SEVENTH FRAMEWORK PROGRAMME

Networked Media

Specific Targeted Research Project

SMART

(FP7-287583)

Search engine for MultimediA environment generated contenT

D6.1 Integrated Applications of the SMART Multimedia Search Engine

Due date of deliverable: 30-04-2013

Actual submission date: 02-07-2013

Start date of project: 01-11-2011

Duration: 36 months

Summary of the document

Code:	**D6.1 Integrated Applications of the SMART Multimedia Search Engine**
Last modification: **State:**	02/07/2013 Final
Participant Partner(s): **Author(s):** **Fragment:**	S3Log, Prisa, Glasgow, Telesto, AIT Massimiliano Tarquini, Dyaa Albakour, Thanos Alexiou, John Soldatos No
Audience:	☒ public ☐ restricted ☐ internal
Abstract:	This deliverable will be a report and an accompanying prototype implementation of the SMART proof-of-concept use cases. This report is intended to describe proof of concepts functionalities, anticipate critical aspects of integration, identify and fix technical risks, test and reporting about the flow of information.
Keywords:	Proof-of-concept, Use cases
References:	See within document for references.

Table of Contents

1. Executive Summary

1.1 Scope

WP6 is devoted to the integration of the SMART open source multimedia search engine, as well as to its subsequent validation in the scope of prototype applications (use cases). Hence, the main targets for this WP include: (A) Integrating results from other Work Packages to produce a unique open source platform and (B) Validating SMART functionalities by implementing real use cases;.

To achieve those goals, SMART Project internal stakeholders will act as external ones getting all the SMART search engine components and trying to integrate them in unique platform through the realization of the prof of concept.

This deliverable will be a report and an accompanying prototype implementation of the SMART proof-of-concept use cases. This report is intended to anticipate critical aspects of integration, identify and fix technical risks, test and reporting about the flow of information. As part of the deliverable we present the live news and security/surveillance use cases, along with the use of the SMART platform for social networks search.

1.2 Summary

This document describes SMART search engine, its components and services have been integrated to realize realistic use cases. The document complements the proof-of-concept implementation of the use cases. The document will be delivered in three version corresponding to the three iterations (M18, M27, M36). Each iteration is intended to incorporate within the POCs features as they outcomes from the SMART research including the most recent releases of the integrated SMART platform.

1.3 Audience

This document is mainly addressed towards:

- **SMART project members**: The document will provide information about the status of the art regarding SMART PoC's. Also it provides information about future plans for the Proof of Concepts. Later versions of the document will provide information about the progress of activities, with respect to the plans described in this first version.
- **SMART software developers**: In its final version, this document will represent an important technical reference for understanding the design of the PoC's, they're components, the communication protocols.
- **Open Source Community**: We hope that the PoC represent an incentive for the open source community to adopt SMART and continue in the development of solutions based on it. The proof of concept should be a stimulus to creating new components starting from a solid bases represented by the PoCs.

1.4 Glossary of terms and abbreviations

- **HMI**: Human Machine interface
- **JSON**: (JavaScript Object Notation) is a lightweight data-interchange format;
- **WS**: Web Service;
- **WMS**: Web Map Service Interface Standard. Provides a simple HTTP interface for requesting geo-registered map images from one or more distributed geospatial databases.

- **WFS**: Web Feature Service. Simple protocol that enables the exchange of geographical components through the web;
- **GML**: Geographical Markup Language;
- **CSS3**: the Cascading Style Sheet language is used to format graphical components in a HTML page;

1.5 Structure

The deliverable is structured as follows:

- In Section 2, we introduce the proof-of-concept spanning on released functionalities.;
- Section 3 illustrates the Security and Surveillance use case and its implementation using the SMART open source search framework.
- Section 4 illustrates the Live News use case and its implementation using the SMART open source search framework.
-
- Section 5 illustrates several Social Search use cases.
- Section 6 identifies future steps for all the use cases.
- Section 7 is dedicated to conclusions and results.

2. Overview of the Proof of Concepts (First Release)

In this section, we will describe three Proof of Concept (PoC) applications intended to demonstrate whether the SMART platform satisfies aspects it was designed for by a system integrator point of view.

Activities are intended to:

- Validate the open source search engine in the scope of use cases and queries, mixing sensor networks and social network;
- Demonstrate how live events and data are ingested by the framework and how information generated can be aggregated and used in different application domains.
- Produce integrated applications (use cases) based on the SMART open source search framework [D6.2].

First release of the PoC applications will be based upon components from:

- WP3: Sensor and Multimedia Signals Processing
- WP4: Edge Node's feed storage and streaming. In addition to the Social Network Manager
- WP5: SMART Search Layer

In the remainder of the section, we give an overview of our PoC applications, namely Security and Surveillance, Live News and the Social Search use cases.

2.1 Security and Surveillance

Speaking of urban security, decision support systems are becoming increasingly important. This security PoC addresses the problem of how to provide, in a simple and immediate way, information relating to the events that could be classified of interest for the monitoring and management of situations of potential danger.

From the point of view of safety, all events related to a sudden change on environmental condition can be relevant. Such events should be combined (correlated) with contextual information to better define the context. For example, in case of mass events (domain), an immediate change of the direction of motion of the crowd (event) may identify a situation of possible danger.

The ability of SMART to merge information from social networks (used to identify a context) with information from sensor networks (used to identify and classify the fault), along with the ability to retrieve data through the search engine, makes SMART particularly suitable to be used as a system to support decisions in all cases of urban security. The purpose of the SMART security and surveillance applications is provide awareness about key events in the surrounding environment, including identification of events in real-time. The application integrates SMART search functionalities with tools and techniques for managing and visualizing alerts, threat and other security events, including a wide range of HMI (Human Machine Interface) components.

In particular, the first release of the PoC includes components that enable the following functionalities:

- Navigate web sites
- Navigate social network data
- Browsing multimedia datasets
- Integrating external data from physical sensors and social network.
- Rendering and manipulating maps

- Rendering and manipulating multimedia contents (video, images, etc. etc.)
- Rendering graphs in different formats
- Tools to interact with the elements shown on the map.
- Tools to manage alarms and threat situations.
- Tools to query stored data
- Tools to monitor and configure the system status

SMARTCOP components are connected to the SMART Architecture through data connectors. In Section 3, we discuss in detail the architecture and the implementation of the Security and Surveillance use case.

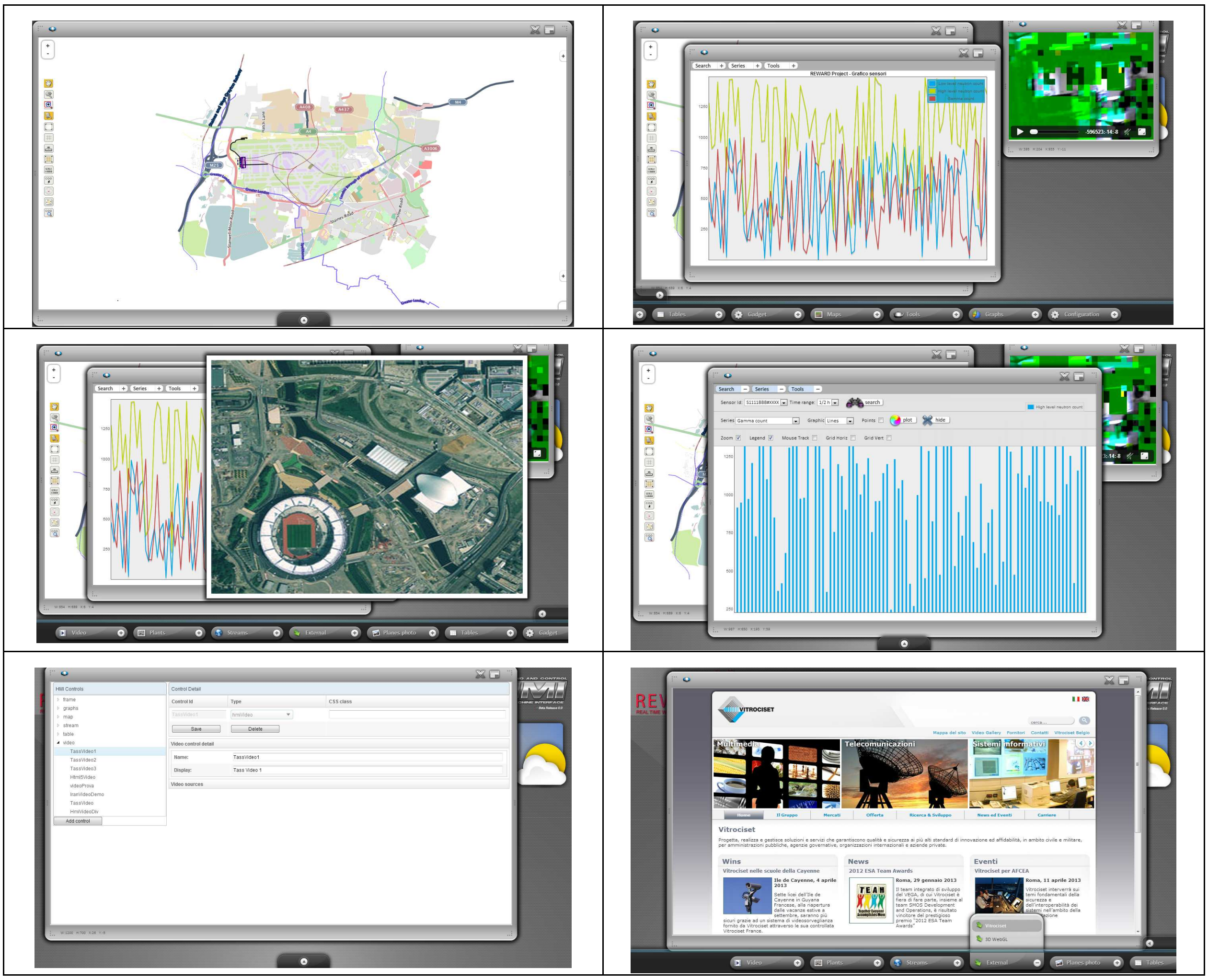


Figure 1: Overview of the SMARTCOP components

Note that smart cities (such as the city of Santander) are ideal environments for deploying the SMART security and surveillance application, given the increasing need to ensure the safety of the citizens in urban environments, but also due to the availability of sensing infrastructures in the scope of modern smart cities.

2.2 Live News

A first approach to the live news application of SMART has been conceptualized and developed in the first live news PoC release. The purpose of the live news application is to deliver information about what is going on in a smart city (such as the City of Santander), including dynamic information stemming from the analysis of data from physical sensors and social networks.

The concept is a web interface to be used by citizens (final users) to access to real time data and news, automatically generated by SMART. These data are presented taking into consideration the user's location (i.e. in a location aware fashion) and based on information stemming from both sensors and social networks.

The first release of the live news PoC has been developed in order to test the SMART architecture, testing the access of the application layer to both Social data and sensor's data in the Edge nodes trough the SMART search engine. Furthermore, the initial implementation includes the following data:

- News from the RSS Feed.
- Social conversation about specific topics, geolocalized in the city of Santander, in this case Twitter.
- Colour tendency: the main colors detected from the cameras on the active area. This functionality is based on visual signal processing algorithms developed in WP3.
- Crowd density. Similarly to colour tendency this functionality is based on visual signal processing algorithms developed in WP3.
- User specific query, retrieving the tweets from a specific area in the city of Santander. These queries are empowered from the SMART search engine and social network manager.

The PoC has been developed with two main views: City View and Data View:

- The City view is the default view, where the user can access instantly to the latest observations of the Smart system in his location.
- The Data View is another visualisation of the Smart observations, but with more details, in this view the user can select the area of the city and the topic he is interested in, or formulate a specific query.

Figure 2: Overview of the City View.

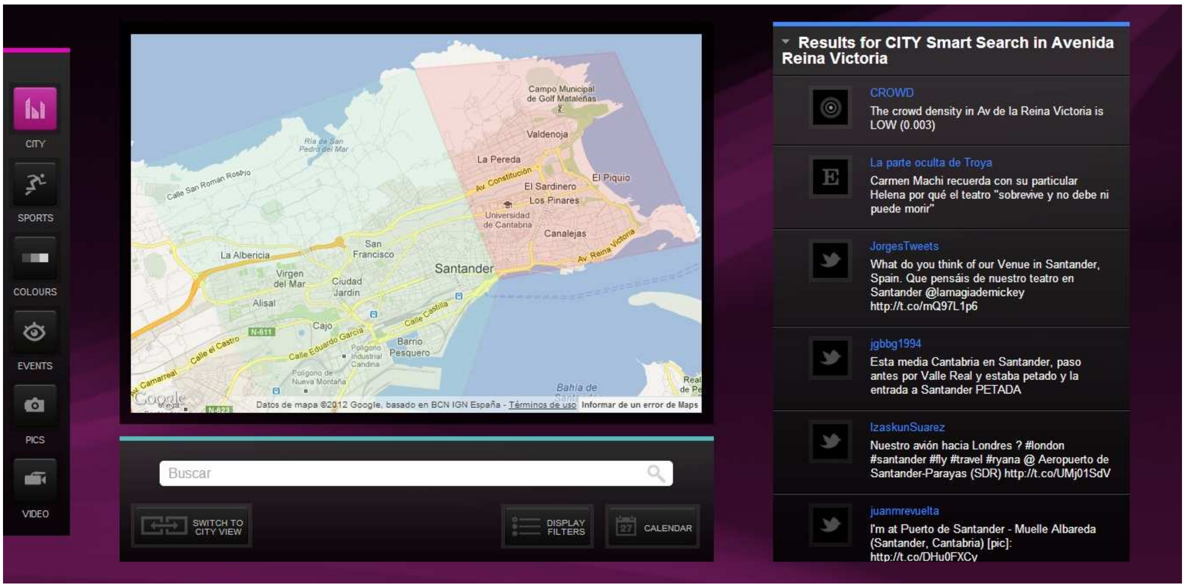


Figure 3: Overview of the Data View

2.3 The Social Search use case

Social media platforms are rapidly growing in terms of their user population and activity, which has led to a huge growth in user-generated content on the web. This has triggered the development of application and services which exploit the content of this data and the social interactions between the users to deliver an added value to an end user. Examples of such services include recommender systems based on collaborative filtering [Su2009], event monitoring and identification with social media [Becker2012], among others. Most notably there have been a lot of attempts to harness social media for personalised search and recommender systems.

The *social search* use case is another example of such applications. It aims to use the SMART search services for smart cities to develop novel social search platforms, which enable citizens to access information about their urban physical vicinity in a socially personalised manner. In particular, SMART can access information about end users from their profiles on popular social media networks such as Facebook and Foursquare in order to personalise the search results of real-world events. For example, SMART will be able to answer user's questions such as: "*where are there music events in the city that are attended or likely to be attended by my friends?*" Or "*which near-by restaurants are now busy and that my friends have been to?"*.

In Section 5, we detail the architecture of the social search use case in SMART and discuss with examples the methods employed to provide personalised search results. We will also discuss the software implementation of the integrated system where we highlight the main components and their interactions.

3. Security and Surveillance

SMARTCOP is a software and architectural framework to build up the operator interfaces for command, control and surveillance systems.

SMART COP provides following essential features:

- **Browser-based Web Interface** – Enables easy access and collaboration among organizations and personnel, also supports mobile and operations centre users – across town or around the world.
- **Open Platform** – Provides integration with virtually any known types of security systems and devices. It provides both interfaces based on standard message queue protocols and software libraries to develop custom connectors.
- **Dynamic Geospatial Mapping** – Enables precise location mapping and display of situations, people and alarms for fixed and mobile devices, and displays location data supporting incidents in near real time for each specific situation or even globally across all situations.
- **Modular Platform** – Provides dynamic adaptation to changes in situations, devices, configurations, policies and reporting while the system is running.

To best fit SMART capabilities and requirements, SMARTCOP defines a common architecture for the presentation layer (mashup components) and an opened communication schema. For each component in the HMI SMARTCOP allows to select the protocols and the standards for interfacing with external systems.

Moreover, SMARTCOP implement an SDK to extend the platform to comply with specific requirements.

3.1 SMARTCOP Architecture

SMARTCOP architecture and its technology stack are depicted in the next figure (Figure 4). In the figure, the boxes represent the logical components, while the arrows show the communication interfaces between them (it is implied that each logical component does not necessarily have to be hosted on dedicated hardware; one computer could accommodate more than one component).

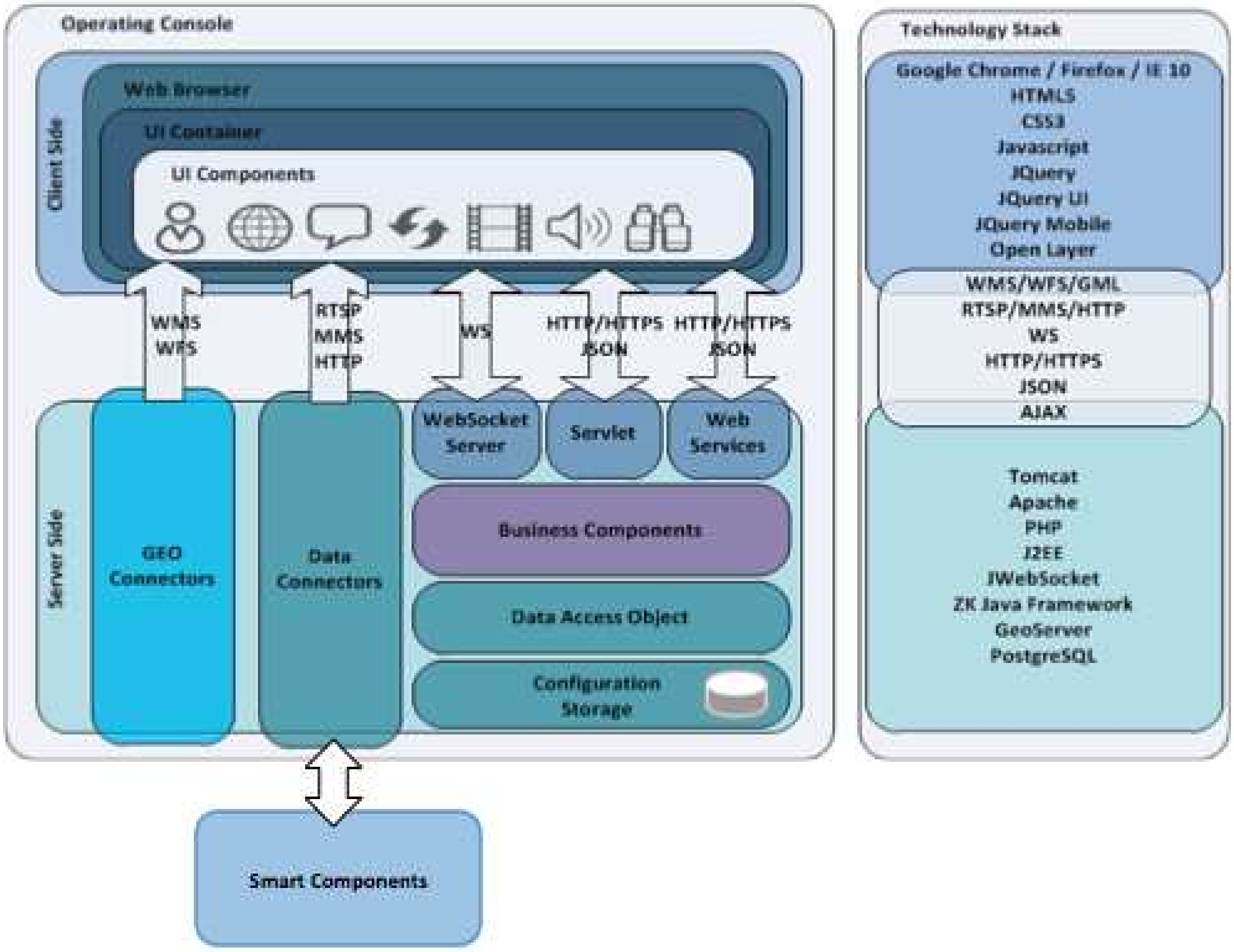


Figure 4: SMARTCOP Architecture

The **Client Side** is the higher level of the SMARTCOP architecture stack. It comprises all the software components with a graphical user interface that will enable the operators to interact with the system. This component represents the operating console of the system. It provides to the operators the 2D/3D representation of the monitored area, the displacement of the sensor network, and the alarms detected by the system.

It gives the operator the capabilities to interact with the console, in order to manage the ordinary and the extraordinary situations.

The **Server Side** provides all the SMARTCOP business logics and the data connectors to GEO Servers and the SMART Search Engine

3.2 Implementation

The implementation of the logical schema represented in Figure 4 is represented in the following figure. The schema represents connections between software components: how components communicate and which protocols have been used to communicate.

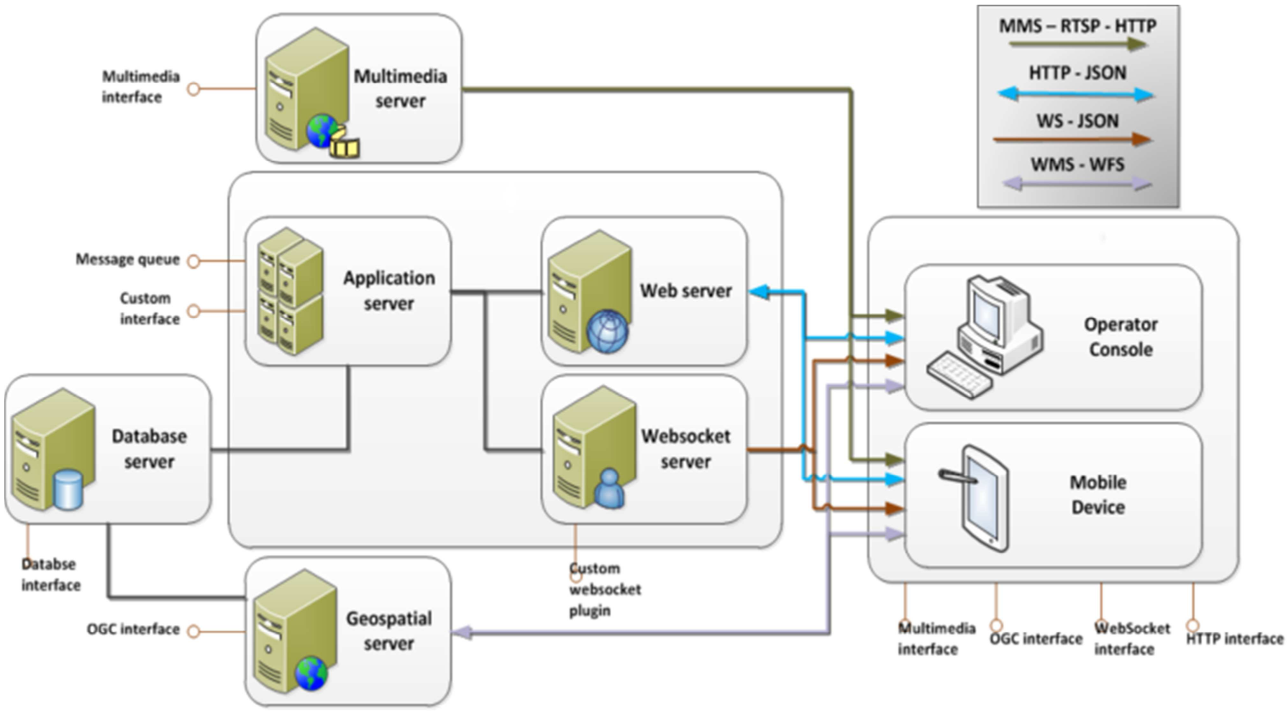


Figure 5: Schema of logical components communications and protocols

As shown, SMARTCOP supports all the most common used communication protocols:

- HTTP/HTTPS – JSON
- WS - JSON
- WMS – WFS - GML

3.2.1 Client side technologies used in SMARTCOP

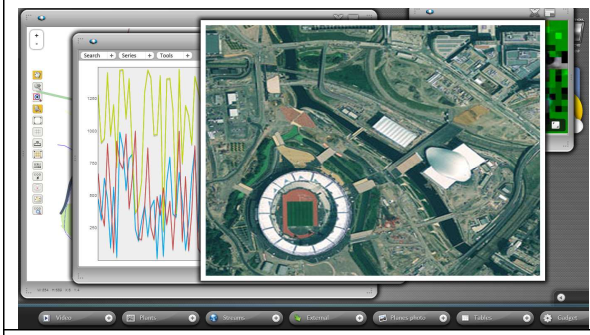
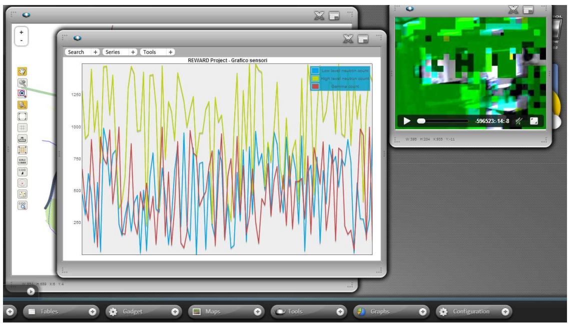
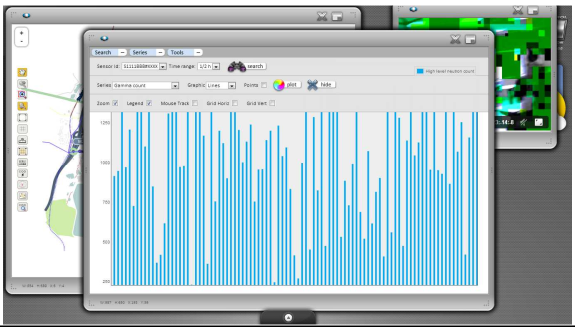
- **HTML5 tags:** HTML5 is a markup language for structuring and presenting content for the World Wide Web
- **CSS3 attributes**: used to dynamically change components within the client console.
- **JQuery framework**: small, and feature-rich JavaScript library;
- **OpenLayers framework**: used to easily integrate maps into a web page;

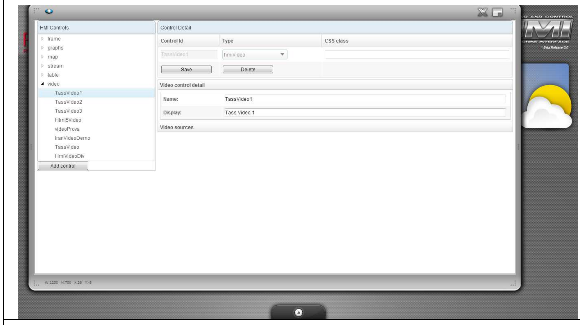
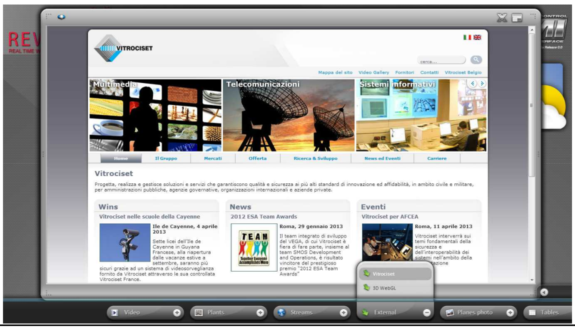
3.2.2 Server side software components in SMARTCOP

- **Tomcat server:** Container for the SMARTCOP web application;
- **JWebSocket server:** Java based streaming and communication applications on the web;
- **GeoServer:** an open source software server written in Java that allows users to share and edit geospatial data;
- **PostgreSQL – PostGIS**: an open source database server and its geospatial component.

3.3 Components released in this version

The first release of the SMARTCOP use case implements a common desktop (the main view used to access to the SMARTCOP functionalities), the communication services between SMARTCOP , SMART, and the HMI components.

 	MAP visualization and navigation. This component uses both open source geo server (based upon Geoserver) and Google maps. It provides all the facility to pan, zoom, tilt, and edit maps. This view allows to easily move and query from different edge nodes including the possibility to move from different views and layers from the same location.
 	SMARTCOP provides a large sets of graphs to display information from the real world sensors. Graphs can be generated in realtime (polling data from different sensors) or batch using historical information locally stored. The PoC provides also a component to access to video libraries both internal to SMART and external (i.e. YOUTUBE).

	Configuration panel allows to configure the SMARTCOP behaviour, protocols, connectors.
	Web and Social Network components allows to connect to the web or the social network.

SMARTCOP also includes components to manage multimedia content like images or weather forecast information coming from different public sources.

3.3.1 Use of SMART Components

The SMARTCOP use case makes use of the following components of the SMART search framework [D6.2]:

SMART Components used in the Social Search Use Case	Purpose
Edge Node Server	**To provide metadata from AV Sensors and display it as graphs**
Terrier Search Engine (SMART Edition)	**Access to events from sensors**
Social Network Manager	**Access to content related to specific words or hashtags**

Table 3: List of SMART Search Engine Components used in the first release of the SMARTCOP

Because of the security domains requires specific data to work, the current version of the SMARTCOP implements all the connectors to the specified components but is capable to get and display raw data coming from SMART components. A specific task is on-going to train the whole system to produce events and data useful to enable SMARTCOP to properly identify security related problems.

4. Live News (Blogs, Portals and Wikis)

4.1 Technical overview

The first release of the Smart Live News use case is built under the architecture detailed in the following figure:

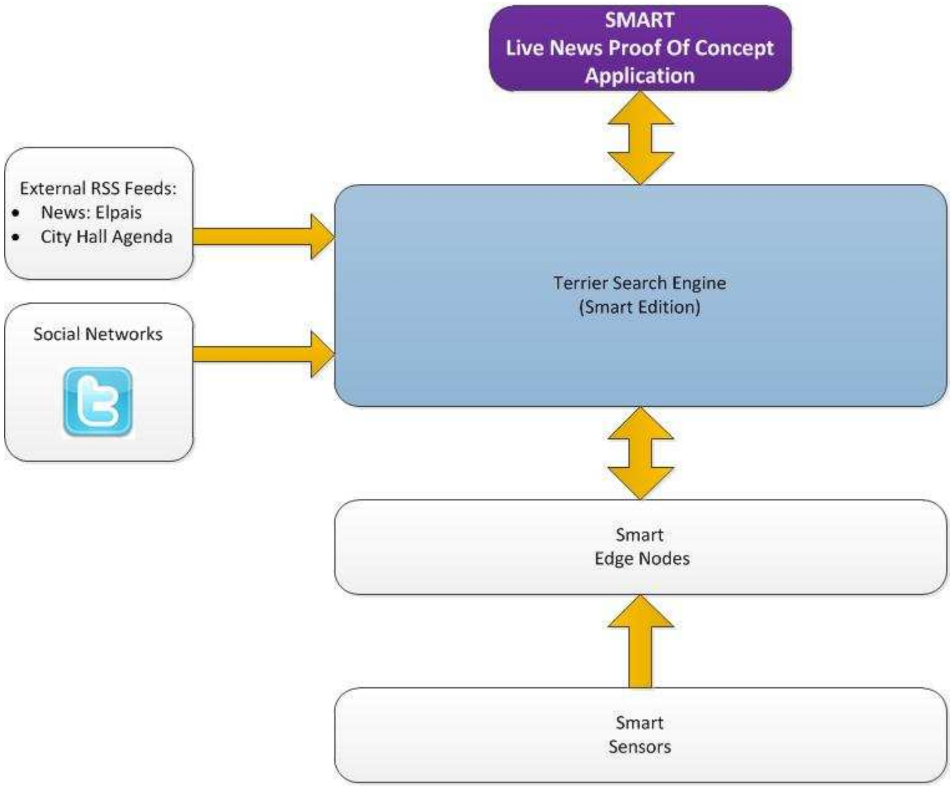


Figure 6: Smart Live news PoC Architecture overview

As described in the figure, the live news application retrieve the data through queries to the terrier Smart Search engine.

The search engine component, indexes data from the Smart sensors through the Edge nodes and also from external sources, e.g. Rss feeds and Twitter.

Data form physical sensors, in this first release are coming from recorded video files looped to simulate live data collection.

4.2 Use of SMART Components

The first release of the SMART Live News makes use of the following components of the SMART search framework:

SMART Components used in the Live News Use Case (First Release)	Purpose
Edge Node	**The data from the sensors (video) are stored in the edge nodes and there runs the audio-visual algorithms for crowd density calculation and the color tendency**
Terrier Search Engine (SMART Edition)	**Access to the latest events detected from the sensors (crowd density, colors), social network conversation, rss news feeds**

Table 1: List of SMART Search Engine Components used in the first release of the Live News Use Case Implementation

4.3 Implementation

The first release of the live News use case have been developed as a web interface with two main views: city view and data view.

4.3.1.1 City View

The city view is an adaptative interface that changes graphically depending on the data coming from the Smart sensors.

In the figure below, we can see the interface changing depending on the crowd's density on the user's location, and the time.

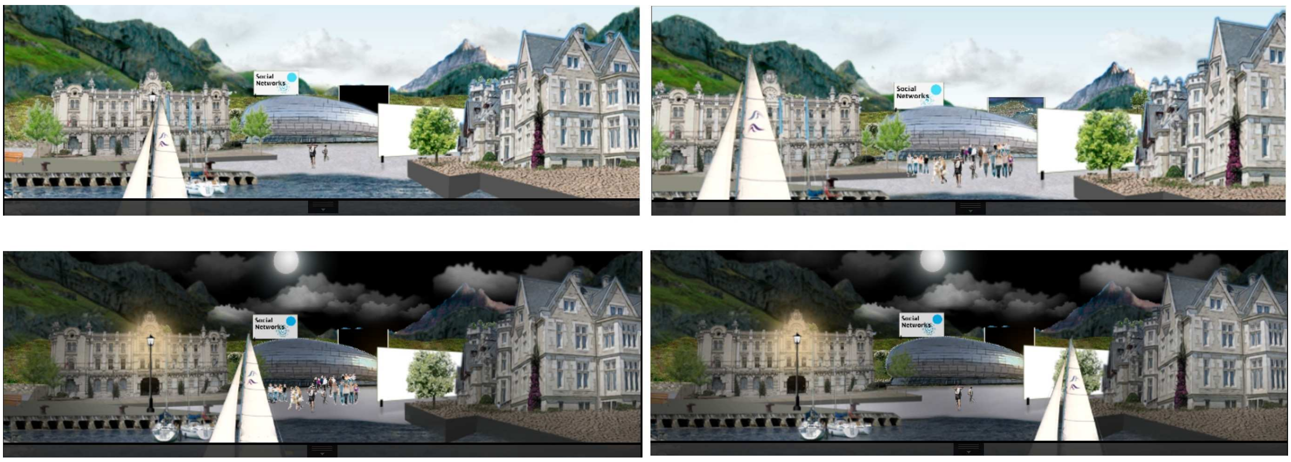


Figure 7: City View

The City view interface is developed using, HTML5 and parallax to build the immersive experience.

On the following figure the layers between the user interface and the Html5 layer are detailed.

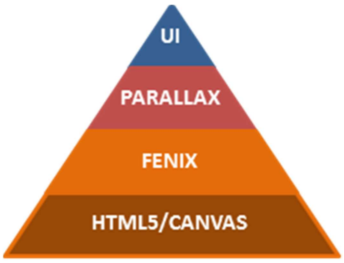


Figure 8: City View Layers

Fenix: is the engine for drawing bitmap on html5, controlling and managing the visualization processes of the graphic components as independent elements.

Parallax: is the visual effect created moving layered 2D images of a composition at different speeds/perspectives creating depth sensation.

The layers of the city view used for the prototype are described in the following figure:

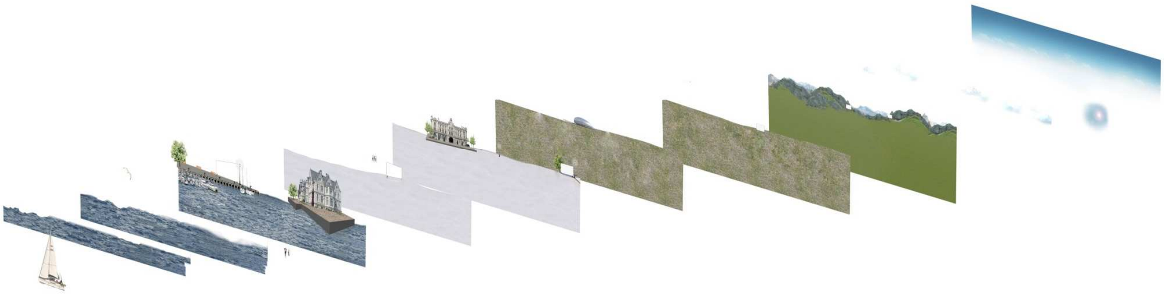


Figure 9: Design layers

4.3.1.2 Data View

The Data view is an interface where the user can retrieve specific data from the Smart system depending on his location or the location selected.

As shown in the following figure, in the first release of the live news use case, the city of Santander has been segmented into two areas.

Santander EAST: "Avenida de la Reina Victoria"

Santander OUEST: "Plaza del ayuntamiento"

The data presented on the view interface are coming from two sensors with simulated location:

- Avenida de la Reina Victoria
- Plaza del ayuntamiento

Figure 10: Santander areas

In the following figure an overview of the data view is presented. In this view the interface detects the user's location and displays by default data from the user's location, nevertheless the user can select another area to consult by clicking in the Map.

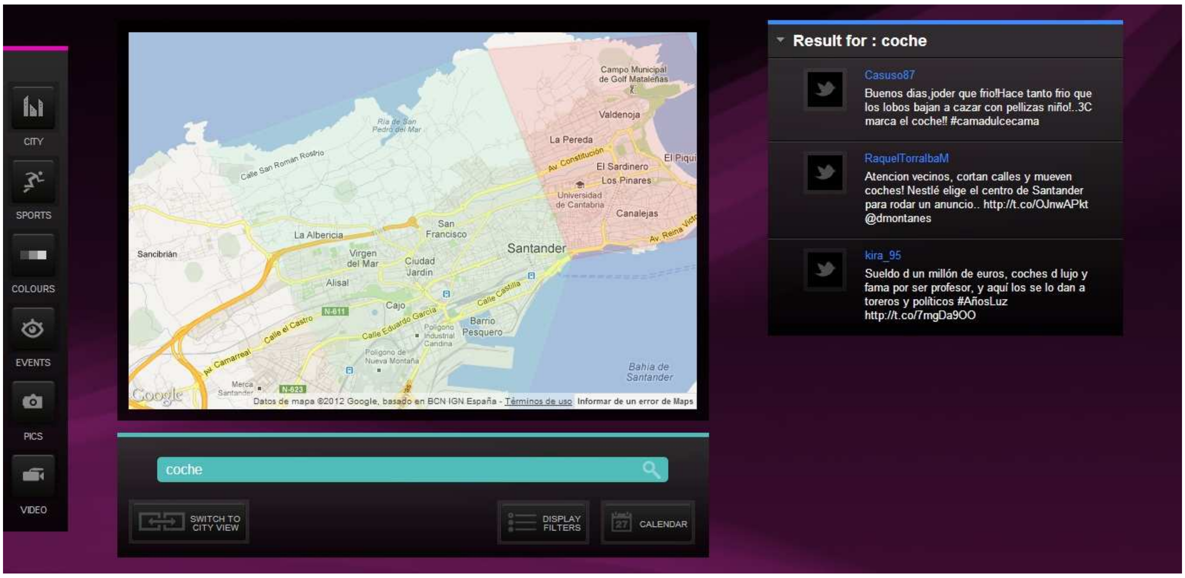


Figure 11: Data View – User's specific query

5. The Social Search use case

As discussed in Section 2, the aim of the social search use case is to personalise the search results of real-world events based on information about users from their footprint in the social media. In this section, we first give a background on personalised social search. We then introduce the architecture of the use case and detail with examples how SMART uses social media information about a user for providing a social search experience. Finally, we present the implementation of the first prototype of the social search use case.

5.1 Background

Personalised search is a challenging problem and it requires modelling the interest of the user, i.e. building some sort of a user profile, which is then used to provide more tailored results to the user needs [Belkin2008]. Traditionally, user's interactions with search engines, e.g. queries and click information, were used to capture user preferences and interests for building the user profile, e.g. [Agichtien2006]. The profile can be then used for personalised search by re-ranking or filtering the search results [Shen2005]. Social media platforms open opportunities for personalised search as they can capture a wealth of knowledge about their users. Two distinct sources of information can provide indications about users' interests in social media platforms and can therefore be used to personalise search. (i) The first source of information is the explicit user interactions which include the content of the posts and the messages they share, the web pages they "bookmark" or "like" and the places they have been to. Personalisation of search based on this type of information has been done for example with bookmarking social networks e.g. [Kim2005] and social tagging platforms [Noll2007]. (ii) The second type of information that is available in social platforms to infer user interests and preferences is the user's social graph, i.e. the people they are "friend with" or whom they "follow", based on the assumption that closely connected people will have similar interests. Search personalisation that exploits the user's social connections has also been investigated, for example in enterprise social networks [Carmel2009]. In both cases, personalised search based on user profiles extracted from social media is often referred to as social search [Amitay2009], which is the focus of our use case.

In the following sections, we detail how the Search layer exploits the social information found about a user by the SMART Social Network Manager (SNM) to model user interests (build a user a profile) and then use this profile to provide personalised search results. In particular, we give details on how we build a user profile and the methods employed by the search layer to perform such personalisation. We will also discuss the software implementation of the integrated system where we highlight the main components and their interactions.

5.2 Social Search in SMART

Figure 12 shows the overall architecture of the social search use case outlining the main software components and their interactions. The end user application proxies the requests of the user to both the SNM and the search layer.

The SNM in return provides social information about the user to the search layer. The search layer has two main components for social search. One deals with building the user profile from his/her social information and the other one personalises results based on the built profile. In this section, we ill discuss these components and we provide concrete examples on how they work using Facebook social media data.

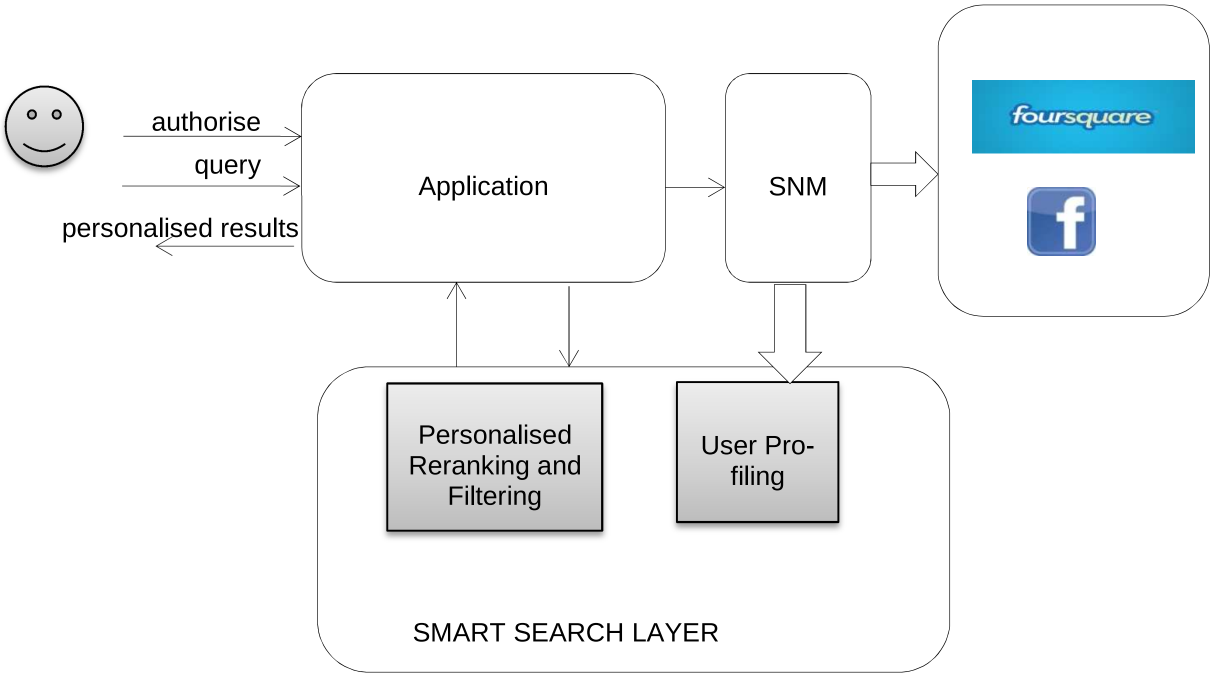


Figure 12: System Architecture

5.2.1 Use of SMART Components

The first release of the SMART Social Search use case makes use of the following components of the SMART search framework:

SMART Components used in the Social Search Use Case	Purpose
Edge Node Server	To provide metadata about the real world in a number of locations in order to identify events of interests to the user
The SMART Search Layer	To index sensor and social streams and personalise search results based on the social profile of the user
Social Network Manager	To access social information about a user

Table 3: List of SMART Search Engine Components used in the first release of the Social Search Use Case

5.2.2 Modelling user social interests

As discussed in Section 5.1, there are two distinct types of information found in social media that can be used to model user's interests: (a) the explicit user interactions, and (b) the social graph of the user. The social search use case is designed to use both types of information. In this section, we detail what information are used to model user's interests (the user profile) in the search layer.

The search layer accesses information about a user in social media upon authorisation from the user. The social profile is built on the fly and is later used to personalise search. For an end user (u), we refer to their social profile as $P(u)$. In the following, we identify the potentially useful information about the user in social media that can be used for personalisation in SMART when searching for events in the physical world. Note that the description here is agnostic to the social network instance (whether it is Facebook or Twitter). We assume that one or many different social networks may be used to provide such information

- The URLs that have been shared or posted by the user. The content of these pages can give the search layer indications about the user's interests. These pages are denoted by $W_u = \{w_1, w_2, .., w_k\}$
- The textual content of the user posts on social media, e.g. their tweets or Facebook statuses. This content may be used to identify the topics that user is interested in. These posts are denoted by $A_u = \{a_1, a_2, .., a_l\}$
- The places or venues that the user has "checked in" recently and the time stamps of the visits to these places. Both the geographical locations and the description of these places can give indication about the preferences of the user in terms of the areas they are likely to visit (their locality) or the topics of the events they prefer. These user visits are denoted by $V_u = \{v_1, v_2, .., v_m\}$.
- The social graph of the user which includes people that are "friends" with the user or whom the user "follows". The profiles of these users may give indication on the user's individual preferences. The direct connections to the user from the social graph is denoted by S_u.

The user profile in the search layer is a combination of all the above. Hence, the user profile $P(u) = \langle W_u, A_u, V_u, S_u \rangle$. It is used by the search layer to get more context when answering a query and retrieving events.

5.2.3 Personalised social search techniques

The Search Layer will implement a number of techniques to the profile of the user built from their social media footprints (Section 2.1) to personalise search results.

In SMART, the search layer works as follows. Given a user query (q), and a user context identified by the location of the user (l_q) and the time of the query (t_q), specialised event retrieval models, e.g. the one we developed in [Albakour2013] are employed to identify and ranks tuples of <l, t>, where there is an interesting event has started in the location l at point of time t, which is also relevant to the user query. To perform this ranking, the search layer assigns a score to each tuple <l, t> based on the input q, l_q, t_q which is denoted by Score(<l, t> |q, l_q, t_q). This is based on the sensor signals from the edge nodes and the topics mentioned in the social posts (tweets) corresponding to l and t. Note that the social media information used for retrieving events are not associated with a certain user and rather aggregated from the entire population. In other words, they should not be mistaken with the extracted user profile from social media. In the special case when there is no keyword query provided by the user (zero-query), the search layer will use to the context of the user to recommend events of any topics, e.g. the ones closer the user can be suggested.

In the social search use case, the user profile $P(u)$ is used as an additional valuable source of information about the user's context. This profile is then employed to modify the ranking of the results such that they satisfy better the user in question. In other words, the search layer assigns to the tuples <l, t> based on the input q, l_q, t_q, $P(u)$ which is denoted by Score(<l, t> |q, l_q, t_q, $P(u)$).

In the following, we detail some of the techniques to perform this personalisation. Those techniques are adapted from previous social personalisation approaches which may work well in SMART. In addition, we will investigate novel social personalisation techniques in SMART for future releases of the use case.

- Personalised query expansion works by adding new terms to the user's original query based on the user profile [Chirita2007]. In SMART, we can rank terms for this kind of query expansion by using the title or the content of the web pages they share W_u and/or the content of their posts A_u. In addition, the description of the places they have visited in the past V_u can be used to extract useful expansion query terms.

- Filtering according to categories of interest has been applied in social search. For example, ODP (the Open Directory Project)[1] data was used to categorise bookmarks and further filter the results to match a user profile in a social bookmarking web site [Luxenburger2008]. Similarly, in SMART we can categorise events identified by SMART according to the venues they occur at. Venues which match the user profile as indicated by their previous visits $V_u$can be then filtered and provided to the user. More elaborate techniques to detect the categories of the places which are of interest to the user from their posts A_u or shared URLs W_u will be investigated.
- Reranking based on the social graph can be adapted for SMART. For instance the UserRank algorithm [Bender2008], derived from the PageRank algorithm, has been applied for social search on bookmarking platforms where a document receives higher scores if it has been "liked" by a friend. Similarly, in SMART we can recommend venues that have been visited by the user's friends if their visits are available publicly on the social media (as we only build the user profile on the fly).
- The user's locality can be modelled using the places and venues that have been visited by the user and recorded in their social media footprint, i.e. the set V_u. Using this information, we can estimate the user's centre of mass coordinates by averaging locations over all the locations of visited places [Cheng 2011]. Using this information the search engine can re-rank results (events) based on their distance to this centre which is the likely location of the user social activities as indicated from their social media footprint.

Next, we will give concrete examples of information found about users in social media to give the reader scenarios where personalised social search can be useful to the user.

5.2.4 Social Profile Examples

In Figure 13, we give an example of social information found about a user in social media where a list of liked pages on their Facebook profile is provided. These pages correspond to the set W_u . The content of those pages can be used to expand the original query in order to match a more personalised ranked list of events. For instance, a personalised ranked list for a zero-query (where no keyword query is provided) to the user in the example will favour events happening in bars or conferences for software developers. In Figure 14, we show another example where a list of pages that the user is subscribed to on Facebook is provided. This is another source to extract the set W_u. Here it is not straightforward to extract terms relevant to the user. We may need to use the content of these pages to derive useful terms for personalised query expansion or filter the results according to a certain category. For instance, we can use the Facebook API to get a description of those pages. It seems that they all refer to musicians/DJ players. Finally in Figure 15, we provide an example of the "check-in" information collected from a Facebook profile of the user. This information can be used to calculate the user centre of mass as discussed in Section 2.2 and rerank results such that events closer to the centre of mass are ranked higher.

Likes on Facebook:

```
{
  "data": [
   {
    "category": "Arts/entertainment/nightlife",
    "category_list": [
     {
      "id": "110290705711626",
      "name": "Bar"
     }
    ],
    "name": "The Place",
    "id": "132630140127858",
```


[1] http://dmoz.org

```
      "created_time": "2013-04-17T15:13:01+0000"
    },
    {
      "category": "Just for fun",
      "name": "4play",
      "id": "356840747753590",
      "created_time": "2013-04-12T16:37:02+0000"
    },
    {
      "category": "Product/service",
      "name": "Facebook Developers",
      "id": "19292868552",
      "created_time": "2013-04-11T13:39:01+0000"
    }
  ]
}
```


Figure 13 Example of Page Likes on Facebook


```
Subscribed to:
{
  "data": [
    {
      "name": "Robbert van Eijndhoven",
      "id": "100000990836444"
    },
    {
      "name": "Jan Xfusionoisufx",
      "id": "530121739"
    },
    {
      "name": "Johan van Roy",
      "id": "1089709235"
    },
    {
      "name": "Siobhan Hazard",
      "id": "100000208346769"
    }
  ],
  "summary": {
    "total_count": 4
  }
}
```


Figure 14 Example of subscriptions on Facebook


```
Locations (Checkin) Response:{
  "data": [
    {
      "id": "10200553673136621",
      "from": {
        "name": "George Sk",
        "id": "1002866274"
      },
      "type": "status",
      "place": {
        "id": "180782971949271",
```

```
        "name": "The Mall Athens",
        "location": {
          "street": "Ανδρέα Παπανδρέου 35, Μαρούσι",
          "city": "Amaroúsion",
          "state": "",
          "country": "Greece",
          "zip": "15122",
          "latitude": 38.044604374061,
          "longitude": 23.790218165769
        }
      },
      "created_time": "2013-04-18T08:35:55+0000"
    }
  ]
}
```


Figure 15 Example on User Check-ins on Facebook

5.3 Implementation

In this section, we describe the implementation for the first prototype of the social search use case in SMART. In particular, the first prototype consists of components within the SNM and the search layer, namely the social extractor service within the (SNM) and the user profiling and personalised reranking within the search layer (See Figure 12).

5.3.1 The Social Extractor Service

A web application was created via Facebook, providing information data about a logged in Facebook user. At first the user is asked to log in with his Facebook account through Facebook's Oauth security protocol. A Facebook application was created to handle this log in and request for permission to access the user's profile data that are needed for the use case. Upon approval the user is successfully logged into Facebook through the Facebook application and the Facebook API provides the requested user data for extraction. The user data is consisted of the Facebook pages liked by the user, user check-ins and the user's followers as well as the people the user is following.

5.3.2 User Profiling and Personalised Re-ranking

The initial implementation parses the Facebook profile provided with the social extractor service and builds a profile $P(u)$ from the information found in this profile. In particular the profile consists of the following:

- Pages liked or subscribed to by the user which maps to the set A_u. Examples are provided in Figures 13, and 14.
- The locations that have been visited by the user which maps to the set V_u. Examples are provided in Figure 15.

Currently, the personalisation performed is estimating the user's centre of mass as discussed in Section 3.2. Using this information the search engine re-rank results (events) based on their distance to this centre which is the likely location of the user social activities as indicated from their social media footprint.

6. Future Steps

In this section, we describes future plans for our three use cases and highlight the new features that we will be developed in the subsequent releases.

6.1.1 Security and Surveillance

Future steps include realization of real use cases intended to test SMART capabilities detecting hazardous situation and planning intervention. Three different use cases will be realized

6.1.2 Explosion detection

This use case uses sensors to detect explosions. This use case will process data from different sensors: audio, video, and temperature.

The information that will be used to infer security alerts are listed below:

- sudden variation of the direction of the crowd;
- temperature suddenly grows;
- light suddenly grows;
- audio sensors detection;

6.1.3 Crowd Panic Detection

This use case is intended to demonstrate how SMART can be used to detect and potentially prevent crowd panic. Sensor Data considered relevant will be:

- crowd noise over a threshold;
- sudden variation of the direction of the crowd;
- the crowd moves away from a location;
- the crowd converges to a location;
- the crowd suddenly change direction;

6.1.4 Hooligans

This use case is cantered on merging data from Social Network and data from sensors (where available). This use case is made of two different scenarios:

Group of hooligans are converging to a location.

1) NO SENSORS AVAILABLE. Each group of hooligan could be detected using social network data evaluating the density of tweets, related to specific words, in a certain location. Tracing locations where the density of tweets is over a threshold, it could determine the position and the direction of the hooligans.
2) SENSORS AVAILABLE. If hooligans reached a location covered by sensors, the crowd analysis and audio sensors could be used to identify what is happening in the area.

6.2 Live News (Blogs, Portals and Wikis)

The next steps of the live news use case could be summarised on the following tasks:

- To integrate real data from the sensors installed in the city of Santander
- To be able to retrieve more data from the Smart edge nodes.
- To retrieve high level events merging information from the physical sensors and from the social networks.(example listed in D4.2 [5])
- To segment the city of Santander into more logical areas..
- Add more filters to the results, for example to be able to select a specific day.

6.3 Social Search

The current release of the social search use case has built the essential components to make SMART a social search engine. In particular, the current release has implemented an interface with facebook to provide personalised rankings based on previous places (venues) visited by the user as indicated from their facebook profile. In the next release, we will implement further personalisation technique, such personalised query expansion and category filtering, discussed in Section 5, using other social information available in the facebook profile (the liked pages and the subscribed pages). Moreover, we investigate using other social networks to derive richer representation of the user profile. Subsequent releases of SMART will provide reasoning components over audio/video and social metadata, which can derive more certainty about the occurrence of different types of real world events. The Social search use case aims to incorporate that in the subsequent releases by mapping these events to different categories and filtering them according to the profile of the user. Finally, we aim to deploy the social search use case as a web application and a mobile app. Initially, our plan is to start using these applications internally in the SMART project to derive initial feedback from real users. Upon receiving this feedback, we aim to further expand our user base in order to validate and evaluate our personalisation techniques using A/B testing methods.

7. Conclusions

In this initial version of the D6.1 deliverable we have provided a brief description of the three validating use cases of the project, along with their initial implementation based on the SMART search framework. This initial implementation has validated the functionality of several components of the SMART project including the SMART version of the Terrier search engine, the SMART edge node, several signal processing algorithms developed in WP3 and the SMART Social Networks Manager. The validation of these components has certified that they can be used to developed/integrate realistic applications. Indeed, as part of the use cases development, application integrators have succeeded in downloading, using and deploying most of the components that have been already been released in the SMART open source project as part of deliverable D6.2. Furthermore, information from visual processing components has also been integrated as part of the live news use case, while the "social" features of the search engine have been integrated as part of the social search use case.

Two more releases of the use cases will be produced till the end of the project. As part of these subsequent releases, the partners will validate additional components of the SMART platform, while also enabling application developers to leverage the improved and more mature versions of the components outlined above. To this end, the implementation of the use cases will make use of improved version of the SMART components, which will be part of subsequent releases of deliverable D6.2. Based on these improved versions application developers will be able to develop enhanced and more robust versions of the use cases. Furthermore, the project's workplan include the deployment of the use cases on the city of Santander, which is currently deploying the SMART baseline infrastructure.

The three SMART use cases that are described in this document are aimed to provide a PoC regarding the potential use of the SMART framework for developing applications that leverage social networks and sensor networks, along with their intelligent combination in a smart city environment. As part of the SMART open source project we envisaged that these use cases could serve as indicative examples that could inspire the SMART open source community towards developing additional innovative applications.

The SMART use cases have demonstrated how SMART can be used in different applications domains contributing to create cross-knowledge between the consortium participants. Such knowledge will represents, in the future, a valid knowledge base for the open source community.

They have also demonstrated how fusing data from sensors and social network can be used to infers detailed information about events from the real word or how social network can be used as a sensor itself (i.e.the case of the SMARTCOP application which will use data from social network and sensors trying to evaluate how groups of hooligans are moving).

Finally, this step has provided useful-planning guides for the future components design and development.

8. References

[Agichtein2006] E. Agichtein, E. Brill, S. Dumais, and R. Ragno. Learning user interaction models for predicting web search result preferences. In Proceedings of SIGIR, pages 3–10. ACM Press, 2006.

[Amitay2009] E. Amitay, D. Carmel, N. Har'El, S. Ofek-Koifman, A. Soffer, S. Yogev, and N. Golbandi. Social search and discovery using a unified approach. In Proceedings of HyperText, pages 199–208. ACM, 2009.

[Albakour2013] M. Albakour, C. Macdonald, I. Ounis. Identifying Local Events by Using Microblogs as Social Sensors. To appear in proceedings of OAIR 2013, Lisbon, Portugal.

[Becker2012] H. Becker, D. Iter, M. Naaman, and L. Gravano.Identifying content for planned events across social media sites. In WSDM'12.

[Belkin2008] N. J. Belkin. Some(what) grand challenges for information retrieval. SIGIR Forum, 42(1):47–54, 2008.

[Bender2008] M. Bender, T. Crecelius, M. Kacimi, S. Michel, T. Neumann, J. X. Parreira, R. Schenkel, and G. Weikum. Exploiting social relations for query expansion and result ranking. In Proceedings of ICDE Workshops, pages 501–506. IEEE, 2008.

[Carmel 2009] D. Carmel, N. Zwerdling, I. Guy, S. Ofek-Koifman, N. Har'El, I. Ronen and S. Chernov Personalized social search based on the user's social network. In *Proceedings of the 18th ACM conference on Information and knowledge management* CIKM (pp. 1227-1236). 2009.

[Chirita2007] P. A. Chirita, C. S. Firan, and W. Nejdl. Personalized query expansion for the web. In Proceedings of SIGIR, pages 7–14. ACM, 2007.

[Kim2005] H. R. Kim and P. K. Chan. Personalized search results with user interest hierarchies learnt from bookmarks. In WEBKDD,volume 4198 of LNCS, pages 158–176, 2005.

[Luxenburger2008] J. Luxenburger, S. Elbassuoni, and G. Weikum. Matching task profiles and user

[Noll2077] M. G. Noll and C. Meinel. Web search personalization via social bookmarking and tagging. In K. Aberer, K.-S. Choi, N. Noy, D. Allemang, K.-I. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudr´e-Mauroux, editors, The Semantic Web, volume 4825 of LNCS, pages 367–380. Springer, 2007.

[Shen2005] X. Shen, B. Tan, and C. Zhai. Implicit user modeling for personalized search. In Proceedings of CIKM, pages 824–831.ACM, 2005.

[Su2009] X. Su, & T. M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, *2009*, 4.