LARGE-SCALE ELASTIC ARCHITECTURE FOR DATA AS A SERVICE

Project Number:	FP7-ICT-318809
Project Title:	Large-Scale Elastic Architecture for Data as a Service
Deliverable Number:	D5.4
Title of Deliverable:	Prototype of the distributed application for a single micro-cloud
Contractual Date of Delivery:	2014.09.30
Actual Date of Delivery:	2014.09.30

Abstract

The document presents the progress on the design and implementation of the representative user distributed application of the LEADS platform. It details the functionalities and requirements that form this application and drive the design of the LEADS components. Prior to describing the technical solution, this document provides a background on adidas business needs and on the unique characteristics of LEADS that could be leveraged to bring new opportunities to the company.

List of Contributors

Name	Organization	E-mail
Jacques Ohannessian	adidas	Jacques.Ohannessian@adidas-Group.com
Pawel Skorupinski	adidas	Pawel.Skorupinski@adidas-Group.com
Wojciech Barczynski	Cloud & Heat	wojciech.barczynski@cloudandheat.com
Marcel Gädig	Cloud & Heat	marcel.gaedigk@cloudandheat.com
Jens Struckmeier	Cloud & Heat	jens.struckmeier@cloudandheat.com
Anja Strunk	Cloud & Heat	anja.strunks@cloudandheat.com
Eleftherios Chatzilaris	TSI	echatzilaris@softnet.tuc.gr
Antonios Deligiannakis	TSI	adeli@softnet.tuc.gr
Ioannis Demertzis	TSI	idemertzis@softnet.tuc.gr
Minos Garofalakis	TSI	minos@softnet.tuc.gr
Nikolaos Giatrakos	TSI	ngiatrakos@softnet.tuc.gr
Ekaterini Ioannou	TSI	ioannou@softnet.tuc.gr
Odysseas Papapetrou	TSI	papapetrou@softnet.tuc.gr
Nikolaos Pavlakis	TSI	npavlakis@softnet.tuc.gr
Ioakim Perros	TSI	imperros@softnet.tuc.gr
Evagelos Vazeos	TSI	vagvaz@softnet.tuc.gr
Christof Fetzer	TUD	christof.fetzer@tu-dresden.de
André Martin	TUD	andre.martin@tu-dresden.de
Do Le Quoc	TUD	do@se.inf.tu-dresden.de
Frezewd Lemma Tena	TUD	frezewd_lemma.tena@mailbox.tu-dresden.de
Frank Busse	TUD	frank.busse@tu-dresden.de
Etienne Rivière	UniNE	Etienne.Riviere@unine.ch

Document Approval

	Name	Email	Date
Approved by WP Leader	Jens Struckmeier	jens.struckmeier @cloudandheat.com	2014-09-30
Approved by GA Member 1	Etienne Riviere	Etienne.riviere@unine.ch	2014-09-24
Approved by GA Member 2	Xiao Bai	xbai@yahoo-inc.com	2014-09-20

Contents

LIST OF CONTRIBUTORS II

DOCUMENT APPROVAL III

CONTENTS IV

LIST OF FIGURES V

EXECUTIVE SUMMARY VI

1. INTRODUCTION 1

2. BUSINESS NEEDS OF ADIDAS 1

2.1 BUSINESS NEEDS EXPLAINED 1
2.2 SUITABLE CHARACTERISTICS OF THE LEADS PLATFORM 2
2.2.1 Core characteristics 2
2.2.2 Business model that could complement the LEADS features 3
2.3 FUNCTIONALITIES DEFINED FOR THE REPRESENTATIVE ADIDAS APPLICATION 4
2.3.1 Functionality 1: Mining product dissemination and pricing evolution over time 4
2.3.2 Functionality 2: Mining the evolution of the product subjective reception over time 5
2.3.3 Functionality 3: Access path mining 5

3. MAKING SENSE OF RAW HTML DATA – CREATION OF A MEANINGFUL DATASET 5

3.1 ALL THE DATA NEEDED FOR THE DEFINED FUNCTIONALITIES 5
3.1.1 Page-level metadata 6
3.1.2 Site-level metadata 7
3.1.3 Keyword-level metadata 8
3.2 THE PROCESSING FRAMEWORK 9
3.3 USING A FRAMEWORK TO EXTRACT NEEDED METADATA 10
3.3.1 E-commerce 10
3.3.2 News sites and blogs 12
3.3.3 Real-time extraction of valuable information 14
3.3.4 Metadata for localization and language 14
3.3.5 Keyword-level metadata 15

4. APPLICATION IN PRACTICE – QUERIES, GUIS AND LIVE DATA STREAMS FOR ANALYTICS 15

4.1 QUERY AND VISUALIZE DATA ON TOP OF THE PLATFORM 15
4.1.1 GUI for pricing evolution over time 16
4.1.2 GUI for product dissemination over time 17
4.1.3 GUI for monitoring dissemination and the subjective reception of topics 17
4.2 ALTERNATIVE WAYS FOR GETTING DATA FROM THE PLATFORM 18
4.2.1 Query data and visualize the output in an in-house tool 18
4.2.2 Query data and download results 18
4.2.3 Set up continuous queries and get a stream of notifications 19

5. CONCLUSION 19

BIBLIOGRAPHY 19

List of Figures

Figure 1: Business needs around understanding the consumer 1
Figure 2: LEADS Platform and interactions with client 3
Figure 3: LEADS Platform supported by the expertise ecosystem 4
Figure 4: Levels of features to be stored as metadata 6
Figure 5: The processing framework prepared during work on the representative application 9
Figure 6: Common characteristics for majority of product offering pages, source: adidas.co.uk 12
Figure 7: General idea of mixing heavy MapReduce jobs and lightweight real-time processes 13
Figure 8: Visualization presenting minimal and maximal price of products with a given keyword in defined shops (based on partial crawls of shop domains in September 2014) 16
Figure 9: Visualization presenting number of products with a given keyword in defined shops (based on partial crawls of shop domains in September 2014) 17
Figure 10: Visualization presenting number of mentions of a given keyword in content of articles on pages in defined language (based on partial crawls of news/blog domains in September 2014). 18

Executive Summary

The document presents efforts that were made throughout the last year in order to prepare the representative distributed application on top of the LEADS platform. These efforts covered four important aspects:

a) Evaluation of platform features in context of requirements of adidas in field of big data analytics of adidas;
b) Definition of functionalities that would leverage unique features of LEADS and simultaneously fit the requirements of adidas;
c) Preparation of framework and components that enable large-scale classification and information extraction to create dataset required by functionalities;
d) Proposition of visualizations that would help business get insights out of the created dataset.

Business needs of adidas. Section 2 gives an overview of adidas business strategy and explains how big data platforms like LEADS are suitable to fulfil our requirements. We present a set of functionalities that we choose to implement on top of the LEADS platform.

Making sense of raw HTML data – creation of a meaningful dataset. Section 3 presents and categorizes data that is needed by the defined functionalities. Then, it outlines the processing framework and components that were implemented and combined together in order to extract these data from raw HTML content.

Application in practice – queries, GUIs and live data streams for analytics. Section 4 defines the ways by which the created dataset could be used by business in order to create meaningful insights and to be able to react quicker and better to the changes in environment.

1. Introduction

The LEADS platform, as a highly-distributed big data analytics platform, has characteristics that create new opportunities for business. Simple solutions for retrieving and pre-processing public data sources are of a great interest of companies in big data analytics era. We have evaluated those opportunities and placed them in the context of business requirements of adidas. The document presents our efforts to create functionalities that reuse publicly accessible data crawled and stored on the LEADS platform. We have extended every phase of the workflow where data is captured, processed, stored, queried and visualised in order to serve those functionalities. Together, they cover characteristic business use cases and could serve other companies in the future.

In the next sections of this document we will describe features unique for LEADS in context of adidas business needs; then we will present functionalities that we decided to implement on top of the LEADS. We will outline dataset required for the functionalities together with implementation efforts that were needed to create it. Finally, we present first visualizations based on the crawled, pre-processed dataset and propose alternative models for leveraging it.

2. Business needs of adidas

Consumer is in the main focus in adidas business strategy (see: Figure 1). In the big data era, new opportunities for understanding dynamically-changing needs of consumer appear. Processing power of machines and vast spectrum of available algorithms make machines capable of taking “educated guesses” based on amounts of data that highly exceed analytical capabilities of human beings. adidas wants to make full use of those opportunities and to take the analytics results into account when making strategic business decisions. As this section presents, LEADS could create novel chances for us and for data-driven and consumer-oriented business in general.

2.1 Business needs explained

adidas has experts in multiple domains that look for the ways to get insights based on the most relevant datasets. Naturally, it is the end consumer that is in the limelight of the use cases for tools. However, there is a vast spectrum of aspects around him that need to be considered in the process of decision-making.

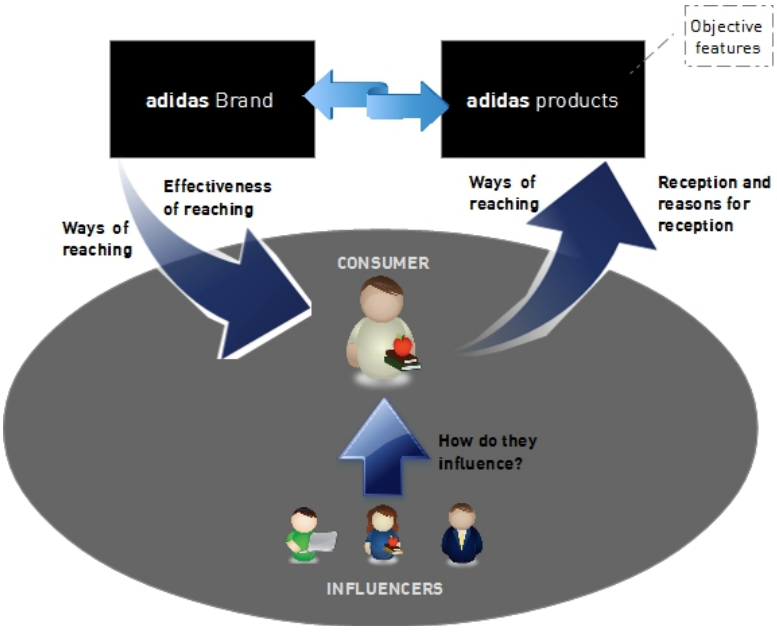

Figure 1: Business needs around understanding the consumer

In the recent years, World Wide Web became the primary source of information about products for consumers. Many of consumers started using blog and microblog platforms to publicly share their opinions on their purchases. This process has changed the market into more dynamic and consumer-centred space. Consumers expect more individualized products along with individualized experience during the process of purchasing them. They are used to the great variety of articles and expect new, more exciting, and more innovative solutions all the time. In the times when everyone can share information that can potentially reach millions, it is much easier for consumers to become influencers on a large scale. adidas, as a consumer-oriented company, adjusts its efforts to target potential consumers as well as understand what their opinions are, and how to influence or engage them.

The quick spread of Internet access and development of Internet technologies caused that a lot of signals can be tracked by analysing the public and private data from various sources. The current market offers a great number of tools to analyse effectiveness of reaching the consumer. There are also plenty of tools that give access to knowledge about ways a consumer can reach information on the products, as well as what reception she has on them.

Even with a great number of tools on the market, evaluation of the factors that influenced each decision and opinion of consumer is still challenging. We know that a very important role is played by objective features of product (like price) and "subjective features" – like opinions of influencers (which could either be world-wide experts, private contacts of a consumer, or collective intelligence made up of unit opinions of many consumers). Here, the complexity level of data is often very high from a computational perspective. The growing amount of data to analyse requires more effective usage of the machine power to recognize more patterns, find more clues and insights. For that demand, a new kind of Big Data platforms appeared; platforms that let combine all of the structured and unstructured data sources and run algorithms to perform hypothesis-based analysis (so called "educated guesses").

Every purchase of a product can be presented as a complex model that consists of a couple of stages. Among others, that includes ways how brand reaches the consumer, ways how other sources can influence him, or the reasons why he personally wants exactly that product. We are able to create a real in-depth insight only when each of those factors is considered in detail. For that, we need an environment that allows easy merging of big amounts of data and custom algorithms. The more customized process we create, the more value we can get out of it; but also the higher privacy requirements we need.

2.2 Suitable characteristics of the LEADS platform

The market of Big Data tools and platforms is extremely dynamic and game-changers with new ideas appear there every couple of months. They offer new approaches, greater precision, greater numbers of possible data sources, and innovative algorithms. If LEADS succeeded to become a product one day, it would land in a dense market. Therefore uniqueness, simplicity of usage, and clear competitive pricing model would be the must-haves in the business model of the platform and are worth considering already during the current research stage. According to our evaluation, the platform as it is planned right now would have a couple of features that could bring a new value to the market.

2.2.1 Core characteristics

The aspects that define the LEADS platform and that are essential from the business perspective are:

1. LEADS would be one of the first vendors to offer a simple access to the versioned copy of publicly available Web resources on such a large scale (see: WP2);
2. LEADS would allow clients to read and append metadata describing the content and type of any resource represented by its URL either manually via the browser plugin (see: WP1) or by executing custom platform plugins (see: WP3 and WP5);
3. With efficient crawlers (see: WP1) and scalable processing engine (see: WP3 and WP4), LEADS would be one of the biggest providers of near-real-time online public data. Combined

with ability to upload and process the private encrypted data (see: WP3), LEADS would offer a great environment for executing the complex Big Data-driven use cases.

These key characteristics of the platform create unique values for the business. By pre-processing the content with components running information extraction and data mining algorithms (by either core platform components or custom clients' components) clients could:

- In near real-time obtain reliable information where in the web the keywords of their interest recently appeared;
- Execute and get access to a customized in-depth analysis of the mentions;
- Execute an in-depth analysis of the Web graph provided by the WGS.

For the most demanding clients of the platform, those features could potentially drastically reduce the complexity of the process of answering an analytical query and enable automatic adjustments of those results as soon as new data appear. That would, however, require lots of the algorithmic expertise as well on the side of a client. For those clients who could not or would not like to invest that much, LEADS platform could still be interesting for simpler use cases with its pay-per-query model. The implementation of such a system would not be trivial and would require large investment.

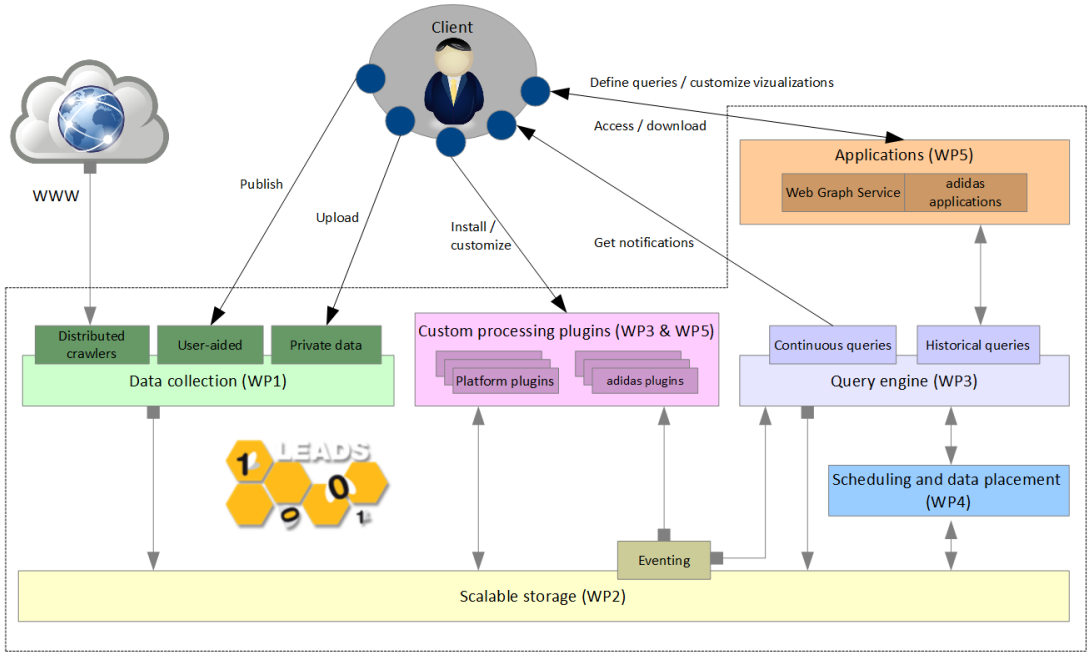

Figure 2: LEADS Platform and interactions with client

In Figure 2 the LEADS architecture from client perspective is presented. Most of presented interaction points fall into the main focus of the research project. To fully exploit platform capabilities, client could at first extend the LEADS data collection by uploading his private (encrypted) data and publish additional web content through the publishing interface (WP1). Then, he could customize the platform processing capabilities by installing new components and configuring already existing ones; consequently running them on the subsets of historical data as well as on freshly crawled data (WP3 & WP5). Thereafter, he could start defining queries on the platform data (including data stored by his components) (WP3). Their output could be either visualized by the customized applications (WP5) or downloaded (full result, or notifications in case of continuous queries).

2.2.2 Business model that could complement the LEADS features

The companies that we consider as the target group of the LEADS platform, are non-IT companies (such as adidas) as well as small IT companies. Lots of those are interested in leveraging richness of

information in public space that would be stored in the LEADS storage. However, since HTML format encodes visual presentation of documents, it is a complex task for those companies to make a use of those. They do not have expertise and are not interested in investing money in extraction of valuable content from the irregularly changing terabytes of unstructured data. Therefore, as a potential client of the platform, adidas proposes to create an ecosystem for companies with expertise in web scraping and data analytics, and also other fields necessary for the data processing. They could take the LEADS framework, extend it with custom processing components and run those on live and historical web data. They would earn money by providing focused analytics results to customers in pay-per-access.

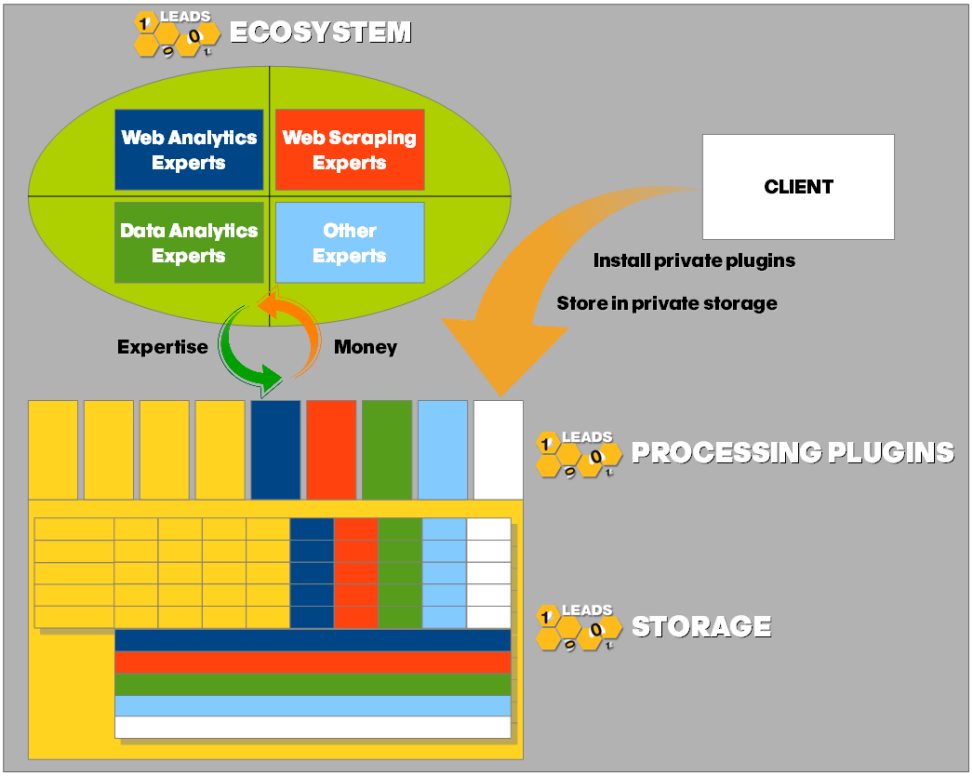

Figure 3: LEADS Platform supported by the expertise ecosystem

2.3 Functionalities defined for the representative adidas application

We defined three functionalities that comprise the backbone of the public data analysis of a consumer-oriented manufacturer, such as adidas. All of those functionalities in general form are already available in many tools. The LEADS differentiators are its openness, the size of available dataset, the scale of customization, and the ways they could be combined together to provide the real value for the business. The openness would ensure that the platform can be extended and companies can easily find partners to adapt the platform to their needs. All of those aspects are going to be inspected in detail in the remaining months of the project and presented in the final deliverable.

2.3.1 Functionality 1: Mining product dissemination and pricing evolution over time

This functionality focuses on objective information about products. Ecommerce sites are the principal source for such information. The main goal that can be achieved with this functionality is to cover: determining the time of appearance and disappearance of specific products in various online shops as well as detecting pricing trends of products. It is a big opportunity for a manufacturing company to have information on where and for how much their products are sold all around the globe. As there are already specific tools for that, the LEADS platform could offer more with: (1) pay-per-

query model; (2) possibility of querying pricing data together with any other data about products inside of the platform; (3) possibility of acquiring specific innovative views on data. Those would be interesting opportunities for specific data science use cases.

2.3.2 Functionality 2: Mining the evolution of the product subjective reception over time

This functionality focuses on the finding subjective perception of a product on the Web, and, in the next step, on the consumers who shape the perception by actively sharing their opinions about the product. Main sources of data are blog sites (and other kinds of social media that allow public access to their content) and news sites for more information about the product and related topics (facts about the product as well as events and news about the brand). The main goal of this web data analysis is: determining the time of appearance of information about the products as well as understanding the context of mentions of these products. For adidas, it is very important to recognize trends about mentions in real-time and how the trends develop over time. Those trends could consider count of mentions, varying context in which keywords were mentioned, and varying emotions around them. That also should be put in historical context. Again, there are tools that could offer a lot of that. LEADS could bring more opportunities with: (1) real-time notifications based on complex filters (like: if a price of a product is drastically reduced, notify on mentions that seem to be influenced by that change); (2) dataset that would contain the history of entire indexable Web. That would again introduce new possibilities for data science models.

2.3.3 Functionality 3: Access path mining

To evaluate the usefulness of the information that is retrieved from the Web Graph, we identified the third functionality. From our observations, the most valuable information can be extracted from the web graph when it is browsed backwards. It gives a possibility to look from where a requested page is linked. Therefore, the definition of the goal from a business perspective is to extract, group and analyse the paths through which a consumer can reach a page about a product/campaign. Unique opportunity would be given by: (1) grouping sources based on any dimension (e.g. sentiment of text around links, language of page, category of site); and again, (2) ability for mixing these data with other data (e.g. inbound links to product page with mentions of that product's name).

3. Making sense of raw HTML data – creation of a meaningful dataset

This section presents the technical efforts we have made that were essential to enable functionalities that would serve the business; and when talking about functionalities, actually we mean the workflow of capturing, processing, storing, retrieving, and visualizations of data – the key drive for the decision process.

3.1 All the data needed for the defined functionalities

Deliverable D1.2 presented the solution in which platform clients could store tags about each URL through the browser plugin. In this deliverable, we focus on the ways of using regularities of web content to automatically tag pages with information referenced as "metadata".

The main goal of LEADS would be to let business people find every little piece of valuable information available in the indexable Web (and not only). To make this piece of information meaningful for business use cases, it should be defined in a multi-level context. The reason is that the mention of some keyword on one page might have a totally different meaning than when it is found on another page or even on another part of the same page.

A keyword might have been mentioned in a title or in a comment (page-level metadata) of the post belonging to the blog site (site-level metadata). Those mentions would be potentially subjective, so it would make sense to count the sentiment around them (keyword-level metadata). However, the same keyword could be found in a product name (page-level metadata) on a page belonging to

Ecommerce site (site-level metadata). There, the objective feature, such as product price of (page-level metadata) would be much more meaningful than subjective features. Furthermore, a given keyword could have some meaning in one of the languages and it would make sense to treat mentions of pages in that language differently (page-level metadata) or even on sites from countries using that language (site-level metadata).

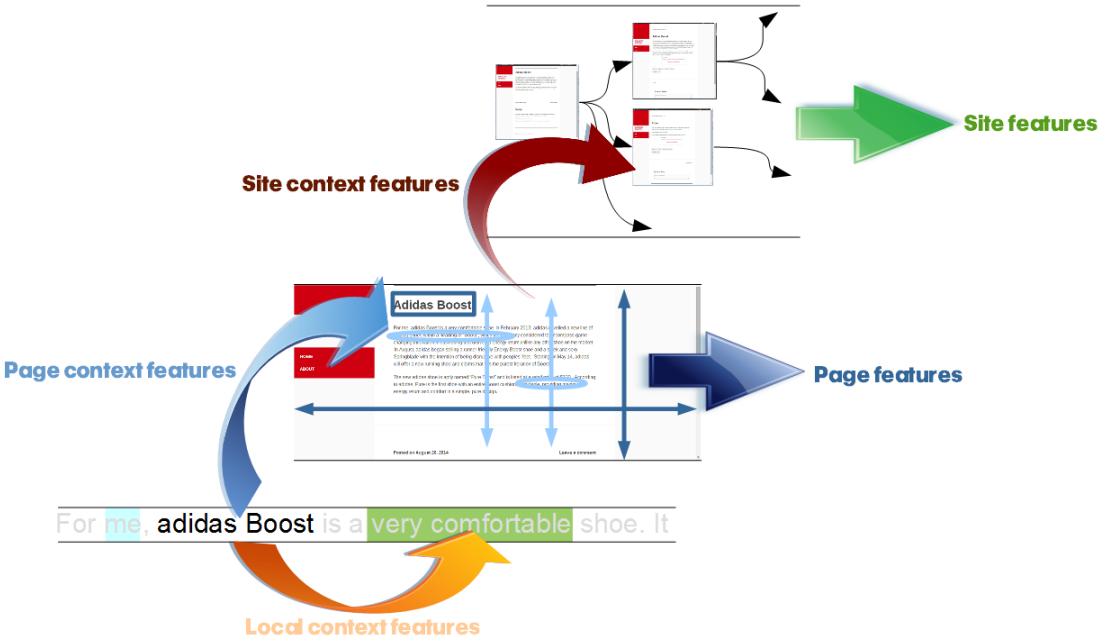

Figure 4: Levels of features to be stored as metadata

3.1.1 Page-level metadata

Every person browsing the web could probably intuitively categorize pages into a couple of types. Looking for news in the web, we can encounter pages that list main news along with pages containing one specific article. We can also start looking for opinions on a given topic. We will encounter blogs, where we can find main pages with a list of posts, as well as pages that present a full post. If we decide to purchase some product, we are going to find pages with product offering and those presenting a list of products of some category. From this short discussion, we can see that the support for the page categorization would be helpful. It would help to focus analytics on particular type of web data. The business intelligence could use the categories to distinguish the value of product mentions (e.g., a private opinion versus a product description) and how to interpret them in the analysis.

The content of Web pages is mostly presented in natural languages. A language is indeed an important feature by which pages can be grouped. However, intuition of the “page language” can be surprisingly misleading in case when a page description and content in navigation panel are written in one language and an article is written in another. Search keywords may vary between languages (unless they are proper nouns). Therefore keyword search components would depend on information about the page language.

The Nutch crawler stores web pages as it is received from the server – together with the HTML tags (see: D1.3). The page structure (defined by HTML tags) is an essential feature for interpreting web page. However the full text extraction of the web page content has its applications as well (e.g., determining already mentioned page language feature).

To interpret individual pieces of information about product, we need to recognize which web pages belong to the same web site. It is not trivial to define the strict frames of the web site. However, an

intuitive approach would be to take a fully qualified domain name for the page and store it as the site identifier. We foresee business queries that would be limited in scope to a set of sites.

Sentiment analysis components can evaluate the content of each page and tag it as more positive or negative. That would make sense mostly for pages containing lots of text content (like articles, reviews or comments).

On most of the pages, we could define parts that bring some additional value and those that we can simply ignore (e.g., navigation panels and other often repeated content within pages). The type and meaning of those valuable parts depend on the page type. The interesting parts of news article pages along with blog post pages would be an article (and its parts like title, date, etc.) and comments. Parts of content that would be valuable for us on Ecommerce product offering pages would be name and price of the product, and potentially images, description, comments and reviews.

Very meaningful information about the page would be provided by the PageRank algorithm implemented in WP3 (see: D3.2). It allows clients to filter web resources by popularity or recognize those pages which popularity grows very quickly.

An overview of page-level metadata is presented in Table 1. Every row represents one plugin for the platform.

Requirement	Reason	Solution
Page web site	Direct use – filtering/grouping; Helpful in determining page functional category	URL analysis, content analysis
Textual content	Input for language detection or sentiment analysis component	Filtering HTML content from tags and scripts
Content language	Direct use – filtering/grouping; Input for text mining algorithms depending on a language	Available language detection libraries (check against generated language profiles)
Sentiment analysis	Direct use – filtering/grouping	Natural Language Processing libraries (see: D3.3)
Page functional category	Direct use – filtering/grouping; Input for content parts extraction algorithms	Classification based on structural features, keywords and URL
Page topical category	Direct use – filtering/grouping	Classification based on numerical statistics like TF-IDF or neighbourhood in web graph
Content parts	Direct use – filtering/grouping	Depends on a page (see: Section 3.3.1 and 3.3.2)
PageRank	Direct use – filtering/grouping	See: D3.2

Table 1: Overview of page-level metadata

3.1.2 Site-level metadata

We evaluate correlation of extracted features of pages of a site very useful in a process of determining features of that site. In addition to that, we might extract metadata about every site from external sources.

Web sites can be categorized in a similar manner to pages. Every site can be categorized according content that it delivers – both, considering function of a site (Ecommerce, blog, news site, etc.) or topical category (sport, business, etc.). As with page categorization, the business need for that is to be able to query only content of sites of some category or, on the other hand, to be able to compare mentions of a keyword based on the site type.

For business, it is important to precisely recognize which markets are targets of particular content of the site. Many sites target a few – often very specific – markets. Others should be tagged as global; however, some of those sites are structured in a way so that every particular section of it targets another country. adidas would like to query information independently for each market. In this way we would be able to evaluate our position, and additionally find specific opportunities and threats on every of those markets.

We propose to extend PageRank algorithm (see: D3.2) to the site level by summing up the value of PageRank of all of pages of the site. As on the page level, it would allow clients to filter web sites by popularity or recognize those sites which popularity grows very quickly.

Overview of site-level metadata is presented in Table 2. Every row represents one plugin for the platform.

Requirement	Reason	Solution
Site category	Direct use – filtering/grouping; Helpful in determining page functional category	Analysis of features extracted from pages of that site
Site country (target markets)	Direct use – filtering/grouping	URL analysis, external sources (WhoIs databases, GeoIP[1]), languages of pages
Site PageRank	Direct use – filtering/grouping	See: D3.2; Summarization of PageRank values of all pages of the site

Table 2: Overview of site-level metadata

3.1.3 Keyword-level metadata

Site-level and page-level metadata give opportunity to select interesting set of crawled data and categorize it into subsets. Keyword search followed by keyword-level metadata extraction completes the former in process of creation of a meaningful dataset. In general, by keyword-level metadata we mean either features of direct surroundings of keyword mentions (such as sentiment) or features of a mention in context of the page (such as relevance).

One of the basic measures to represent the surroundings of a mention is a sentiment metric. A value of sentiment indicates a degree of positivity or negativity of a mention. Sentiments of all of the mentions of a keyword on a page would be then usually merged to count sentiment of a keyword in the context of that page. It is one of the most intuitive indicators for business for general reception of a product or brand.

A way to represent a keyword in context of a page is its relevance. In short, the more a keyword seems to be an important topic of the page, the greater is its relevance value.

The context of a keyword can be also represented in a more descriptive way – for example, as a list of adjectives appearing in its surroundings.

Some of queries that business would like to ask should be narrowed to specific parts of content of pages. Therefore, we propose to store a keyword together with a part of a page where it appeared (e.g. title, article or comment).

Overview of keyword-level metadata is presented in Table 3. Every row represents one plugin for the platform.

[1] See: https://www.maxmind.com/en/geoip2-databases

Requirement	Reason	Solution
Basic representation for keyword surroundings	Filtering/grouping/ trend visualizations	Natural Language Processing libraries (for sentiment analysis see: D3.3)
Descriptive representation for keyword surroundings	Further in-depth analysis of how keywords are mentioned	Natural Language Processing libraries
Basic information about keyword in context of the page	Mainly filtering	Text search/Indexing libraries and Natural Language Processing libraries

Table 3: Overview of keyword-level metadata

3.2 The processing framework

The processing framework was implemented with awareness of the following facts:

- There are many various possibilities for storage and processing logic;
- Platform as well as client plugins should work properly independently of strategic optimization choices on storage and processing layers.

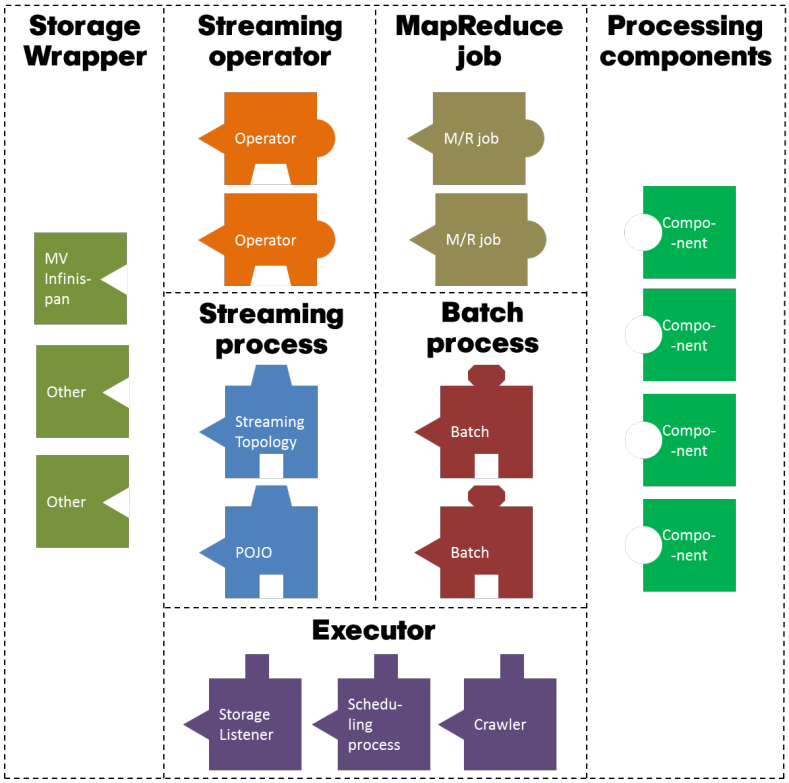

Figure 5: The processing framework prepared during work on the representative application

In general, a client might choose one of two models for data processing:

- Batch processing – use the MapReduce model that would spread computation tasks of client over micro-clouds. An example for this use case is finding features of a web site based on processing a random subset of pages belonging to that site;
- Stream processing – programming paradigm for real-time computations. It would be powerful in means of letting client process any page of his choice and extract valuable information

as soon as it is crawled. An example for a use case is extracting an article from content of a news page and finding/calculating topic, sentiment, language, etc.

There are a variety of options in which those models could be implemented on the platform. The ones chosen for the research project purposes are MapReduce over Infinispan for batch processing and listener calls based on Infinispan eventing for stream processing (see: D2.4). MapReduce jobs could be run either on demand or automatically based on a scheduling plan provided by a client (e.g. once a day). Stream processing model used within the project assumes listeners calling processing plugins that are executed locally.

Other possible options for running processing tasks that could be considered is by using Nutch plugins facility to run custom MapReduce jobs as a part of crawling process. That would be especially efficient when output of component execution should influence Nutch crawling decisions (like which page to crawl next).

Design of the processing framework is presented on Figure 5.

3.3 Using a framework to extract needed metadata

Previous sections presented business web analytics needs of adidas and corresponding LEADS platform functionalities that helps to address them. Then, we described dataset that would be needed to execute those functionalities. This section will show how we used the presented processing framework in order to obtain the needed dataset.

3.3.1 E-commerce

Online shops require very specific information retrieval model. Fortunately, we can extract a lot of features from content of their pages and use the features to recognize them.

In the proposed implementation, for classification of E-commerce sites as well as information extraction from pages, we use both batch processing and stream processing.

A key part of the process of extracting valuable information from E-commerce is a batch process that runs on every site on which more than *n* pages have already been crawled (notice that in the current version every subdomain is treated as a separate site).

The process goes through the following steps:

- If a site is not yet classified (or needs to be re-classified), check if it is E-commerce site:
  - Retrieve URLs of *k* pages of the site;
  - Draw *m* random pages and retrieve their freshest content;
  - For each of those pages, extract from its content the features characteristic for pages of E-commerce site; (*)
  - Count how many of the pages have at least a couple of those characteristic features. If factor of those pages in the site exceeds a value *p*, assume that this is an E-commerce site;
- If a site is assumed to be E-commerce:
  - Take again the set of *k* drawn pages and analyse their content. The purpose of this analysis is categorisation of pages into two subsets: “product offering pages” and “product category pages”; (**)
  - Take all of the pages categorized as product offering pages and find a model for valuable information extraction for that E-commerce site. (***)

The steps presented above contain a couple of complex algorithms that are based on the prepared models. Discussion on values of parameters *n*, *k*, *m*, *p* is out of scope of this document.

The first model (*) is a definition of characteristic features for E-commerce sites. It is approximate to the one described in A9.com patent [1]. In our model the features to characterise an E-commerce page are:

- *Add to bag* functionality existence – this feature is recognized by checking appearance of language dependent features (dictionary with a list of words that are used for describing "Add to bag" depending on page language) in text and attributes of "clickable" HTML elements;
- *Bag* functionality existence – for now, this feature is recognized by checking appearance of language dependent features in attributes (similar to *Add to bag*) of very specific HTML elements in the DOM tree;
- Number of times a word "price" is used in page text content (depends on language);
- Number of times a word "price" is used in attributes of HTML elements (depends on language);
- Number of times the words indicating *wishlist* are used in anchor elements;
- Number of times the words indicating *warranty terms* are used in anchor elements;
- Number of times the words indicating information on *shipping* and *returns* are used in page text content;
- Number of times the words indicating information on *taxes* are used in page text content;
- Number of times the strings looking like price values are found in page text content.

When they are extracted, every page is tentatively tagged as E-commerce or non-E-commerce. Since *Bag* and *Add to bag* are very strong indicators of E-commerce, if they are found on a page, the algorithm would not seek much more for confirmation that this is E-commerce page. However, if a site has a very specific way of how those functionalities are defined in HTML, then most of the keyword-based features would need to be present to tag a page as E-commerce.

The second model (**) is used to cluster pages of Ecommerce site into those that offer products and other ones. Because of the amount of noise caused by variety of structures of pages in almost every site, this cannot be done without some false classifications. Nevertheless, the main goal is here to filter as many pages without product offers as possible, to have a better dataset for finding appropriate model for information extraction in the next step.

To perform clustering, the state-of-the-art *k-means* algorithm is used. Observations based on which the clustering is executed are those features of pages that usually differ between product offering and category pages, namely:

- *Add to basket* button – typically, on category pages, there is either none or many buttons of this type. On product pages there is usually exactly one;
- Mentions of word "price" – usually there are more on category pages;
- Mentions of strings looking like price values – usually there are more on category pages;
- Number of images on a page – usually there are more on category pages.

When a subset of pages assumed to be product pages is determined, they become an input for a component responsible for finding a specific way for the site to extract product name and price (***). This task is non-trivial. However, during research a couple of common characteristics for great majority of E-commerce sites were found. They are presented in Figure 6.

The algorithm for product name extraction is based on standard in E-commerce that product name can be found in a page title. During evaluation of which node in a page DOM tree is the one that really presents the product name, a couple of things are taken into account: (1) very often this name is placed in HTML header element (h1, h2, …); (2) it is never an anchor (in contrast to recommended products); (3) it is placed in the same place on many pages (in contrast to mentions of product name in comments); (4) the text string of the node contains (almost) only product name (in contrast to mentions of product name in comments).

When the node containing product name as well as the node with add-to-bag button are known, the algorithm for finding a node with product price is executed. At first, all of the text nodes are checked for containing a string indicating a price value (e.g. $100, € 99.99, 89.50). Then all the candidate nodes are scored based on a distance from other nodes (proximity to product name node and/or

add-to-basket button is rewarded most). Afterwards, best candidates per every page are cross-checked with other pages and the ones that seem to work best for most of the pages are chosen.

At this point, best candidates for product names and product prices in DOM trees of pages of the site are ready to be stored in the LEADS storage (see: WP2) in form of XPaths.

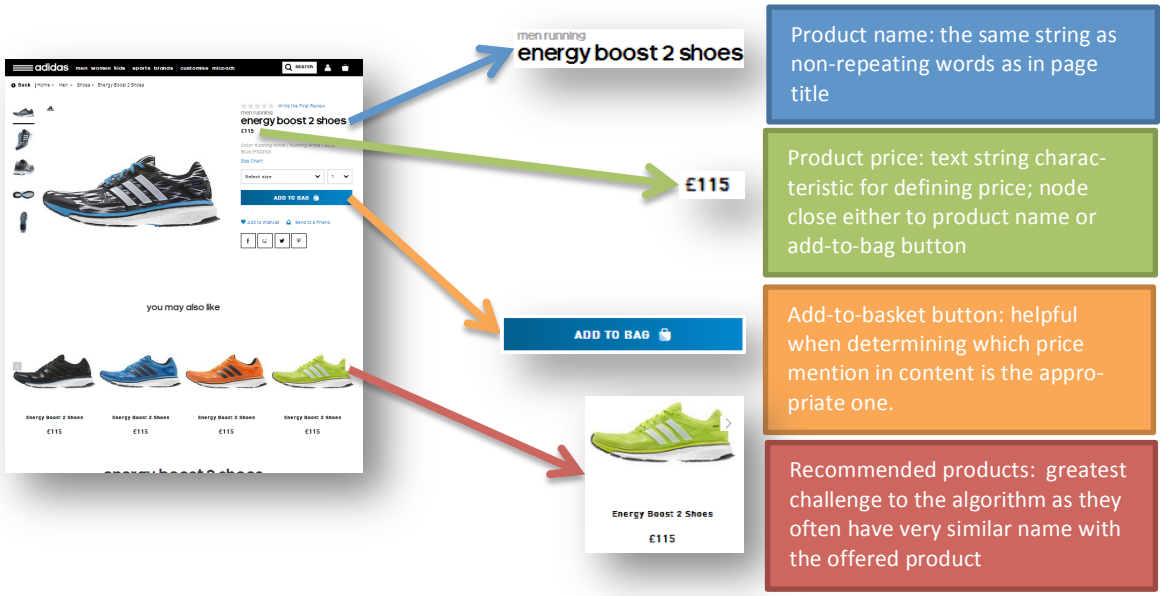

Figure 6: Common characteristics for majority of product offering pages, source: adidas.co.uk

3.3.2 News sites and blogs

In this domain, the lesser progress was made hitherto and further research will be probably left out of scope during the project. For the moment, we use a simple method for classifying a site as news site (or blog with news) based on external data source and appropriate information extraction needs intervention of a user. Available libraries for article extraction are tentatively evaluated as not precise enough for the large scale extraction that would happen automatically on LEADS and a lot of fixes would need to be implemented to provide results that might appear to be good enough.

The way we chose to classify a site as a news site (or news-like blog) is a good example of reusing external resources/services to enrich the LEADS dataset. Every site that is not yet classified is searched in the Google News feed. If some articles from pages of that site can be found there, the site is evaluated as a News site. That is an example of a very simple solution for enlarging dataset that can encounter problems when run in extremely large scale that is planned for LEADS. A politeness policy would need to be worked out for that case.

During the past months, we have tested and evaluated one research approach as well as one state-of-the-art approach on extracting information from blogs. Although we have achieved some interesting results, they are still far from being sufficient for letting a fully automated process extract articles and comments from pages. Moreover, we see opportunities unconsidered in that research which would appear when lots of data from every domain is constantly crawled as it is assumed for the LEADS project.

A part that appears to be the greatest challenge in the process of extracting articles is the evaluation whether a page is the one holding an article. Although there are state-of-the-art libraries for article extraction called Goose[2] and Boilerpipe[3] available, which achieve good results while extracting arti-

[2] See: https://pypi.python.org/pypi/goose-extractor/

cles from article pages of many domains, they give random output (false positive) when given a page that does not contain an article. Algorithms executed by those libraries examine documents by evaluating measures like words per DOM node, sentences per DOM node, words per sentence, words per line, along with types of HTML elements that contain that text.

On the other hand, there is a recent research from another FP7 project presenting a library called SD-algorithm [2]. It goes slightly beyond the state-of-the-art and tries to determine structure of a page (article / article with comments / multiple articles) and following that – to extract more of valuable information than other libraries. It is based on similar measures to previously mentioned algorithms. Again, the library achieves decent results but they are not good enough to be relied on.

During the research, we have put some effort on extending the way how SD-algorithm works by taking extraction details from each page of the site and try to figure out the extraction model based on a couple of assumptions that were mostly derived from E-commerce extraction:

- Pages with articles have the same URL pattern (e.g. http://domain/year/month/day/articlename or http://domain/articlename.html);
- Valuable parts of the content are placed in the same DOM nodes all around the pages;
- When extracted content is repeated between pages, it is a part of the navigation panel.

Following those rules for the site level and SD-algorithm-like algorithm on the page level, we claim that the decent article and comments extraction algorithm could have been implemented. However, we did not set the high priority for this task and it is still waiting in the queue.

For purposes of the demo, we decided to prepare a component that runs Boilerpipe library on every page that is crawled. For some pages, extracted information page extracts an article and its title using Boilerpipe library and then per every Web site a human input is needed on definition of the pattern for URLs of pages with an article.

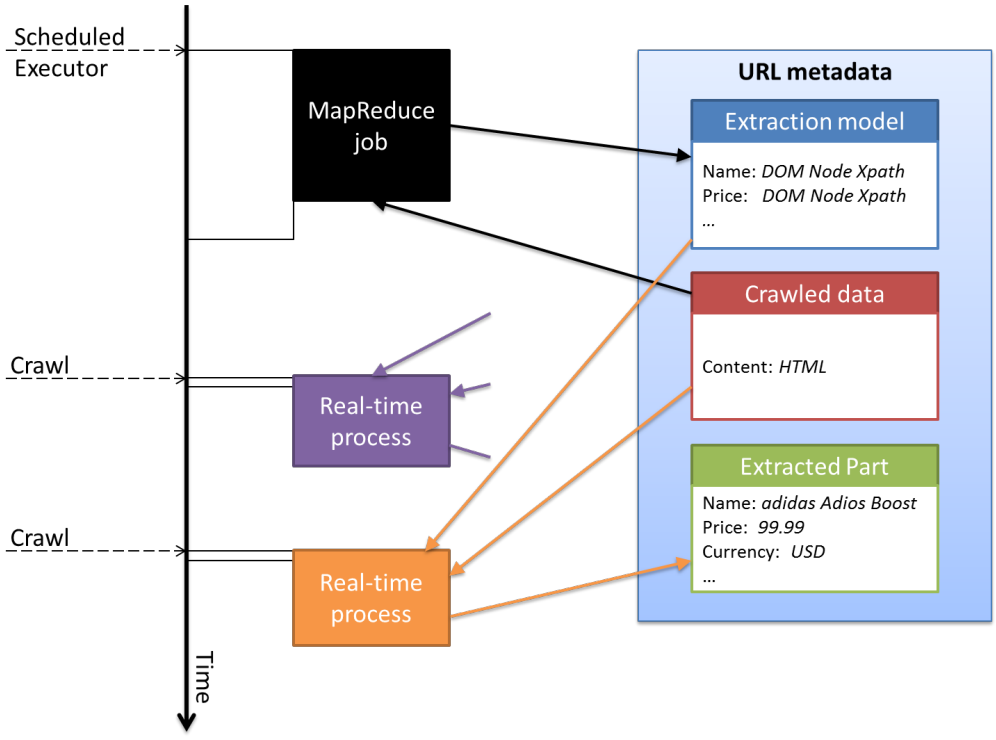

Figure 7: General idea of mixing heavy MapReduce jobs and lightweight real-time processes

[3] See: https://code.google.com/p/boilerpipe/

3.3.3 Real-time extraction of valuable information

An effort in domain of E-commerce as well as preliminary research in domains of news and blog sites had one main goal –be able to extract valuable data at low cost in nearly real-time. For now, that model works for E-commerce sites (see: Figure 7).

As discussed in Section 3.2.1, when a site is classified as E-commerce, we determine the model for name and price extraction during a batch process run according to the schedule. Every page of that site is going to be checked whether it belongs to the "product offering page" cluster and, if yes, have a product name and price extracted right away just by grabbing the content of DOM nodes determined by the batch process.

The process of extraction is easy to extend. Hence, it would be simple to enhance the process by taking into account, e.g., images or comments. We can also adapt the process to work on other page categories.

3.3.4 Metadata for localization and language

Not only would function of a web site be important information for business, but also markets it targets. Recognition of target markets is tricky when static content of pages is the only input. We could find the target market with decent probability using following heuristics:

- Check languages most commonly used in the domain;
- Extract names of geographic locations from content;
- Check for special sections that list countries;
- Check for a contact page to get official information about the owner (a valid contact page is mandatory in Germany, Austria, and Switzerland).

Language detection is a well-researched topic and therefore a standard solution using language detection library[4] could be chosen in order to tag every page with a language it is written in. Other content analytics for determining localization are out of scope of the project as they are evaluated as being too complex comparing to the value that they bring.

Another way for guessing localization is by parsing URLs. The possible opportunities are given by:

- Country-specific TLD names (e.g. www.adidas.de);
- Grouping of domain resources by country and language in multinational domains (e.g. http://msdn.microsoft.com/en-US/)

The first solution is not straightforward, as there are more and more non-country-specific names for TLDs and many country-specific domains are being sold to subjects from outside of that country (e.g. the site of Polish radio is registered under Micronesian TLD[5]). However, we find the first solution sufficient for the time being and decided to not pursue the second, more complex approach.

The solutions presented above are still far from providing even a decent image on which markets could the site be focusing on. Fortunately, there are additional opportunities based on external data:

- Site domain's IP address can be converted into country of the closest servers hosting the site (using MaxMind GeoIP[6]);
- Information on registrants and registrars of a web site can be queried using WHOIS protocol.

The components implementing those two functionalities where prepared for the project purposes. What one needs to be careful with is that usage of MaxMind GeoIP can bring some false positives, as the big international companies from overseas often store a copy of their content in European locations. Therefore, information on servers' locations would need to be verified against other resources like whois databases. Those are databases that keep records on formal details of every web site. What is essential, quite often they include information on country of registrar and/or registrant. The

[4] See: https://code.google.com/p/language-detection/
[5] See: http://www.rmf.fm/
[6] See: http://dev.maxmind.com/geoip/

way that we have chosen to access easily all of those databases is scraping the pages retrieved from the all-whois.com site.

To sum up, statistics about languages of pages of the site, countries mentioned in sites' registry, country where closest servers hosting the site are placed, and country of site's TLD (if applicable) are the extracted features that might be used for determining site's main target market.

3.3.5 Keyword-level metadata

For the time being, we use two most basic measures for keyword in a given context. The first measure is the sentiment analysis that we discussed in deliverable D3.3 (WP3). The second measure is an approximate relevance of keyword in context. This measure was developed within WP5. This algorithm is based on algorithm for content-based ranking of a keyword for search engines [3]. We use those basic metrics to approximate keyword relevance in context of an article:

- Word frequency (how many times each of keywords is used);
- Location within document (whether keywords are mentioned in title, at the beginning of an article or somewhere in the end);
- Word distance (if there are multiple keywords, the closer they are from each other, the more relevant an article is).

4. Application in practice – queries, GUIs and live data streams for analytics

Figure 2 (on page 4) presents interactions of client with the platform. The ways in which she could get results of processing executed on LEADS are as follows:

1. Querying historical data and visualizing the output on the platform (using built-in Web Graph Service capabilities);
2. Querying historical data and visualizing the output with an in-house tool;
3. Querying historical data and downloading the results. Clients might be interested in reusing the data for in-house processing;
4. Setting up continuous queries and getting their results in forms of notifications.

Notice that client could also set up continuous queries and store their results in a private table on the platform. Consequently, she would be able to query those results in historical queries.

In this section, we go through the aforementioned ways of handling results. We show how adidas could make use of them in the context of functionalities for the platform that we have defined. We focus mostly on the opportunity for visualizing data on top of the platform as a part of the Web Graph Service.

4.1 Query and visualize data on top of the platform

On top of the entire workflow of crawling, pre-processing, classifying, extracting and storing data, we created components that run queries and feed data visualization components. This way we have created a prototype for the entire process of translating noisy, unstructured web data into potential business insights. Once we already have information extracted, we can define perspectives to be used by data analysts who could bring valuable insights. For purposes of this deliverable, we have prepared three visualizations: two in the context of the functionality 1 (see Section 2.3.1) and one in the context of the functionality 2 (see Section 2.3.2).

Querying and visualization of data inside of the platform is considered a default use case for the end user. In such a scenario, data analyst could create a query using Apatar (see D3.3), choose an appropriate visualization and utilize it in order to develop a new business insight. In the considered use case, the entire process would happen with no engagement of in-house data centres of the client company.

The figures that we show in this section present visualizations of the real data – the result of crawling and pre-processing during evaluation of the prototype in September 2014.

4.1.1 GUI for pricing evolution over time

The first user interface that we have prepared allows user to check how price of a product is changing over time in shops defined in his query. Along with it, user can compare evolutions of prices of various products, depending on keywords that he has defined.
This is a simple SQL presentation of a query that serves as an input for the visualization of the discussed functionality (notice that because of performance reasons and limitations of currently used CQL language, the queries look different in practice):

```
SELECT site_uri AS shop_name, product_name, product_price_min,
product_price_max, product_currency, timestamp
FROM leads.page_core pc
JOIN leads.ecom_product ep ON pc.uri = ep.uri
AND pc.timestamp = ep.timestamp
WHERE shop_name IN (<<list of shops>>)
AND timestamp BETWEEN (<<time boundaries>>)
AND product_name CONTAINS (<<list of keywords>>);
```

User can put values of parameters of the query in the form. After choosing the values and confirming the query a visualization of data is generated (see: Figure 8). In menu on the left side, user chooses products of which prices he is willing to visualise. Data analysts could use this functionality in order to recognize when B2B partners of adidas as well as our competitors offer discounts on specific products. Moreover, they could directly see comparison of our product prices with prices of counterparts of other companies.

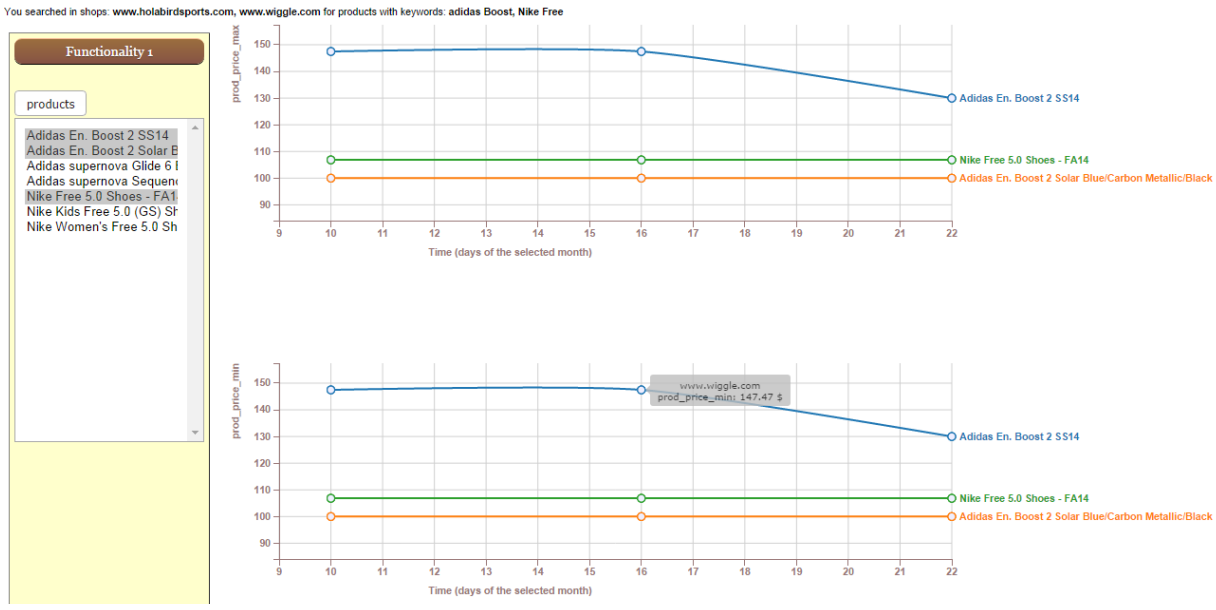

Figure 8: Visualization presenting minimal and maximal price of products with a given keyword in defined shops (based on partial crawls of shop domains in September 2014)

4.1.2 GUI for product dissemination over time

The second user interface presents another view on E-commerce data. This time, we focus on dissemination of products represented by given keywords in each of the chosen countries.
In order to retrieve information needed for visualizations, we query our database with queries similar to the following simple SQL:

```
SELECT country, site_uri AS shop_name, product_name, timestamp
FROM leads.page_core pc
JOIN leads.site s ON pc.site_uri = s.uri
JOIN leads.ecom_product ep ON pc.uri = ec.uri
AND pc.timestamp = ep.timestamp
WHERE site_country IN (<<list of countries>>)
AND timestamp BETWEEN (<<time boundaries>>)
AND product_name CONTAINS (<<list of keywords>>);
```

User is able to define parameters in the form. After choosing values for parameters and confirming the query, visualization is generated (see: Figure 9). In menu, user can choose a country of his interest and see how many versions of a product represented by his keyword are offered in the shop at that time. This way, data analysts would be able to recognize how many adidas versus our competitors' products are offered in any online shop.

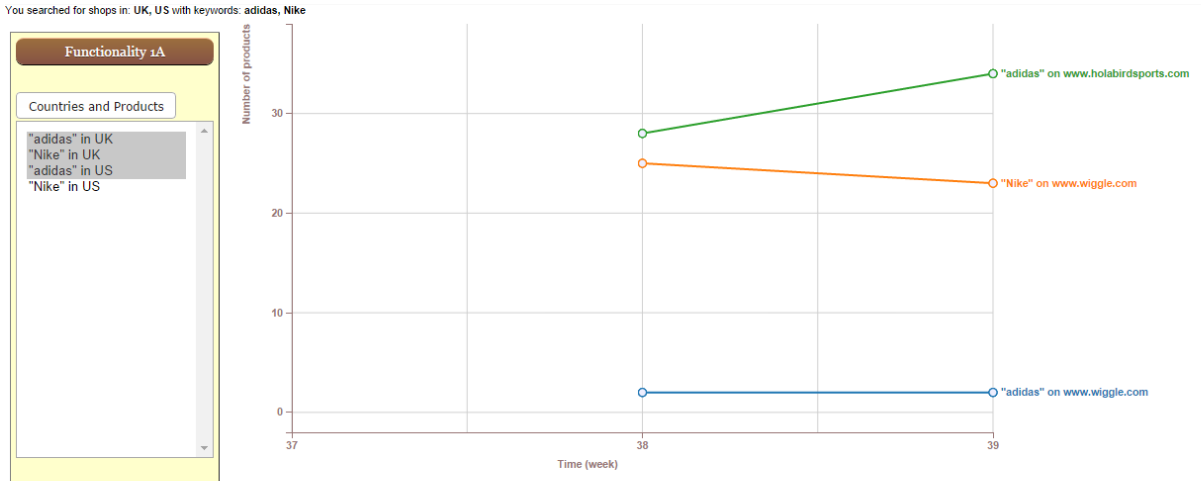

Figure 9: Visualization presenting number of products with a given keyword in defined shops (based on partial crawls of shop domains in September 2014)

4.1.3 GUI for monitoring dissemination and the subjective reception of topics

Another user interface prepared for the M24 deliverable deals with monitoring of dissemination of keywords as well as subjective reception of topics. User can see which sites mention topics of his interest and get details about basic analytics of context of those mentions.
The following simple SQL query represents an interaction with the database in order to get the relevant data. Keywords which user can choose need to be defined beforehand, so that engine can search them and extract from content as soon as any page is crawled.

```
SELECT site_uri AS site, keywords, timestamp, relevance, sentiment
FROM leads.page_core ps
JOIN leads.page_part pp ON ps.uri = pp.uri
AND ps.timestamp = pp.timestamp
JOIN adidas.extracted_keyword ek ON pp.uri = ek.uri
AND pp.timestamp = ek.timestamp and pp.partid = ek.partid
WHERE language = <<language>>
AND timestamp BETWEEN (<<time boundaries>>)
AND keyword IN (<<list of keywords>>)
AND relevance > ((<<relevance value>>);
```

User can parameterize the query in the form. After choosing the values and confirming the query, the visualization is generated (see: Figure 10). User can see how many times each site has mentioned each of the defined keywords on weekly basis and what the average sentiment was.

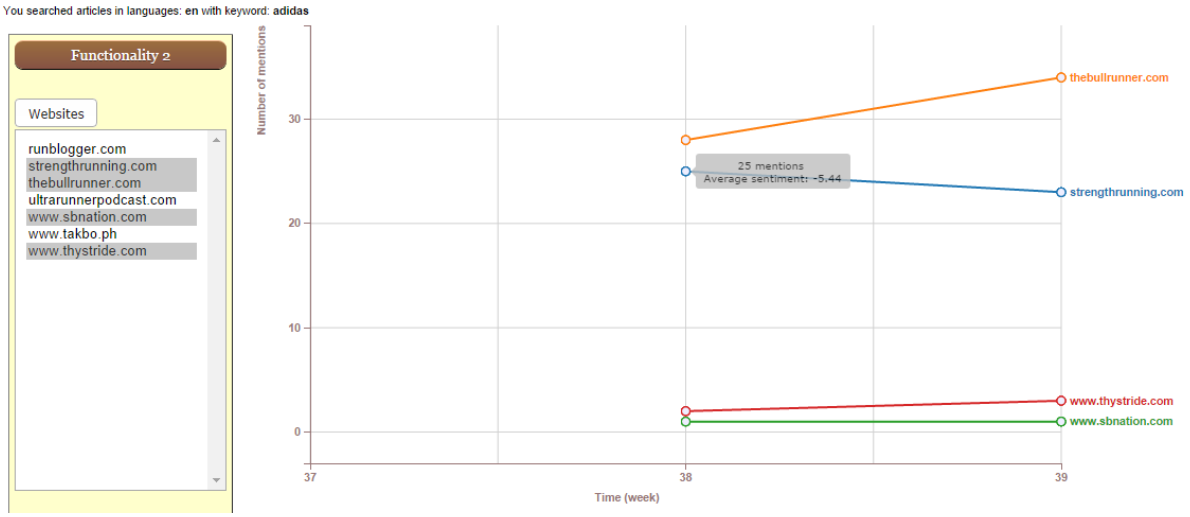

Figure 10: Visualization presenting number of mentions of a given keyword in content of articles on pages in defined language (based on partial crawls of news/blog domains in September 2014)

4.2 Alternative ways for getting data from the platform

Running visualization service on top of the platform is not the only possibility to allow user leverage the dataset pre-computed and stored on the platform. There are plenty of use cases where other models for cooperation would be preferable. Those are shortly described in this section.

4.2.1 Query data and visualize the output in an in-house tool

Analysts in companies use a set of Business Intelligence tools in their day-to-day work. Examples of popular tools are Tableau or MicroStrategy. Those tools have a set of connectors to data sources, usually based on ODBC drivers. Analysts can then mix data from various sources and visualize them together to bring new business insights. An opportunity for LEADS would be to create ODBC driver on top of the query engine, so that end users would be able to query LEADS storage from their favourite BI tools.

4.2.2 Query data and download results

Another option that would be attractive for business is a possibility to download results of queries into internal (or another external) database. Companies need models produced by data mining and machine learning algorithms in order to leverage internal data. It is often the case that those models could be enriched with data coming from public external sources. In that case, LEADS could play a

role of a source for pre-processed public data. LEADS query engine would basically become a filter that end user would apply in order to make sure that only the relevant data are downloaded.

Data scientist could use results from queries of the functionality 1 in order to define additional features in his model for describing a product. Dataset of the functionality 2 could empower models for evaluation of campaigns, categorization of consumers, and description of products. Dataset of the functionality 3 could extend models for evaluation of campaigns or description of products.

4.2.3 Set up continuous queries and get a stream of notifications

The model that would create next interesting opportunities for clients would be based on extending LEADS real-time streams, so that they could directly reach either inbox of analyst (alert/notification) or feed internal decision-making systems.

Before notification reaches the destination, every piece of information would go through a set of filters that could be defined with SQL-like queries. Every piece of information that matches all of the rules set by filters would be specific enough to directly influence business decision. For example, a notification based on a query such as “get adidas product that was discounted for more than 20% in at least 3 of 10 TOP E-commerce sites of Germany during last 24 hours” could directly influence a business decision on making a discount on that product in adidas online shop. Another notification based on a query such as “get a person sponsored by adidas who is a subject of drastically growing number of positive/negative mentions in the last couple of minutes” could accelerate the reaction of the business on some achievement or misbehaviour of one of the stars sponsored by adidas.

5. Conclusion

This document presented the effort of adidas to provide the representative application of the platform. That includes tasks enabling classifications of sites and pages, extractions of valuable content, and analysis of context of mentions. Together with efforts of other work packages and visualizations we have created a prototype for the full workflow from capturing unstructured data from web to presenting clear insights to potential platform clients.

Throughout the year, we have achieved our goal which was implementation of the functionalities which we have defined in work document WD5.2. The project is not mature enough for us to start taking business decisions based on the dataset that we have gathered. However, visualizations show that it is possible with the LEADS model to achieve functionality of state-of-the-art tools. We believe that with the unique features of the platform, LEADS could bring additional values.

For months M25-M30 we are going to focus on further adjustment of the use cases that leverage public data. We want to showcase how the LEADS platform and its processing model could help achieving real business goals. We will also spend time on evaluating whether all of the functionalities are working properly on multiple micro-clouds.

Bibliography

[1] D. R. Bailey, A. Rajaraman and T. J. Feldman, “Search engine system and associated content analysis methods for locating web pages with product offerings”. USA Patent US7395259 B2, 1 July 2008.

[2] N. Pappas, G. Katsimpras and E. Stamatatos, “Extracting informative textual parts from web pages containing user-generated content,” New York, 2012.

[3] T. Segaran, Programming Collective Intelligence, 2007.