D7.6

Integrated USEMP Platform v2

v 1.2 / 2017-02-20

Noel Catterall (HWC)

This document provides information on the final version of the Integrated USEMP Platform, which constitutes part of the deliverable alongside the code release available at 'https://usemp.hwcomms.com'.
It integrates the comments made during the final project review.

Project acronym	USEMP
Full title	User Empowerment for Enhanced Online Presence Management
Grant agreement number	611596
Funding scheme	Specific Targeted Research Project (STREP)
Work program topic	Objective ICT-2013.1.7 Future Internet Research Experimentation
Project start date	2013-10-01
Project Duration	36 months

Work Package	WP7
Deliverable lead org.	HWC
Deliverable type	Prototype
Authors	Noel Catterall (HWC)
Reviewers	Alexandru Ginsca (CEA) Eleftherios Spyromitros-Xioufis (CERTH)
Version	1.2
Status	Final
Dissemination level	RE: Restricted Group
Due date	2016-09-30
Delivery date	2017-02-20

Version	Changes
0.1	Initial Document
0.2	Sections Updated
0.3	Document Restructuring
0.4	Revised DataBait Images
1.0	Internal Review Revision
1.1	Review Alterations
1.2	Final Review Alterations

Table of Contents

1.Introduction

This prototype deliverable serves as a report alongside the release of the second version of the integrated USEMP platform deployed for the final pilot studies. This document gives an overview of the system delivered within the design goals set out within the second architectural design document (D7.4), and serves as a companion to the system code base available at 'https://usemp.hwcomms.com'.

After the introduction, the first section examines the architecture of the deployed system. A variety of issues are covered: the role of each server in the DataBait system, the way servers are configured, intricacies of the DataBait account management, database schema and the information flow around the system at the most critical time of account creation, when the initial data retrieval requests are made for a user.

The second section gives a description of integrated modules developed within other work packages, and finally, the last section gives an overview of the user interface, and how data is passed back to the user in an intuitive format.

2.Prototype Overview

This section gives an overview of the architecture of the first integrated USEMP platform as delivered in real terms with the design goals set out within the second architectural design document (D7.4). As the development of the system progressed both D7.4 has been updated, and several elements defined within D7.4 have been redefined and re-scoped as necessary. That is, integration of components meant certain elements had to change to adapt to the individual components' architecture, additions were also made in order to cover additional features not previously decided upon.

2.1. System Functionality

For the second prototype, the functional elements are as follows:

- System Services
- Social Media Integration
- Textual Geo Location Data Mining
- Image Concept Detection
- GUI Web Interface
- Privacy Scoring Framework
- File Upload Integration

System Services

These elements provide core functionality within the backend system necessary for the system to work as a whole. Many of these components are described in more detail later in this document.

Social Media Integration

For the purposes of social media, the system has been designed to support multiple social networks, however, due to the scope, breadth of information, Facebook has been taken as the primary source. Facebook also has the most comprehensive API for collection of both Textual Data and Imagery Data, so allows for use of all data mining modules. The system has been designed to be extensible however. Imagery and textual data is handled differently due to the different processing requirements. This is therefore performed within separate streams within the server architecture.

For the second prototype, preliminary Instagram support was added, however this was at a time where Instagram were changing their policy of use, making it much more restrictive, and therefore would have made it difficult for proper testing of this module. As a work-around, an additional feature was added allowing the user to upload and analyse a batch of photos agnostic of any particular social network.

Textual Processing

This module has been integrated directly into the backend server architecture, providing analytics on text to determine their geographical source or context. More information on this module can be found in D5.1. This module was further refined and integrated into the privacy scoring framework.

This module was updated with regard to later prototypes as per deliverable D5.4. To allow for a larger degree of textual processing, which then forms a large part of the disclosure scoring framework. As an integration step, as the interfaces remained the same between deployment revisions, updating the component simply meant updating the module within the backend system.

Image Processing

The initial concept detection module was deployed on a separate server with specialised hardware in order to improve the performance, providing detection of a number of concepts from visual sources. Due to the hosting being on a different server, integration of this component is via a Message Queue system, to which the Image Processing module is subscribed. This was implemented within the first primary prototype, and more detailed information can be discovered within D5.2.

This module was further enhanced with a newer concept detection module, and an addition module was added to cater for logo detection. Due to the different processing pipelines and system requirements for both types of processing, the backend system runs a queueing system for both types of processing to ensure image information is kept up to date. A FIFO queue is established for each the image processing modules. The queue is then emptied by the processing servers, which processes 100 images in a batch for concept detection, but only 30 for the logo detection. On completion, data is pushed back to the backend system and recorded alongside each image. More detailed information on the updated processing modules can be found within D5.6.

In addition to the updated modules, the resultant output of the processing also plays a key role in the disclosure scoring framework.

GUI Web Interface

This provides the primary way for a user to interact with adding their social media profiles, and view results of the analytics. This is described in more detail in the later stages of this document, and within D7.5.

A number of additional elements were added to the GUI to take account of the privacy scoring framework, and also to return back results from image uploads.

Privacy Scoring Framework

This is a key element added within the second prototype, which informs the user regarding their privacy settings, and what may be exposed, catered to their own specific preferences. This module is integrated with many of the other elements within the USEMP backend framework. More detail on this module can be found in deliverables D6.4 – D6.6.

Integration requires interaction with multiple modules, this is provided by the central data store however, accessed via wrapper APIs. These APIs allow for retrieval of data specific to the logged in user such as likes, and like topics, information created from the processing modules, and data available direct from Facebook. Where associated information is missing, such as detailed information regarding a like, the backend system will attempt to fill from a local cache, failing this, a remote database lookup, or as a final step, from Facebook directly, at which point the local cache will be updated.

File Upload Integration

This is a social network agnostic integration element which allows data to be processed from any source provided the user is able to upload it manually themselves. This means that a user is able to process images they have stored on their own computer or any directly unsupported social network.

The methodology here allows for a user to uploaded a zipped package of images directly to the server for analysis, meaning the data does not have to be sourced from a social network, but has been made compatible with the common OSN export formats. The backend modules then run exactly as they would do for a direct OSN data fetch, albeit with only the information included in the package. This means items such as privacy settings and image comments may be reduced in comparison to a direct export. This data still is used, processed and fits in with the disclosure scoring framework as far as sensitivity is concerned, but then lacks the advice on changing privacy, for which such an action is not possible for user uploaded content.

2.2. System Deployment

The system is deployed as two complete configurations running side-by-side, with a live server available at ‘databait.hwcomms.com’ and a staging test environment running at ‘databaittest.hwcomms.com’. The staging area allows for new features to be tested and developed while not affecting running pilots with participants. This way it is ensured that new features meet the required stability and performance standards.

Due to the nature of some components, certain elements are shared between live and test platforms. This requirement is due to specialised hardware necessary for running imagery analytics. A high-level overview of the components comprising the deployed system can be seen in Figure 1.

The vast majority of backend services are not connectable to the internet; they are hidden behind both a firewall and a reverse proxy. The reverse proxy also acts as a Transport Layer Security (TLS) termination gateway, and enforces secure connections. This ensures the protection of the data of DataBait users that are stored in the USEMP servers. From this point, dependent upon the URL used, requests are either directed to the live or test system. Both systems have an equivalent server arrangement, the only difference being the version of the software that runs on the primary servers.

GUI components presented to the user reside on the presentation services server, from which requests can be made to the DataBait services server. The DataBait services server hosts the backend API matrix, which triggers and manages most actions relating to data analytics, data management and retrieval. This is the only server that has direct access to analytics results, and access to data stores. These data stores have no direct access to the internet, and all access to them is via the DataBait services server.

The system deployment remained the same across all prototype iterations, with individual components being updated, and expansion of capability.

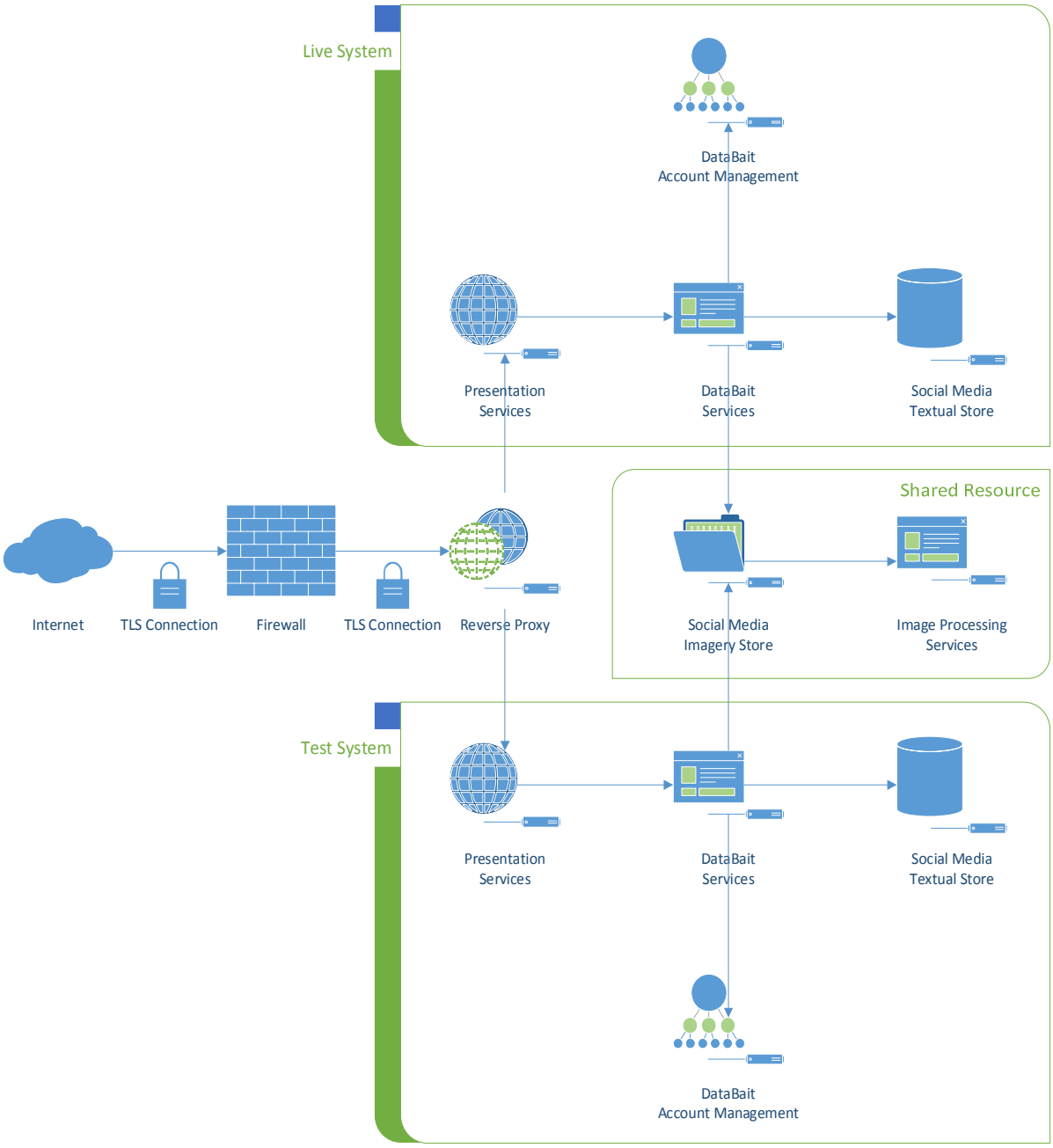

Figure 1: DataBait Deployment Diagram

The role of each server will now be defined.

Reverse Proxy

The reverse proxy acts as the TLS termination point, and acts as a data cache to reduce processing requirements on backend systems for regular queries. It also handles balancing of connections to backend systems and directs requests to the live or test deployments.

During large scale tests, it was found more processing capacity was required, and therefore this also acted as a load balancer for processing requests in order to pass requests to the least busy backend server. The need for load balancing was more dependent upon network bandwidth, rather than processing capability. The load balancer therefore performed better distributing network traffic across multiple nodes – particularly with fetching of image data which with multiple users could saturate the network bandwidth. Distribution allowed for

QOS rules to be more effective, saving bandwidth for API requests, which were otherwise starved due to image fetching.

Presentation Services Server

The presentation services server hosts components related to the user's browser, i.e. the visual components of DataBait which reformat and appropriately present analytics results prior to releasing them to a user. Through user interaction, the presentation services server also makes requests to the DataBait services server to initiate account creation and execution of data analytics processes.

It is this server that hosts components defined within deliverables D7.2 and D7.5.

DataBait Services

The DataBait services server manages all backend requests, including but not limited to, account creation, fetching social media data, triggering analytics processes, and formatting results in a format suitable for the presentation services components.

During a large influx of new users, this element was frequently used, and therefore additional servers were spooled up to cope with demand, load balanced dynamically, as described earlier with regard to the reverse proxy.

DataBait Account Management

This server handles the DataBait account. This includes managing social media access tokens, account linking for provisioning of multiple networks, and keeping track of prior logins and information required to ensure data is kept as up to date as possible, and that data is properly attributed to a user should a data withdrawal request be made.

Social Media Textual Store

This store acts as a storage area for all textual social network data, and stores the results of any social media analytics.

Social Media Imagery Store

The imagery data store is segregated from the textual data store due to the nature of the analytics and the larger amount of storage space required. The increased retrieval time of images also makes it beneficial for this data store to be shared between the live and the test environments. Results from image analytics are stored alongside the imagery data.

Image Processing Services

This server hosts the hardware components necessary for running imagery processing services. As the hardware is specialized and expensive, only one server has the component installed and therefore this is shared between the live and test environments.

2.3. DataBait Account Schema

This section describes the schema for the DataBait account, which stores information required for users to login to their DataBait profile, and manages linking between their social networks. This section will describe the role of each table within the DataBait database, and the type of data contained within, as well as its purpose, and how this relates to the DataBait system as a whole.

See Figure 2 for a diagrammatical view of the schema.

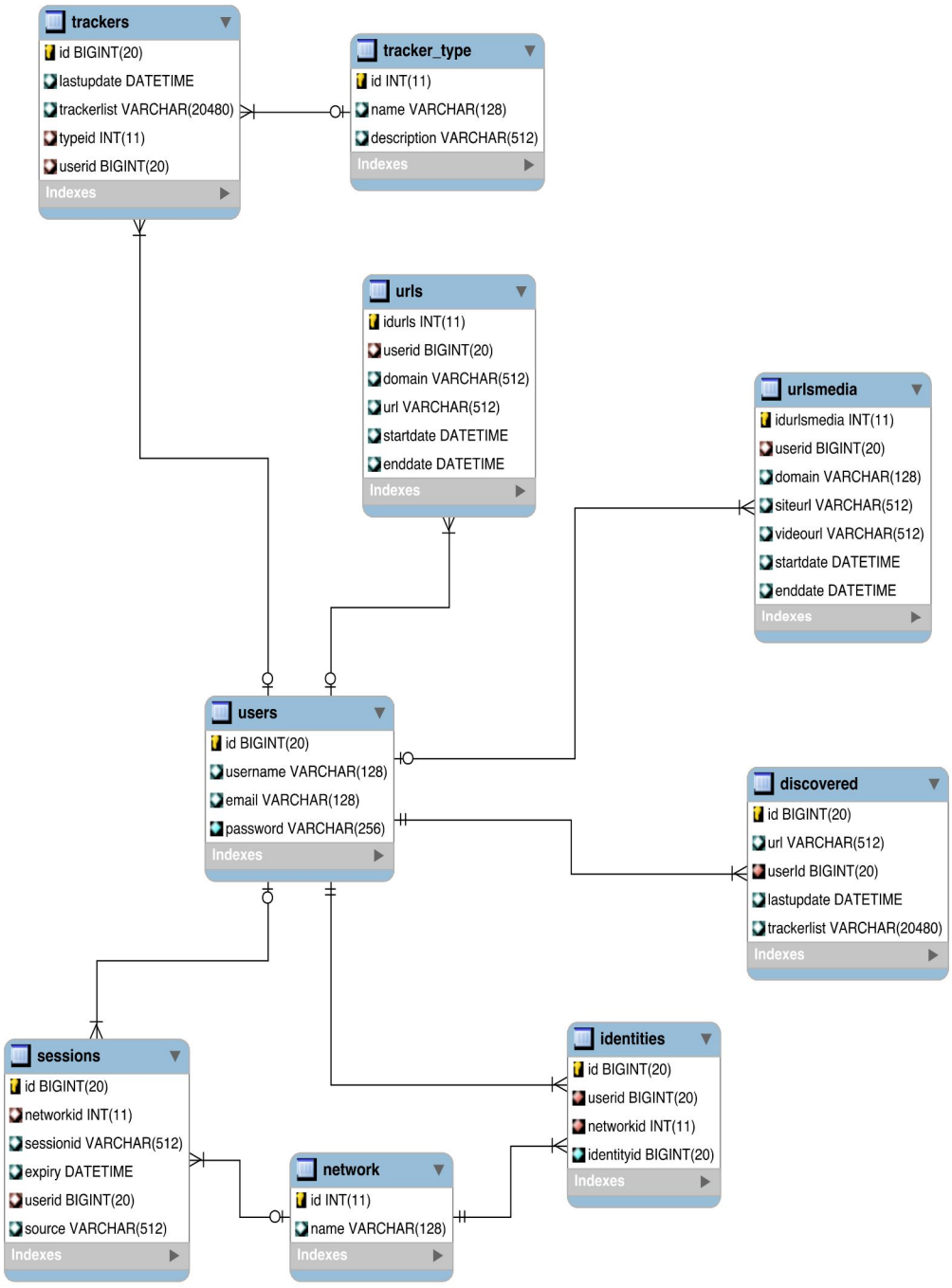

Figure 2: DataBait Database Schema

Users

This is the primary table storing a user's DataBait identity credentials and ID. The password is hashed using a random salt prior to being stored.

Identities

This table manages identities across multiple social networks or services. This way a user can have a single DataBait identity, but can have multiple social network accounts linked to it.

Network

Contains a list of pre-defined networks linked to the 'networkid' field within the identities and sessions tables.

The contents of the table are as follows:

id	name
1	LOCAL
2	FACEBOOK
3	TWITTER
4	LINKEDIN
5	LIMESURVEY

Table 1: DataBait Network Table

Whereby the DataBait system has been designed to handle multiple social networks from construction.

As this table is also used for session management, 'LOCAL' network is used for sessions management within the DataBait system. The others are for handling account linking and session management across the associated social networks, albeit with only Facebook being implemented within the Pre-Pilot phase. The 'LIMESURVEY' network is for account management with the LimeSurvey system which was used during the Pre-Pre-Pilot phase to gather user views. This was used in order to generate user accounts within the LimeSurvey system, and allow for user interaction directly from the DataBait system.

Sessions

The sessions table is for management of sessions and access keys for the associated social network / service. In the case of Facebook, this is where a long-lived token is stored with the expiry date, that is a token that lasts for a period of months rather than hours. This allows the system to keep users' privacy data up to date. For the case of Lime Survey, it contains the account generated token. For DataBait itself, it manages a generated universally unique identifier (UUID) based token, which is used for all user interaction with the backend, once a user has logged in.

Trackers

The trackers table contains management data for the DataBait plugin. It keeps track of web tracker whitelists and blacklists.

Tracker Type

Records the type of tracker data to be recorded within the Trackers table.

id	name	description
1	WHITELIST	NULL
2	BLACKLIST	NULL

Table 2: Tracker Type Table

Discovered

Keeps track of trackers present on a given website, such that a user can view their online trace.

URLs

Collates data for a user browsing profile when the DataBait plugin is installed.

URLs Media

Collates data for a user browsing profile when the DataBait plugin is installed, but specifically targeted at media resources i.e. video.

2.4. Information Flow

This section describes the backend sequence of events when an account is created, the flow is similar to that described briefly from the GUI context in section **Erreur ! Source du renvoi introuvable.**, and a more detailed account of the GUI workflow can be seen in Deliverable 7.5.

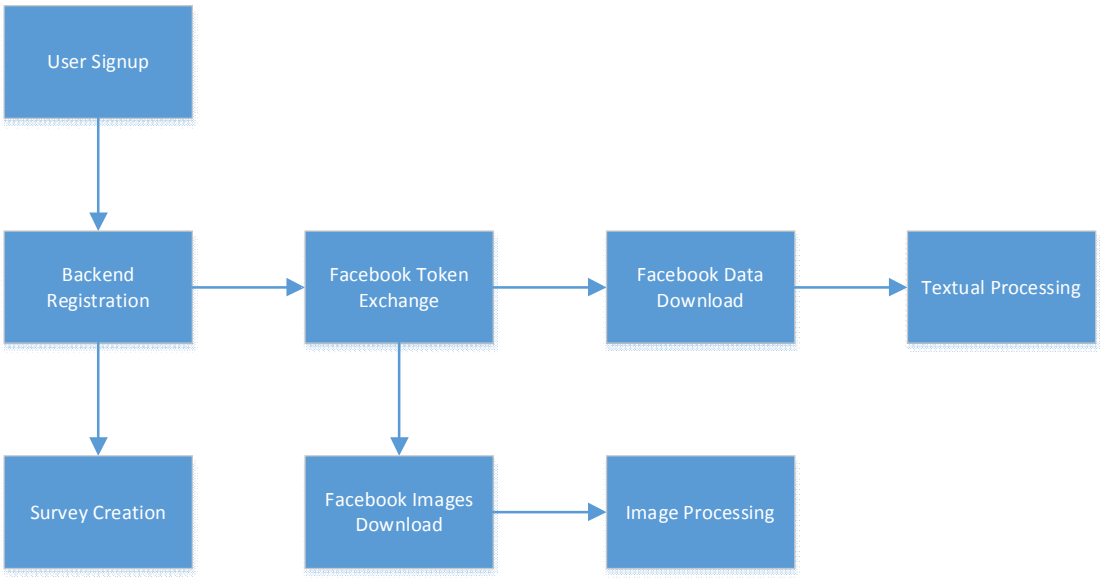

Figure 3: DataBait System Workflow

The collation of data is first initiated when a user makes an account on the system. When user registration has completed, this sets off a number of processes. The first of which is exchanging a short-lived Facebook token for a long-lived token, allowing the backend systems to continue retrieving data from Facebook while the user is logged out of the service.

Simultaneously, a survey profile is created for the user, such that a number of probing questions can be asked relating to how a particular person perceives and feels regarding their privacy (more information and results on this can be found in deliverable 4.2, and is considered out of scope herein).

On retrieving the Facebook long-lived token, data fetching and processing is handled in two distinct streams independently. That is, one stream for the textual data and analytics, and one for imagery data.

Textual data is processed in-situ as the information flows into the system, the data is analysed as it passed through the system, and is then stored in backend systems with the associated analytics output.

Imagery data is downloaded and placed on a separate imagery analytics server. This process requires all images be first downloaded, and are then processed in a batch after download completion. Analytics information is then stored alongside the images.

This system can handle multiple users signing up at the same time, and is multithreaded as to not affect the running of the primary systems. Should a very large number of users sign up simultaneously, to maintain performance, jobs are queued to ensure overall system performance.

3.Integrated Components

This section covers additional components currently deployed within the DataBait systems, and their current state of operation.

3.1. Image Processing

In the first prototype, the only image processing module was concept detection, which was later updated with an improved module, along with the addition of a module capable of logo detection.

Processing is handled by two separate queues, from which both processing chain pull in the latest batch of images to be processed respectively. If the system requires expansion in the future, this queueing system means that additional processing servers can be added later to cope with additional demand simply by spooling up more servers. For the pilots however, the processing capability of the server was more than enough to handle the number of requests.

Each processing system will now be discussed.

Concept Detection

The image processing modules were integrated as per D5.2 for the earlier prototypes, and updated in line with the latter D5.5. It is these modules that provide the “Photo Insights” and “Brand Insights” functionality, more on which will be shown later in regard to the user interface. This module supports the detection of over 17,000 concepts that are part of the ImageNet dataset (Deng et al., 2009[1]). To keep abreast with developments in computer vision, concept recognition is done with convolutional neural network features. Independent learning was implemented for individual concepts in order to enable the easy extension of the concept set. This characteristic is important in USEMP to easily cope with the potential evolution of DataBait functional needs between the different user evaluation stages.

This forms part of the batch processing for imagery analytics, running on dedicated hardware. Output is exported directly to the backend systems, attributed to each individual image with the detected concept tags, in addition, another server view is calculated listing images by concept.

In addition to concept detection, for the second prototype, logo detection was an added feature, based on the same backend processing framework.

Logo Detection

Logo detection was added in order to detect common brands within a users’ images to determine brand affiliation. It is capable of detection of over 1400 different logos covering a number of products and brand affiliations. More detailed information on the module can be found within D5.5 and D5.6. As with the concept detection it forms a part of the information provided to the disclosure scoring framework. Similarly, with the concept detection, processing takes place on the same hardware.

---

[1] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, L. Fei-Fei. (2009). ImageNet: A Large-Scale Hierarchical Image Database. Proceedings of CVPR 2009.

3.2. Textual Location Detection

The textual location detection module takes as input the textual posts of the DataBait users and for each of them it estimates the most likely location associated with it. The estimation is carried out using a probabilistic model that is learned from a large collection of geotagged content collected from social media. The model splits the globe in a regular grid and each piece of content is assigned to some cell of the grid according to its geotags. The content assigned to each cell is then used to learn a probabilistic language model for the cell. Estimation for some new piece of content is carried out by utilizing the set of probabilistic language models in order to find the most likely cell that is associated with it. The location of the cell in the globe is then translated from a latitude / longitude pair to a geographical name in order to be shown to the user. It should be noted that no estimates are produced either when the confidence of the model is below some empirically defined threshold or when the text of the post is too generic and can be associated with many different locations (this is typically the case when the post does not contain any elements that are strongly related to some location, a case that is quite common). For more details of the approach that is used for estimating the location associated with a post please see D5.1 and D5.4.

This module is integrated directly into the backend systems, enabling processing of textual content to be processed in-situ as soon as the data has been made available by the social network.

3.3. Disclosure Scoring Framework

The disclosure scoring framework (DSF) was a key element added within the final prototype. The DSF takes in all information gleaned from the processing modules as a whole, with the addition of user likes, and provides output based on a user's preferences as to how a user's privacy setting should be set for a particular image, alerting those scored inconsistent will how a user chooses to be viewed via a user's disclosure factors.

Processing takes place in situ within the primary backend servers, processing and appending data as the associated modules return their output. Scorings are recalculated on the fly when a user alters their preferences through the interface (shown later).

Further information on the workings of this module can be found within D6.4 – D6.6, and an abridged description of the interface provided in Section 4.5.

3.4. Web Trackers Plugin

The DataBait Browser Plugin allows users to block 3rd party trackers in websites they are visiting. Users can login to their DataBait account from there and see a history of visited sites and list of trackers, see the ones already blocked or even block them through this page. Both the plugin and the site share the same user credentials.

The web trackers plugin works externally to the DataBait system, and interacts with the backend systems via API calls, to store any information gathered. It is provided for the chrome browser.

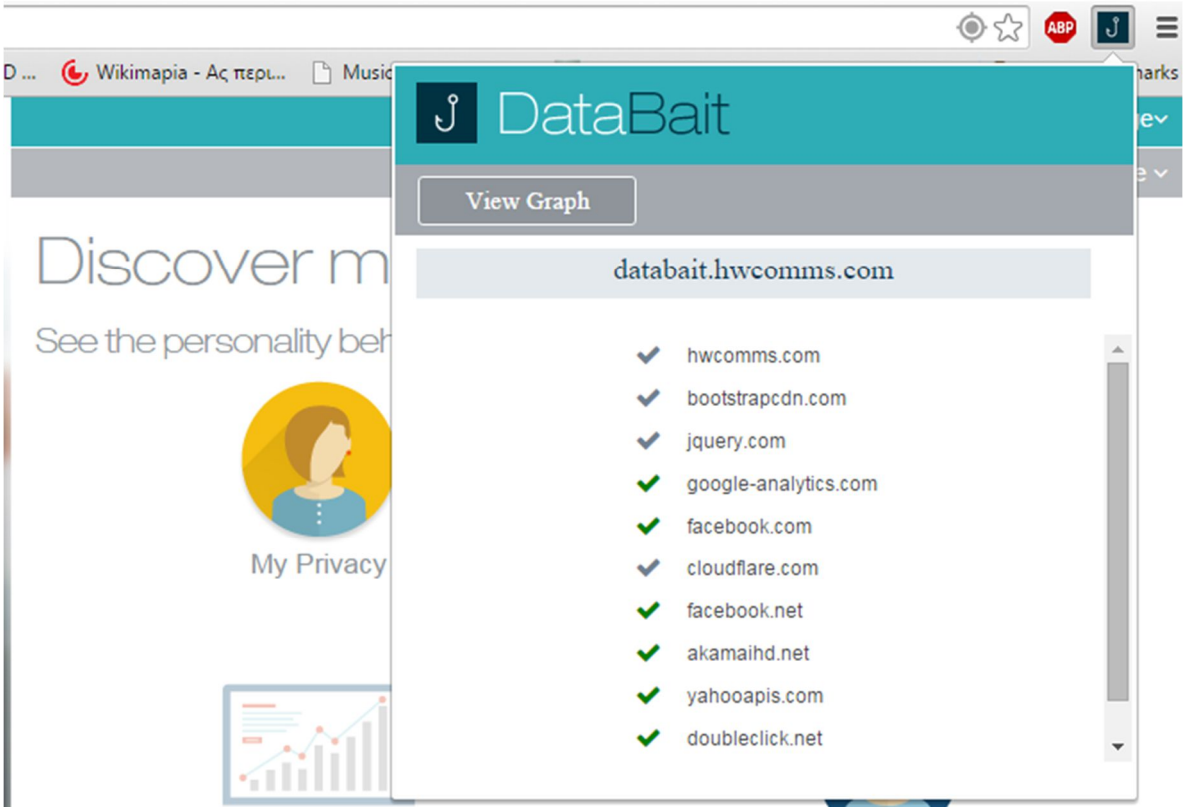

Figure 4: Web Trackers Plugin

4. User Interface

This section covers the GUI elements launched with the final system. This section will provide a brief overview of features, as this information is covered more prominently within deliverable D7.5.

4.1. Landing Page

On first opening a browser to one of the DataBait staging servers, the user first encounters the landing page, which enables them to either login, or to make an account. The registration process ensures privacy concerns are taken into account, and that the user is of an appropriate age. On successful creation of an account, or after login, the user is taken to the home page.

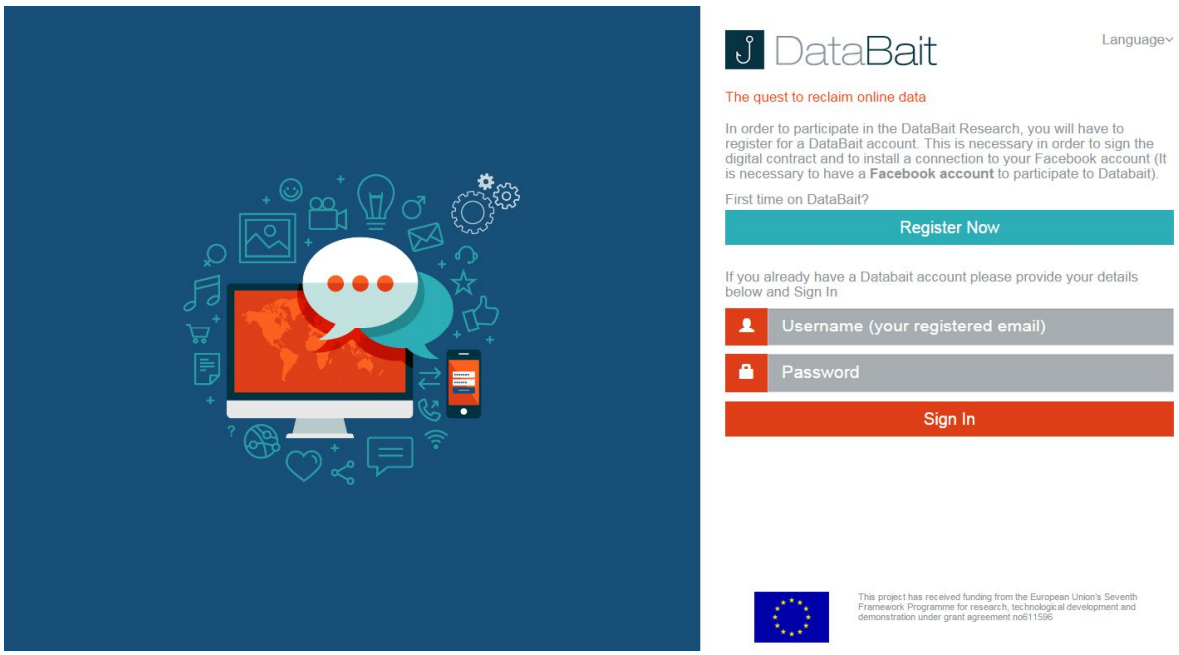

Figure 5: DataBait Landing Page

4.2. Home Page

The home page allows a user to follow through to explore what the DataBait analytics components have discovered about the user. For the system, all analytics components that were presented in the previous section were active. The web tracker blocking system was also active should a user have installed the necessary browser plugin. Web trackers is predominantly a feature of the plugin system, with the associated data being stored in the backend for storage and retrieval to be synced across all devices.

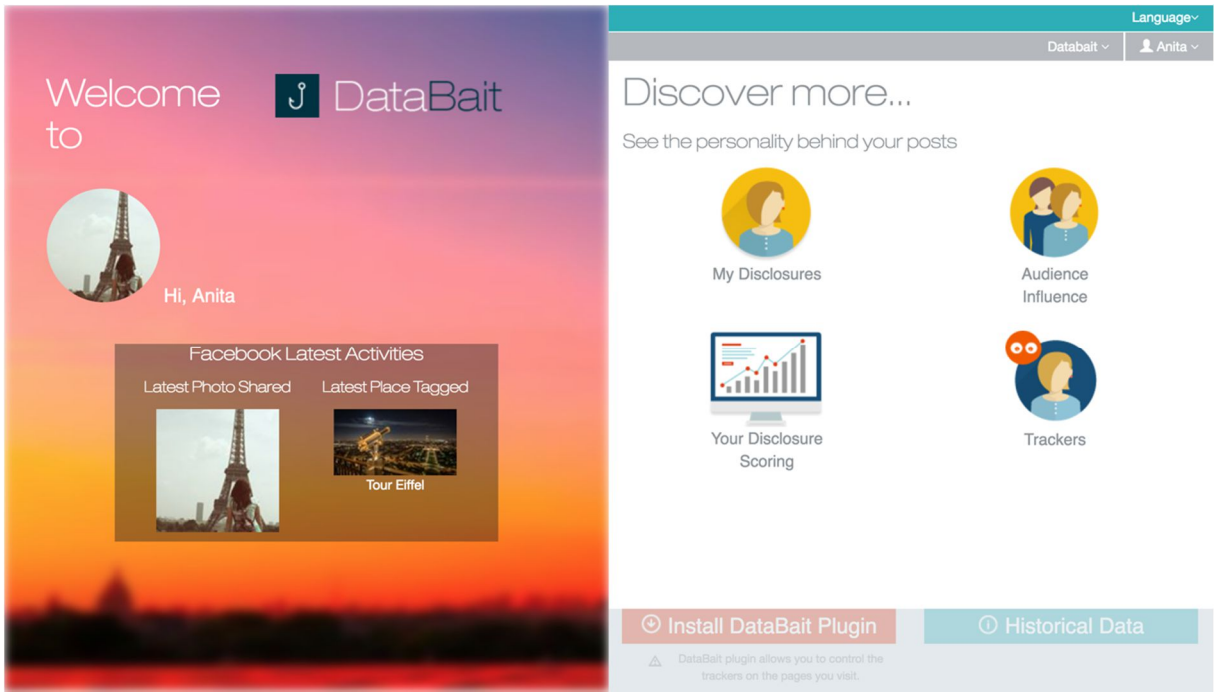

Figure 6: DataBait Home Page

4.3. My Disclosures

The 'my disclosures' section relays the results of image and location detection back to the user, through three primary elements, 'Photo Insights' relaying the output of the imagery concept detection module, and 'Location Insights', relaying the output from the textual analytics module, and final 'Brand Insights' relaying information determined from imagery regarding brand affiliation.

Photo Insights

Image leaks displays a clickable word cloud of the most dominant concepts found amongst all the user's images. Clicking on a concept, reveals the associated images tagged with that concept. The most dominant concepts are listed first, down to the least common concepts.

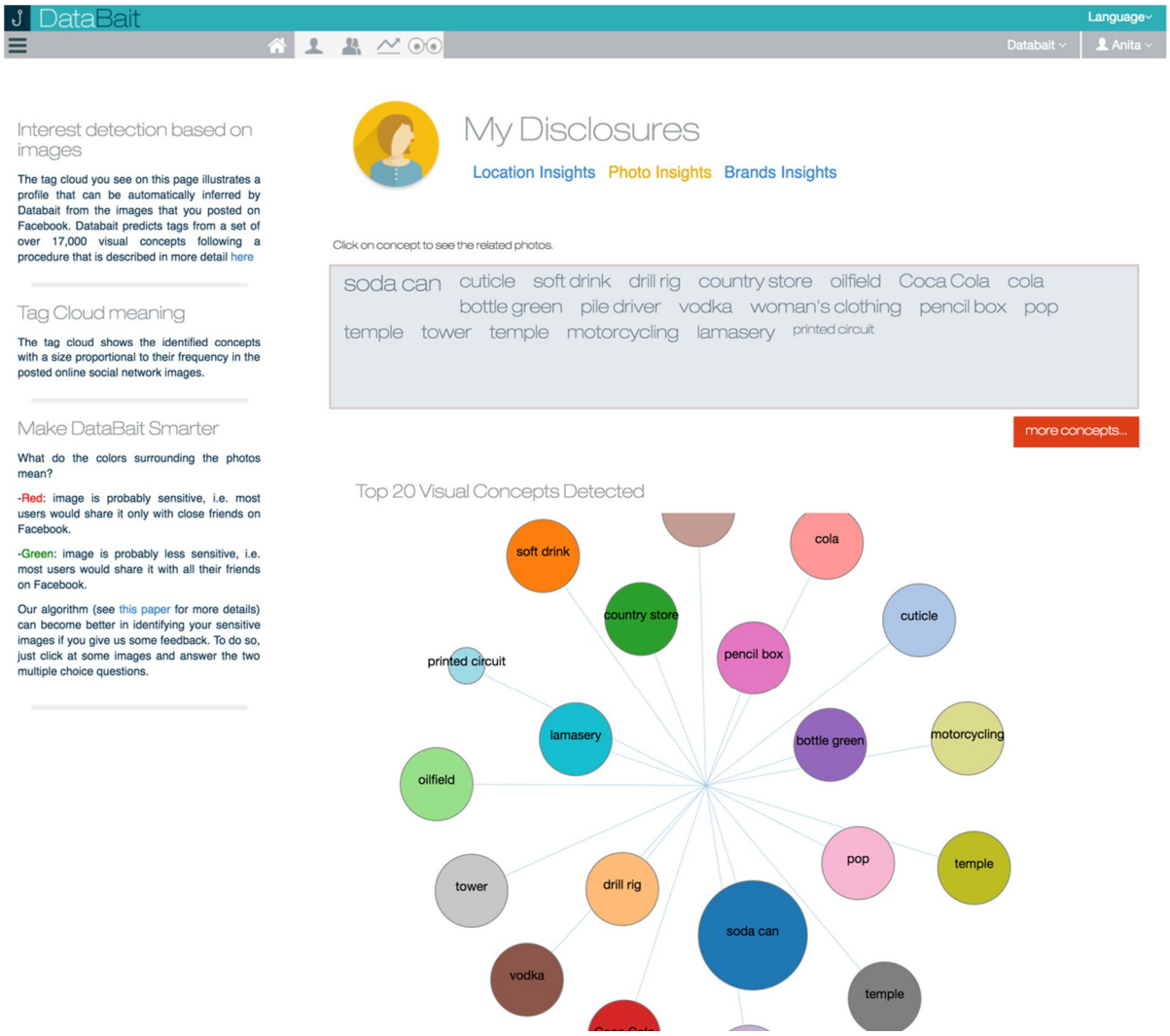

Figure 7: DataBait Photo Insights

With the addition of the private/non-private image classification module (detailed in section 4 of D5.6), on selecting a topic, photos are then coloured with regard to likelihood of containing sensitive content.

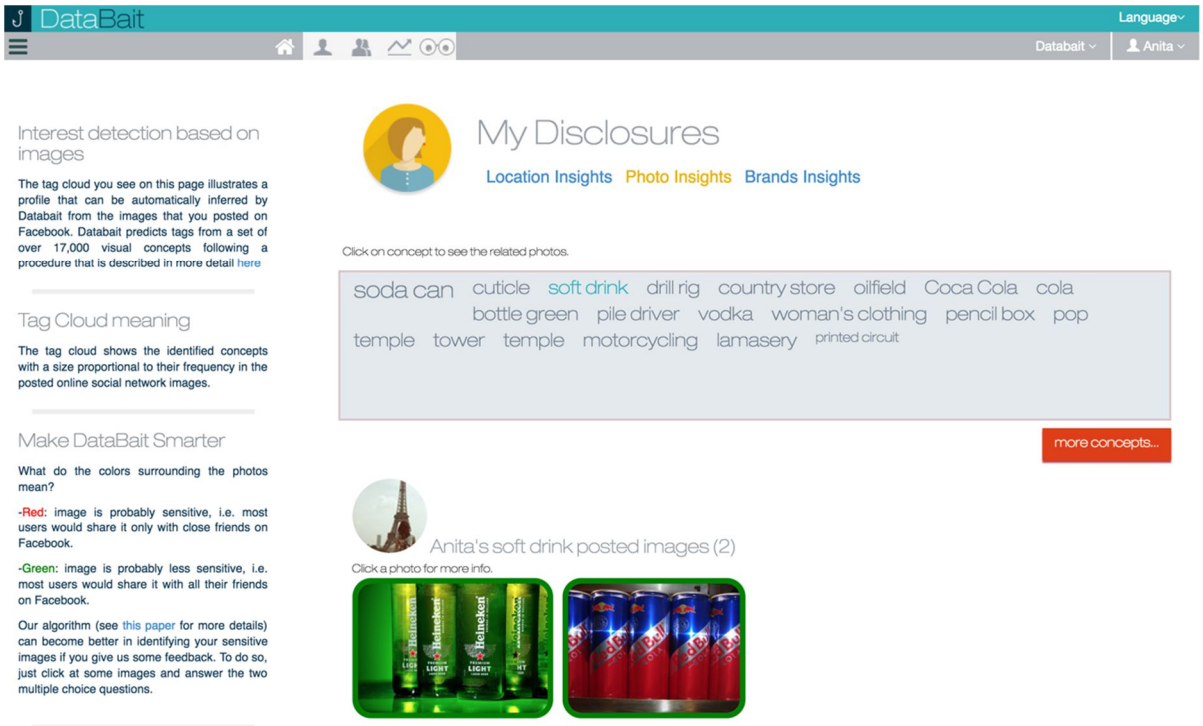

Figure 8: DataBait Photo Privacy Prediction

The user in encouraged to provide privacy-related feedback whenever a prediction is incorrect, which will assist the backend system to better predict privacy in the future.

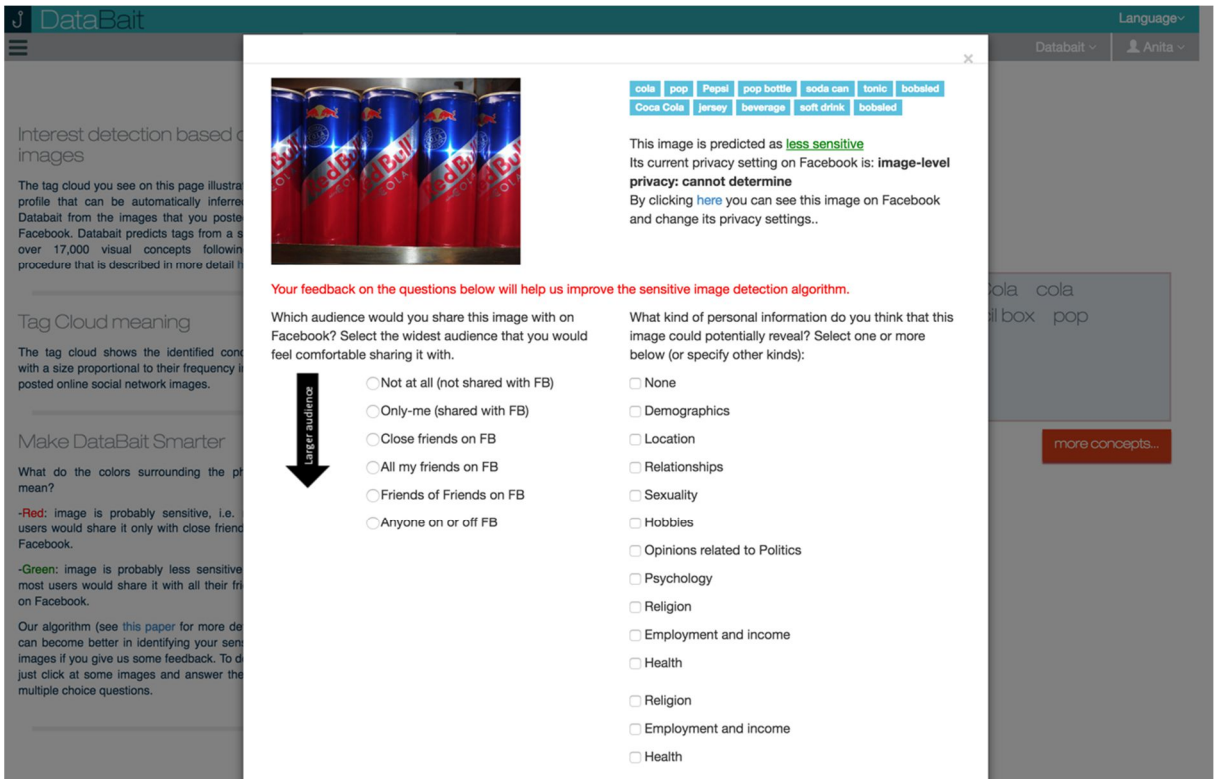

Figure 9: DataBait Photo Privacy Feeback

Location Insights

Location insights presents all the locations detected from textual analytics, determining a likely location from posted content. In addition to this, the system reports explicitly defined locations, i.e. where user tag themselves at a specific location. This again uses the word cloud model, albeit tagged locations appear in yellow, while locations from analytics are in grey. All locations are then placed on a traversable map.

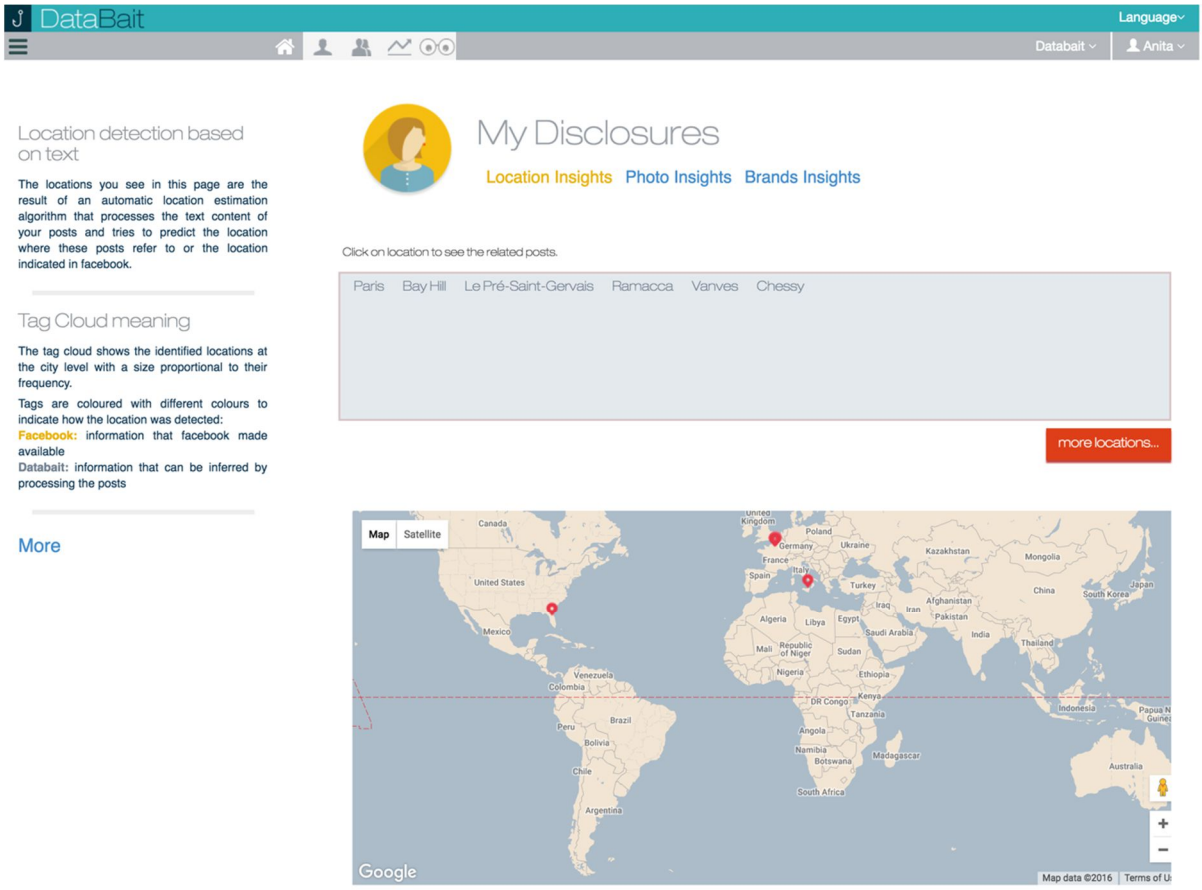

Figure 10: DataBait Location Insights

Brand Insights

Similar to the photo insights, this provides similar feedback with regard to brands.

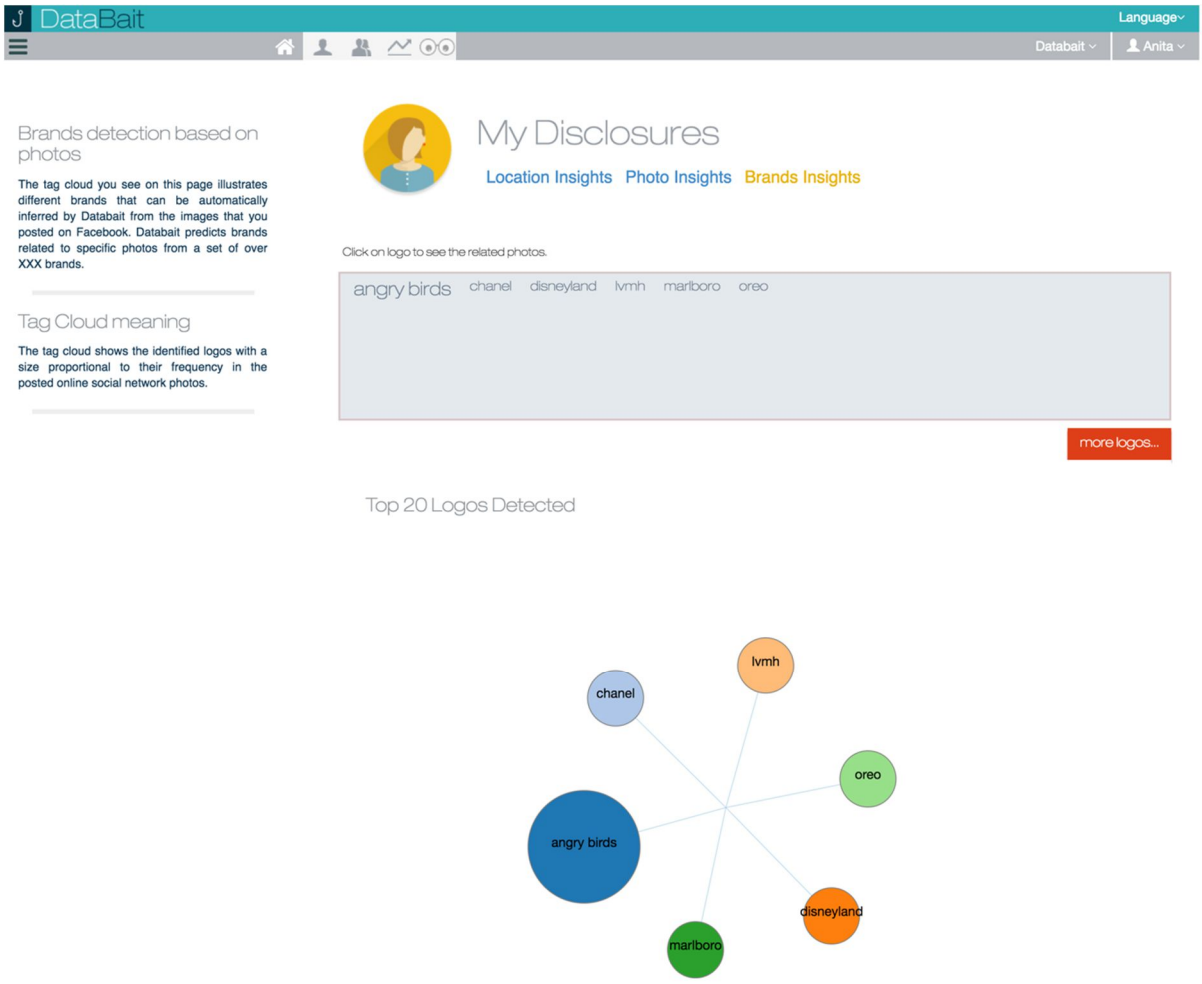

Figure 10: DataBait Brand Insights

With picture perusal on clicking on the desired brand.

Figure 11 DataBait Brand Identification

4.4. Audience Influence

Audience influence provides the user with who they are most influential, i.e. which users like their posts the most, with which users they have the most interactions with, and in general provides a user useful feedback on who is interacting back through their own profile.

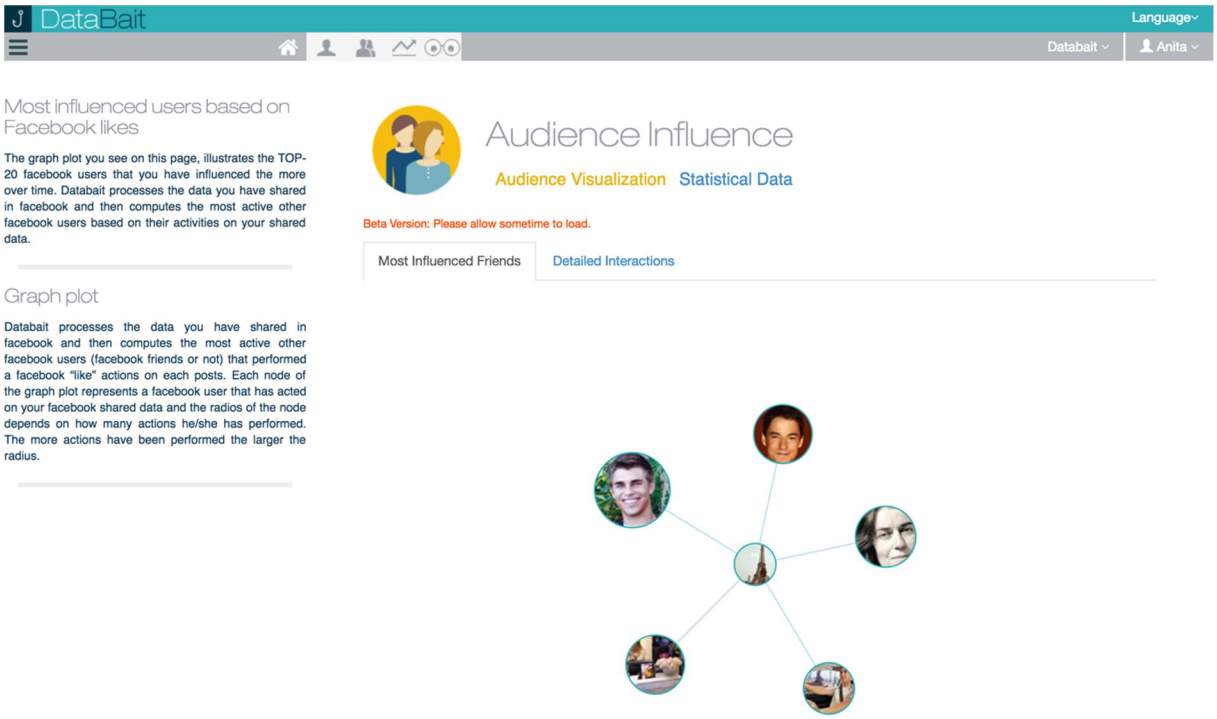

Figure 123: DataBait Audience Influence

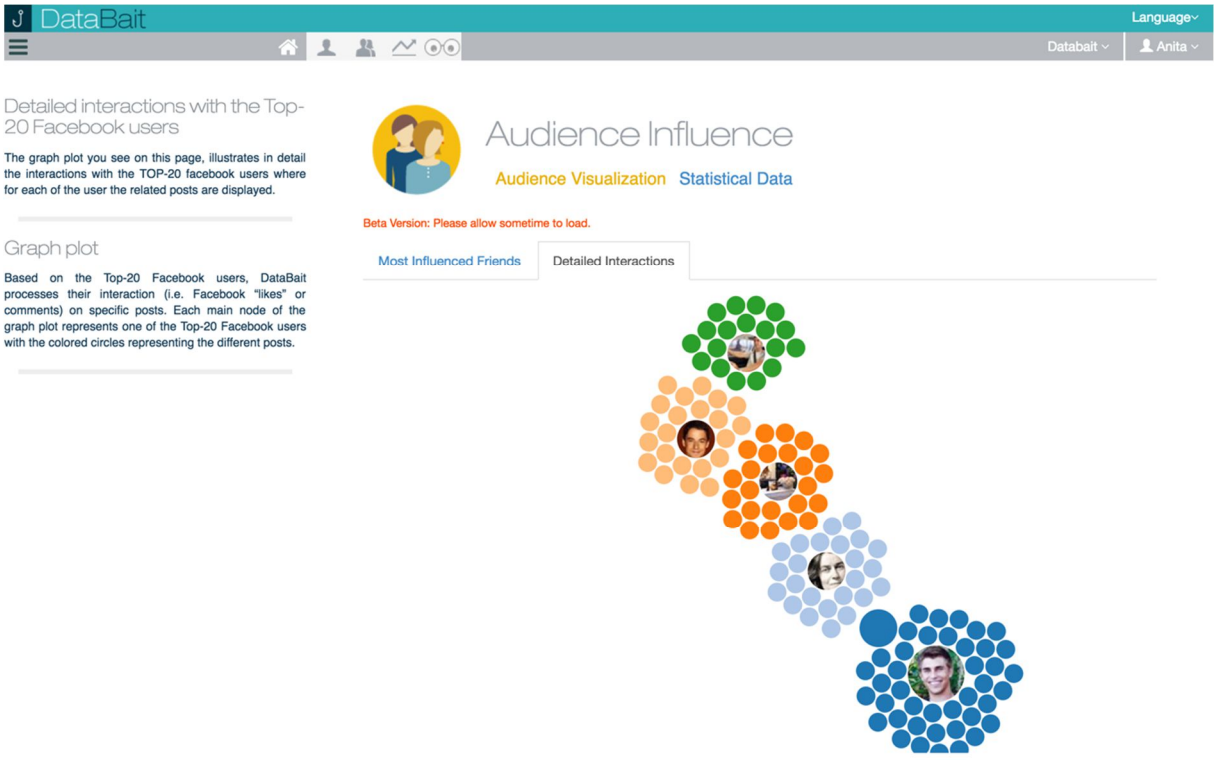

Figure 134: DataBait Audience Interaction

4.5. Your Disclosure Scoring

The disclosure scoring element informs the user what can be inferred about themselves which has not been explicitly stated on their profile using the previously described analytics modules, and user likes. This is returned via a number of factors which are deemed most prominent. On clicking on a particular factor, a user can then drill down to see why a specific factor was flagged, and what data was used for that inference.

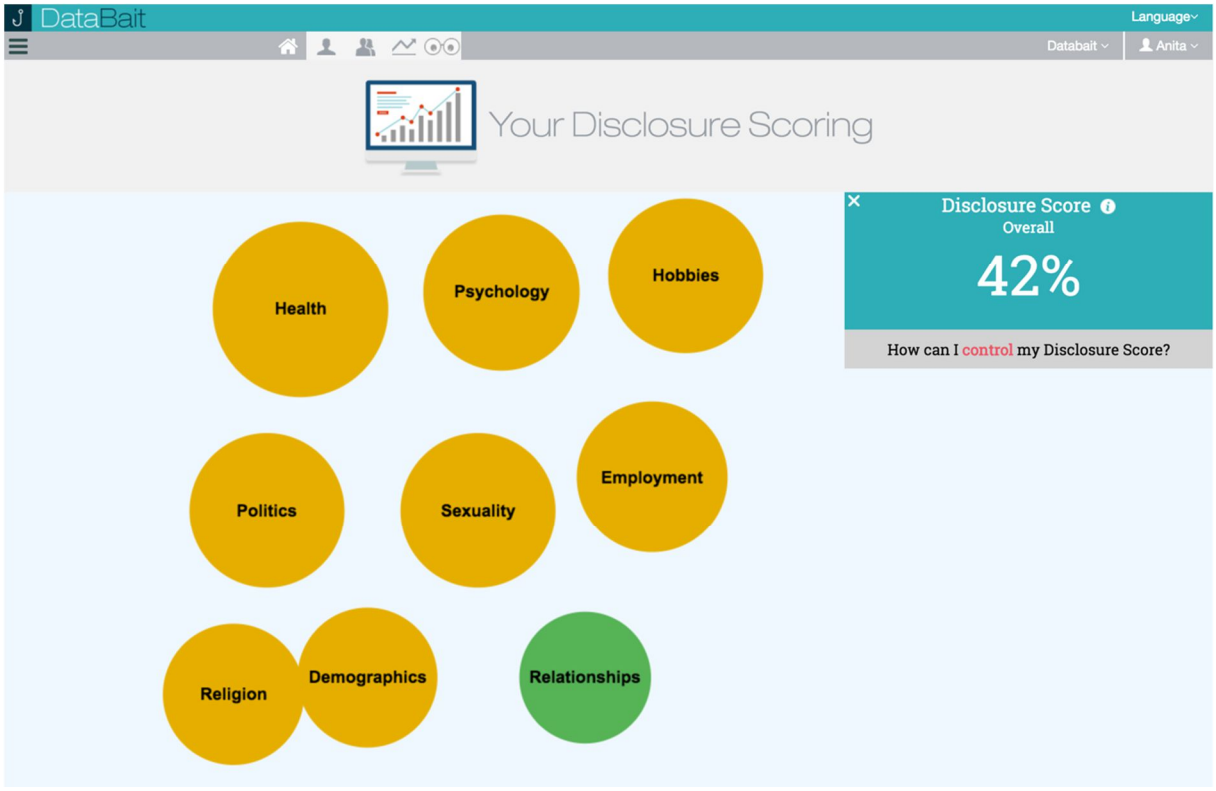

Figure 145: DataBait Disclosure Factors

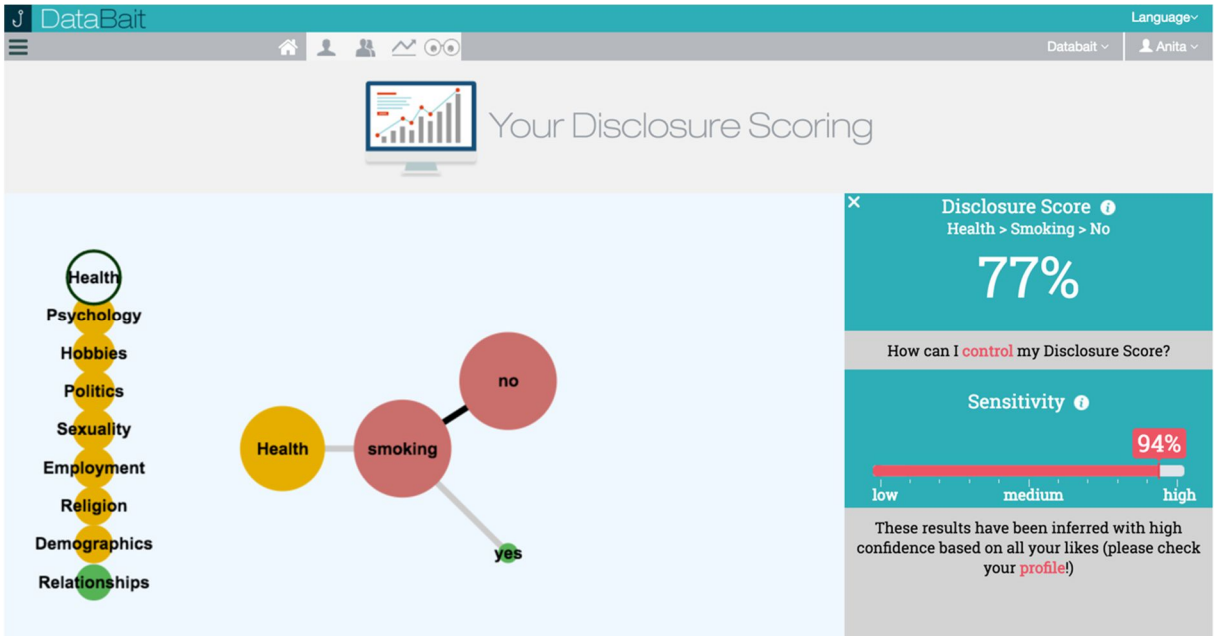

Figure 15: DataBait Disclosure Inference

4.6. User Trackers

The User Trackers feature works with the DataBait browser plugin, detecting which trackers are present on a range of websites, allowing the user to block them at will. The tracker page allows a user to view and block the associated trackers, this information is then passed back to the plugin, such that any changes take effect immediately.

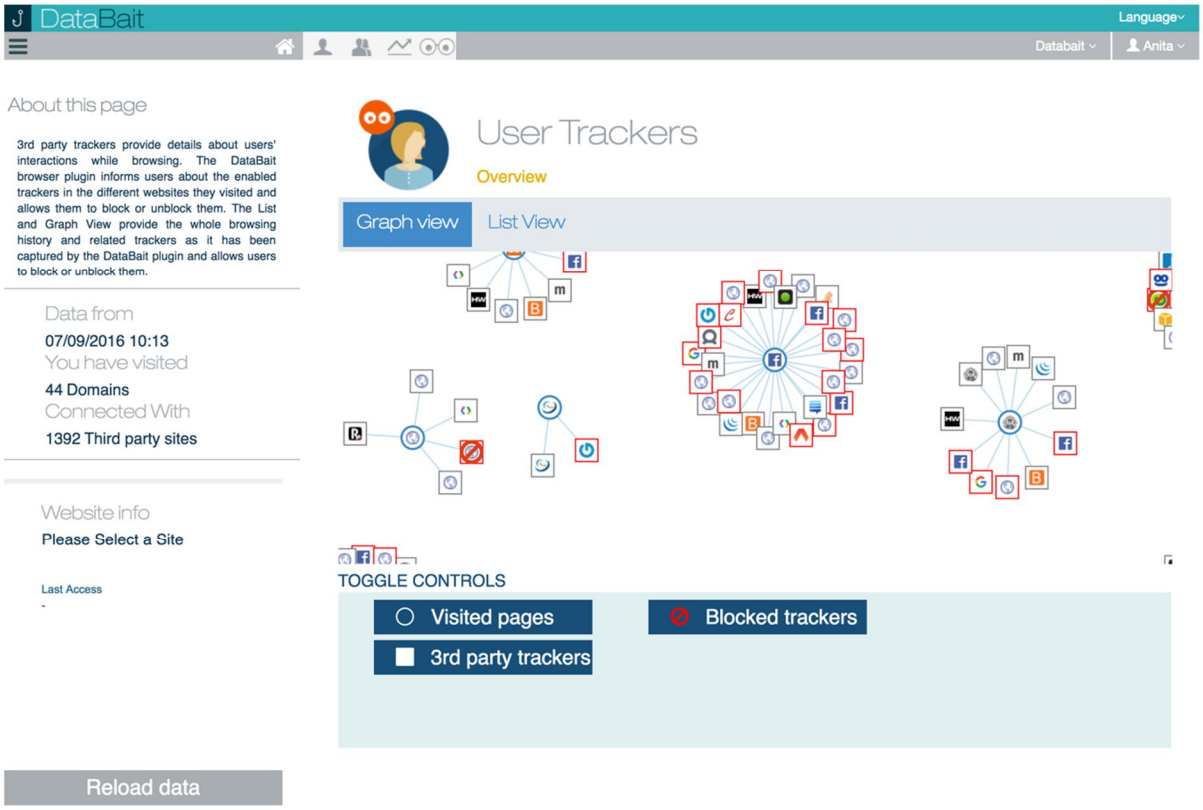

Figure 17: DataBait User Trackers

5. Conclusion

As described in the USEMP DoW, a fully operational version of the system was developed and tested within the project. To facilitate the testing and integration of new features without impeding on the use of the live system, a test system was specifically created.

The preceding sections provided a detailed description of the current state of the system that shows that most of the components developed throughout the projects were effectively integrated and exposed to the final users. Results obtained with the technical components developed in WPs 5 and 6 are either directly presented to users, as it is the case for the visual concept detection, or integrated to higher level functionalities, such as the disclosure scoring which includes a number of text and image processing components.

The results obtained during the pilots that were run as part of WP8 indicate that a wide majority of DataBait functionalities are considered to be useful by the final users. One important improvement point is related to the processing speed, which is currently not optimal for a part of the components. However, while very important in practice, this optimization goes beyond the immediate scope and means allocated to the project. Dedicated material and human resources that were not available are needed in order to deploy DataBait in parallel on several machines and to speed up processing.

Finally, as anticipated in the DoW, the integration with Facebook was not straightforward. This is mainly explained by the fact that USEMP goes against some of Facebook's current practices. The consortium's first application to release DataBait to the general public was not accepted. The problem signaled by Facebook was immediately corrected and a resubmission was done which is still pending.