** Community Detection module a part of SMAP service (see section 3.3.5).
 - **ms-svc-socialMediaAnalysis:** Influential User Detection (Social Media Analysis service see 3.3.5)
- **wp4:** WP4 artefacts
 - **ms-svc-categoryClassification:** Category Classification Service (see section 3.3.3.9).
 - **ms-svc-contentAlignment:** code related to Content Alignment pipeline (see D7.2, section 4.2.2.5).
 - **ms-svc-multimediaStructure:** indexing Service (see section 3.3.3.13).
 - **ms-svc-topicDetection:** Topic detection service (see section 3.3.3.9)
- **wp5:** WP5 artefacts
 - **ms-svc-decsupport:** UC3 Decision Support Service (see D7.2, section 4.2.3.4).
- **wp6:** WP6 artefacts
 - **ms-vc-summ:** Summarisation service (see D7.2, section 4.2.3.1).

¹⁴ See <https://subversion.apache.org/>

- **wp7:** WP7 artefacts
 - **crawler:** Crawler engine (see D7.2, section 4.2.2.2).
 - **ms-common:** Shared Java library and services for services.
 - **ms-crawler-socialmedia:** Yahoo! Crawler, Maven package (see D7.2, pp. 32).
 - **ms-js-common:** Shared Node.js modules and utilities.
 - **ms-parent:** Parent Maven package for all MULTISENSOR packages.
 - **ms-svc-cdelivery:** Content Delivery service (see section 3.4.1.1).
 - **ms-svc-refdata:** UC3 Reference Data service (see section 3.4.2.2).
 - **supervisor:** Supervisor Node.js (see D7.2, section 4.2.2.1).
 - **uc:** Use Case portals
 - **uc1:** UC1 Node.js application and related artefacts
 - **uc2:** UC2 Angular JS application and related artefacts
 - **uc3:** UC3 Node.js application and related artefacts
 - **uclib:** Shared Node.js modules and libraries for UC applications.
- **wp8:** WP8 artefacts
- **wp9:** WP9 artefacts

5.2 Continuous integration environment

Due to the involvement of different development groups and constant evolving of the services the automated build, testing and deployment become very critical tasks.

Advantages we have obtained:

- Build automation
- Transparency in between development partners
- CI newly checked-in code into a build
- CI run unit tests and rejects the build if they fail
- CI deploy builds to other servers
- Transparency in between development partners
- Integration with Jira issue tracking

Infrastructure for CI is based on Jenkins open source continuous integration server. It allows continuous integration by pulling newly committed code from SVN. Builds can be triggered either on a schedule or by hitting a URL.

5.3 Packaging

5.3.1 Java modules

All Java modules are packaged as Maven¹⁵ artefacts for automated build, test and deployment capabilities.

In order to keep dependency management in check and ensure consistent use of package and library versions, all packages in the MULTISENSOR platform use a parent package,

¹⁵ See <http://maven.apache.org>

wp7/ms-parent. This package provides versions for common dependencies and specifies shared build properties etc.

Additionally, a transversal module, **wp7/ms-common**, provides shared features for all services. This includes constants, common classes and interfaces, access to shared resources, wrappers to access common services, and more. All services must depend on this package.

Most notably, the ms-common package contains a Bootstrap class, which calls the supervisor to bootstrap into the platform, retrieving the shared configuration for coordination with the rest of the services.

5.3.2 Node.js modules

The Node.js modules are built as self-contained applications. They all have a package.json file, which describes their dependencies and allows using npm¹⁶ to download and install them. A special module, named **ms-js-common**, contains shared modules across the rest of Node.js applications.

¹⁶ See <https://www.npmjs.org/>

6 INFRASTRUCTURE

The Second Prototype is running in Amazon EC2 cloud infrastructure provisioned by EVERIS. Rationale and plans for scaling and provisioning are discussed in D7.2, section 5.

6.1 Current farm (D7.4 updates)

All servers run Ubuntu Linux 14.04.1 LTS (“Trusty”) on x64 architecture. Ubuntu is hugely popular and as such, Personal Package Archives (PPAs) and vendor repositories are readily available providing very recent versions of core packages of MULTISENSOR (mongodb, elasticsearch, nodejs).

6.1.1 Mscrawler1

The mscrawler1 hosts the Yahoo! Crawler described in D7.2, pp.32. This is a small server dedicated to crawling targeted sites using a combination of Hadoop, Nutch and HBase. It has been successfully deployed but not yet integrated with the main platform.

The crawler1 has the following specs:

- 1x x64 core (2 ECUs).
- 3.75 GB RAM.
- 32 GB local SSD storage (ext4).
- 100 GB EBS SSD storage (ext4).

6.1.2 Msgrinder1

An extra server, called **msgrinder1**, has already been commissioned for use in the platform but has been integrated into the prototype. It is now hosting the Content Extraction Pipeline Services, the repositories and the three UC applications.

The grinder1 has the following specs:

- 16x x64 core (52 ECUs).
- 122 GB RAM.
- 300 GB local SSD storage (xfs).
- 100 GB EBS SSD storage (ext4).

7 DEMONSTRATOR URLS AND INFORMATION

The following URLs can be used to access the different parts of the MULTISENSOR Second Prototype:

UC1 Application: <http://grinder1.multisensorproject.eu/uc1/>

UC2 Application: <http://grinder1.multisensorproject.eu/uc2/>

UC3 Application: <http://grinder1.multisensorproject.eu/uc3/>

SVN repository: <https://quark.everis.com/svn/MULTISENSOR/trunk/>

CEP testing tool: <http://grinder1.multisensorproject.eu/cepTesting/>

Credentials for access to the environments will be provided privately by other channels.

8 SUMMARY AND CONCLUSIONS

In D7.6, the status of the Second Prototype is presented. It explains the system in terms of repositories, services, processes and workflows. In addition, it includes some updates regarding the architecture and the integration of all the Offline and Online services. This can be summarised as follows:

- Most of the services have been significantly improved.
- New services have been developed, deployed and integrated.
- Development infrastructure has been improved with automated test building and deployment system.
- The system has been divided into production and development environments.
- The RDF repository has been populated with the extracted knowledge produced by the CEP.
- The UI for the three Use Cases has been improved with the interaction of the available online services.