



SEVENTH FRAMEWORK PROGRAMME
Networked Media

Specific Targeted Research Project

SMART

(FP7-287583)

**Search engine for Multimedia
environment
generated content**

D6.2 Integrated Open Source Framework

Due date of deliverable: 28-02-2013

Actual submission date: 15-03-2013

Start date of project: 01-11-2011

Duration: 36 months

Summary of the document

Code:	D6.2 Integrated Open Source Framework
Last modification:	13/03/2013
State:	Final
Participant Partner(s):	ATOS, AIT, GLA, S3Log, Telesto
Author(s):	Dyaa Albakour, Craig Macdonald, Aristodemos Pnevmatikakis, Jose Miguel Garrido, Thanos Alexiou, John Soldatos, Massimiliano Tarquini
Fragment:	No
Audience:	<input checked="" type="checkbox"/> public <input type="checkbox"/> restricted <input type="checkbox"/> internal
Abstract:	This deliverable is the report accompanying the the first release of the open source implementation of the integrated multimedia search engine, which is available for download from the SMART open source portal at: http://opensoftware.smartfp7.eu/ . This release integrates functionalities and results elaborated in other technical work packages of the project (notably WP3, WP4 and WP5). This first release of the SMART open source framework will be used to integrate the first version of the SMART applications.
Keywords:	<ul style="list-style-type: none">• Integration, open source release
References:	DoW



Table of Contents

1	Executive Summary	4
1.1	Scope	4
1.2	Audience	4
1.3	Summary.....	4
1.4	Structure.....	4
2	Overview of the First Release	6
2.1	SMART User Groups	6
2.2	Objectives of the First Release	6
2.2.1	Application developers.....	6
2.2.2	Special interest groups	7
2.2.3	Metadata providers	7
2.2.4	End-users.....	7
3	Components of the Release.....	8
3.1	The Edge Node Server	8
3.1.1	Feed handling	8
3.1.2	Perceptual components	8
3.2	The Social Network Manager.....	9
3.3	The Search Engine	9
3.4	Third-party Components	10
4	Description of the PoC (Proof-of-Concept) Application.....	12
5	Building a SMART Community.....	13
5.1	Plan for next releases	13
5.1.1	Plans for the edge node server.....	13
5.1.2	Plans for the social network manager.....	13
5.1.3	Plans for the search engine	13
5.1.4	Plan for the Multimedia Data Manager (MDM)	13
5.1.5	Demonstrations.....	14
5.2	Tools for Engagement and Development	14
5.3	Monitoring progress	16
6	Conclusions	17
7	References	18

1 Executive Summary

1.1 Scope

Delivering an open source platform and building a community around it is one of the major objectives of the SMART FP7 project. The first open source release of SMART is the initial step and the most critical milestone towards achieving this objective. As part of the present deliverable, the SMART consortium has delivered the first open source release as a number of software components, which are open source and can be integrated into a fully working SMART stack representing all the layers of the SMART framework. Furthermore, the consortium has built the infrastructure to distribute the release and to provide comprehensive documentation and support for the users and the developers. The open source release of the SMART project (including all the software components that it comprises) is available for download at: <http://opensoftware.smartfp7.eu/>.

The present document is the report that accompanies the deliverable of the first software release of the SMART open source integrated framework. It describes the contents of the release and the tools built to distribute it and support the users. However, it does not comprise the documentation of the released software, and it does not constitute a user manual either. Instead, the comprehensive documentation is provided on the SMART open source software portal.¹ This documentation is also part of deliverable D6.2, along with the related software and the present report.

1.2 Audience

This document is mainly addressed towards:

- **SMART project members:** The document will provide valuable insights to all SMART project members about the activities within the SMART consortium to release our software as open source and build a community around it. Based on the present document, the members of the SMART project can gain access to summarised information about the open source release of the SMART search framework.
- **The Open Source community:** The open source community can find this document a useful reference to obtain a wider background on the objectives of the first open source release and the management of the SMART open source project. Similar to the SMART project members, open source community members can resort to this document in order to obtain focused summarised information about the open source release, rather than browsing the more extensive documentation (which is available on-line).
- **SMART software developers:** This document will be a useful historical reference for SMART developers, within and without the consortium, to recall information about the first release. We expect that subsequent versions of this deliverable will anyway subsume this initial (yet important) release.

1.3 Summary

This document describes the objectives of the first release and how it contributes to the big picture of building a SMART open source community. It also describes the actual software components distributed in the release. Finally, the document describes a plan for the next releases of SMART and the procedures the consortium have devised to engage with the users and developers, and monitor the progress towards building a SMART community.

1.4 Structure

The document is structured as follows:

- In Section 2, we provide give an introduction where we describe the objectives of the first release.

¹ <http://opensoftware.smartfp7.eu/projects/smart/wiki>



- Section 3 describes the actual components that are distributed in the first release.
- In Section 4, we present the proof-of-concept end-user application that distributed with the release to demonstrate how the full SMART stacks operate.
- Section 5 presents our plans and the procedures we have in place to build and maintain a SMART open source community.
- Finally, Section 6 concludes the document.

2 Overview of the First Release

The major objective of the SMART project is to build an open source framework for multimedia search over social and sensor streams. The SMART consortium has made a significant progress towards developing the various components of the framework. This includes the audio/visual processing algorithms to produce metadata, the metadata storage and streaming API [SMART-D4.1], the reasoning components, the efficient and scalable stream indexing infrastructure and the retrieval models over multiple sensor feeds [SMART-D5.1]. Moreover, these components have already been integrated to build a proof-of-concept application for the live news use case (on the basis of information and data sources associated with the city of Santander which participates as a partner in the project). Although some specialised parts of these components are proprietary code (e.g. selected A/V processing algorithms), all of them will have an open source full or partial implementations, which allows a full stack open source implementation of the SMART framework. Thanks to the “business friendly” SMART MPL licensing scheme; this open source framework can serve as a basis for third-parties to implement added-value applications.

The first open source release aims to provide the public access to the full stack of the SMART framework. In particular, it provides the public with the source code and the accompanied comprehensive documentation to build, install and deploy the various components. Moreover, it offers fully functional deployments (a “sandbox”), hosted by the consortium, where the integrated SMART framework can be tested. The first release is an important initial step towards building a SMART community allowing users and developers to have a feel of what the SMART framework can offer. It also gives them an opportunity to provide feedback to the consortium and the community at an early stage.

In the remainder of the section, we first identify the different user groups of our framework. Then we discuss the objectives of the first release with regards to each group of users.

2.1 SMART User Groups

Capitalising upon the extensive media coverage of SMART, this first software release aims to attract attention of users to SMART and start building the SMART community.

This community comprises of:

- *Application developers* who write their own applications on top of SMART to provide enhanced services to the users.
- *Special interest groups* who need a customised solution for a restricted area and who are detached from the rest of the community.
- *Metadata providers* who run Edge Nodes. Typical example of those include smart cities organisations (e.g., municipalities, end-users), which may attach multiple sensors in the SMART system on the basis of one or more Edge Nodes
- *End-Users* who interact with the system, e.g. by issuing queries to the search engine or through a SMART application that comprises and issues several queries to the SMART system on their behalf.

2.2 Objectives of the First Release

The first release offers features needed for all user groups. This release primarily targets early adopters of the SMART technology within the different SMART user groups.

2.2.1 Application developers

Application developers need access to the API of the search engine, to push queries and pull results from and to their applications. Although they can start building applications with just a handful of Edge Nodes in the community, these applications will only become interesting to end-users once there is a large base of metadata providers. The first release offers the application developers access to a fully

functional implementation of the integrated framework and to the framework itself. In addition, it offers them access to sample metadata. This means that application developers can use the first release to build prototypes and provide feedback at an early stage to the SMART community on various issues such as problems identified, features desired in the upcoming releases and so on. The reception of early feedback is an important element of the agile methodology of SMART, which foresees the iterative releases of its major software deliverables.

2.2.2 Special interest groups

Special interest groups are actually application developers that need a local SMART system for some purpose related to a specific area, for example security systems. Such groups will have limited end-users but with some specific interest and they will most probably be detached from the rest of the SMART community. As for the application developers, the first release will help this group of users to build prototypes for their specialised domain using the SMART framework. They can achieve this by running their own version of the search engine. Similar to application developers, special interest groups may also provide feedback, comments and suggestions regarding the open source SMART software. Such feedback will be invaluable to the SMART consortium as part of the continuous improvement discipline of the project.

2.2.3 Metadata providers

Metadata providers need the Edge Node functionality, as it allows them to feed their data and makes it available for search. This is the most sensitive group: If we manage to attract a large number of providers we get lots of diverse metadata for the SMART system to use and become more interesting. The first release provides this user group with the required infrastructure to stream their metadata and make it available for SMART applications such that the end-users can benefit from this metadata. It also provides them with examples on how to write their own code for connecting their sensors and processing the signals they obtain into useful metadata.

The first release of the SMART software will enable the attraction of metadata providers, thereby leading into the contribution/integration of Edge Nodes into the SMART system. However, we expect that a critical mass of metadata providers will be only attracted following the later more robust and fully functional releases of the SMART open source framework. Nevertheless, the consortium will endeavour to stimulate the interest of possible metadata providers even at this early stage, given that metadata providers can provide real-life distributed datasets that are essential for the development, testing and evaluation of the SMART open source framework.

2.2.4 End-users

End-users are not expected to deploy any software component. They are though the toughest group to satisfy and attract, since an end-user application typically requires a large number of metadata providers to look more attractive to an end-user. End-users have access to the fully-functional integrated implementation of the SMART stack. The first released will allow them to interact pre-installed proof-of-concept demonstrators. This would allow the consortium and the SMART community to obtain an important feedback on the typical usage behaviour of the SMART system (e.g. examples of queries submitted, types of metadata they are interested in). Metadata providers and application developers will be using this input to expand SMART and make it even more useful. Finally, we recognise that the number of end-users can only increase if we attract more metadata providers, since without meaningful metadata from geographically dispersed sources, no query can be answered.

The attraction of end-users will be very important for the sustainability and the exploitation of the open source project. At this early stage however, the attraction of metadata providers and application developers is undeniably of higher priority for the SMART consortium.

3 Components of the Release

In this section, we describe the actual software components that are distributed in the first release. Each of these components is hosted on the SMART software portal [SMART-D7.4]. The source code of each component is hosted on the Git repository of the portal which is currently restricted for access to developers within the SMART consortium. However, the sources and the binaries can be downloaded via the SMART portal upon completing a survey as discussed later in Section 5.2. A comprehensive documentation is provided on the project's Trac portal.² It covers all aspects of the release for the different user groups described in Section 2.2. The documentation allows users to learn about the SMART framework and provides instructions to install the various components. The first release will be made public under the MPL license 2.0 which is chosen among other licenses due to its compatibility with background libraries used, while ensuring possibilities of commercial exploitation of the SMART open source platform [SMART-D7.4].

In this section, we describe the components of the first release and how they can be integrated into a full SMART stack. However, in this description, we are not providing the documentation or the manual for using, installing and extending those components. We redirect the reader to the SMART open source portal if they are interested in that documentation and relevant details.

3.1 The Edge Node Server

The release comprises components for the Edge Node server, which is responsible for *handling* and *generating* metadata. The mechanism in SMART, by which metadata is collected into and streamed out of the Edge Node database, is called *feed handling*. The components that generate metadata by processing raw sensor data streams are called *perceptual components*.

3.1.1 Feed handling

Feed handling requires a few external components to be installed. These are detailed in the accompanying documentation, found at <http://opensource.smartfp7.eu/projects/smart/wiki/Installation>, together with installation instructions.

Having set up the system, metadata is handled via one of two interfaces: (1) the *direct interface* and (2) the *common interface*. The direct interface uses the underlying HTTP streaming interface of the Edge Node database (CouchDB). To add control on what and how it is being streamed, we have built the common interface component. This component provides a database independent API. It is distributed within the release in the project SMARTEdgeNode. Both the Java source code and a WAR (Web Archive) executable are being distributed.

3.1.2 Perceptual components

The perceptual components are used as interfaces to sensors. They process the sensors signals and provide understanding their context. They are installed into Edge Nodes by their managers and generate metadata that populate the respective feeds. Note that it is beyond the scope of the project to provide the Edge Node managers with perceptual components. Nevertheless, some baseline visual processing components are included in this release. They are written in C/C++ and are located within the SampleClients/C_CPP folder of the release. The release contains the source code, Windows executables and MS Visual Studio 2010 projects. The perceptual components provided are:

- **simple_camera**: This project expects a USB camera connected to the computer and calculates the average frame intensity and the frame-by-frame difference. It serves as an example on how to interface the camera and how to populate feeds within C. It requires the OpenCV (for processing) and the curl (for communicating) external libraries.

² <http://opensource.smartfp7.eu/projects/smart>

- **blob_tracker**: This is another visual perceptual component. It tracks moving objects (people, vehicles, etc.) in successive video frames and reports bounding boxes as a metadata stream. It has the same 3rd party library requirements.

3.2 The Social Network Manager

Part of the Edge Node's imported data comes from multiple social networks. For the extraction and fusion of this data, SMART Edge Nodes incorporate a component called "Social Network Manager".

Using specialised drivers, social networks (Twitter, Facebook) have been integrated into the SMART system as another source of information and each of them is considered another sensor for SMART. Using a REST API back-end and the specific drivers, the Edge Node can stream social posts, from supported social networks, which are filtered to contain certain words or phrases. These posts enrich the knowledge the edge node has about events that may be happening in the real world.

Extension capabilities are given for further development of the component in the form of an interface that guides but not limits the development of new drivers for other social networks as well as new custom filters.

3.3 The Search Engine

The components for the search engine released will offer a fully functional implementation of the SMART search engine. The search engine part of the release can be integrated with its remaining parts. It can be run to index feeds from multiple edge nodes and allows application developers and users to interact with the search engine, i.e. issue queries and retrieve results.

The first release includes an open source version of SmartReduce, the real-time distributed scalable infrastructure to index and retrieve streams [SMART-D5.1], which is built using the open source Terrier³ retrieval platform and the open source Storm⁴ framework for distributed stream processing. Within SmartReduce, Terrier has been extended to operate with in-memory index structures rather than on-disk indices which are not suitable for real-time search. Moreover, the release includes one implementation for event retrieval models within SmartReduce that are used to answer queries by examining evidence from multiple social and sensor feeds originating from the edge node servers. Finally, the RESTful interface of the search engine (the Search API) is also provided in the release to allow access to the search engine by end-users and developers.

The search engine is released as two separate Java projects, namely SensorSearch and SearchAPI. In the following we describe these two

- **SensorSearch**

This project contains the source code for the open source version of SmartReduce. It also includes a sample implementation of the specialized event retrieval models within SmartReduce. Since this is only the first release, this version of SmartReduce can only be run locally and cannot be run on a Storm cluster. Developers who are interested in running SmartReduce on a production Storm cluster need to make the required changes.

Full documentation for building the source code and running SmartReduce are provided on the SMART Trac portal.⁵ Application developers can install SmartReduce and connect to existing edge node servers with a simple configuration procedure described in the documentation.

The source code is written in Java 1.6 and can be directly open with the Eclipse IDE. All the dependencies (the 3rd party libraries) used are also distributed in the 'lib' folder of the project.

³ <http://terrier.org>

⁴ <https://github.com/nathanmarz/storm>

⁵ <http://opensource.smartfp7.eu/projects/smart/wiki/InstallingSearchEngine>

This includes the real-time extension we have developed for Terrier as described in [Smart-D5.1], for which the source code is not distributed in the release.

- **SearchAPI**

This project contains the source code for the Search API. The Search API provides an interface to SmartReduce. Application developers can run the API after configuring it to connect to SmartReduce. The API can be then used to issue requests to the Search API and obtain results in a JSON format.

The source code is written in Java 1.6 and can be directly open with the Eclipse IDE. All the dependencies (the 3rd party libraries) used are also distributed in the 'lib' folder of the project. Full documentation for building the source code and running the SearchAPI are provided on the SMART Trac portal.

3.4 Third-party Components

The section describes the major third-party software that is required to run the various components of the release, which are all open source. The purpose here is not to list all these dependencies, which is not plausible anyway, but we want to highlight the major platforms/libraries we are relying on. Moreover, it should be noted that some of the third-party software is distributed within the release and some are not and need to be downloaded and installed separately. All our dependencies have MPL compatible licenses, which are not “viral”, i.e. they make no demands on releasing the software using their original license.

In the following, we list the major third-party open source software required to run the SMART framework:

- **CouchDB⁶**: is a database server, developed by the Apache Foundation⁷ and released under the “Apache License”, which was created specifically for web applications. It is a NoSQL database that uses JSON to store data and stream it using an HTTP interface. CouchDB is the backend used by Edge Node server to store and stream feeds of metadata.
- **Apache Tomcat Server⁸**: is an open source implementation of the Java Servlet and JavaServer Pages technologies. It offers the container to build java web based technologies. Like CouchDB it is developed by Apache and released under the “Apache License”. The Edge Node server has an interface to handle meta-data feeds which runs as web application within Tomcat.
- **OpenCV⁹**: Open Computer Vision is a C/C++ library available for various operating systems. It offers an open source implementation for a number of computer vision algorithms. This library is required by some of the perceptual components for processing videos from cameras within the Edge Node. Note that OpenCV provides a basis for video processing at the Edge Node. We expect however that application developers and deployers of the SMART open source framework may opt for integrating their own (possible proprietary) video processing infrastructures. The integration of OpenCV into the SMART open source release is meant to provide an example of the potential of the SMART search system, along with an illustration of the video processing concept within SMART.

⁶ <http://couchdb.apache.org/>

⁷ <http://www.apache.org/foundation/>

⁸ <http://tomcat.apache.org/>

⁹ <http://opencv.org/>



- **Storm**¹⁰: is an open source distributed real-time framework for processing unbounded streams of data. It defines the primitives for parallel real-time processing of data streams and ensures guaranteed processing of all incoming data items. Storm is released under the EPL license, which is an MPL compatible license. SmartReduce, the backbone of the SMART search engine, is built on top of the Storm framework.

¹⁰ <https://github.com/nathanmarz/storm>

4 **Description of the PoC (Proof-of-Concept) Application**

The first release includes a command line search application that allows an end-user to interact with the SMART search engine. It is an example of a fully functional SMART stack. It is a proof-of-concept application since it demonstrates how technically the various components of the SMART framework can be integrated into a software that brings an added value to an end-user.

In particular, the command line search application allows an end-user to enter a keyword query (e.g. crowd, traffic, blue, etc.), and it sends the query via the search API to the SMART search engine. The application can be configured to use an arbitrary Search API. By default, it will use the search sandbox which is connected to the Edge Node sandbox indexing multiple continuous feeds of crowd analysis. Upon receiving the results from the Search API in JSON format, the application parses the results and presents them to the user in a human readable format. In particular, the application displays as a response to a query a ranked list of up to 10 event results. Each result item (event) contains the following:

- 1- The location where the event happened.
- 2- The time of the event
- 3- The crowd level on that time

This application is distributed in the first release as a Java project 'SmartCMD':

- **SmartCMD**

This project contains the source code for the command line search application. The source code is written in Java 1.6 and can be directly open with the Eclipse IDE. The project has a single dependency, the Gson library for parsing JSON, which is also included in the 'lib' folder of the project.

Full documentation for building the source code and running SmartCMD are provided on the SMART Trac portal.¹¹

The PoC application is provided as a very simple sample application that enables end-users and application developers to test the components of the first release. The SMART project work plan foresees the development of complete applications in the areas of live news and security/surveillance as part of deliverable D6.1 of the project.

¹¹ <http://opensource.smartfp7.eu/projects/smart/wiki/CommandLineSearch>

5 **Building a SMART Community**

In order to sustain an open source project, a community needs to be built around it. Therefore, building and growing the SMART community is an important issue for the success of the SMART project. In this section we describe our plan for the next SMART releases and how they support growing the SMART community. Note that future plans will aim at improving both the user-friendliness and the developer-friendliness of the open source project, as means to facilitating engagement of both users and contributors. Apart from future plans, we also describe the procedures we have in place to manage and engage with the community. Finally, we discuss how we monitor the progress of growing the community

5.1 **Plan for next releases**

In this section, we describe our plans for the next open source release. We split our discussion to cover each component separately.

5.1.1 **Plans for the edge node server**

The next release will have improved versions of the common interface for feed handling and many more baseline perceptual components, including a speech activity detector, a face tracker and a crowd analysis system. These enhancements are expected to motivate the engagement of users and developers, given that they will be offered with a larger number of sensors and data feeds, which will be readily available for integration in applications, use cases and demonstrators.

5.1.2 **Plans for the social network manager**

The following release will feature mainly usability enhancements for ease of use and visual improvements as well as new additions of social media in the supported list of the component. The provision of support for the most popular social networks (such as Twitter, Facebook and FourSquare) is expected to boost the engagement of users and developers, given that they will be able to leverage their accounts in social networks towards implementing interesting applications.

5.1.3 **Plans for the search engine**

The following lists some of the features and components that will be available in the next open source release of SMART:

1. Retrieval models for events for multiple streams: the next releases will reflect on the progress of the activities in T5.4 (query scoring and anticipation). It will contain improved retrieval models which will have by then been evaluated on larger scale of metadata streams. The higher effectiveness of these models, the more confidence the users and the developers will be with the SMART technology.
2. Support for running queries, which is currently not part of the release. Running queries refer to the case where the query has been submitted but since then new relevant events may develop and the user should be notified with those relevant events (information filtering problem). Including the information filtering models would open up opportunities for application developers to build new types of SMART applications. Moreover, it fosters more engagement with the end-users who will be notified with results even if they are not issuing queries to the system.
3. Efficient indexing and retrieval: the next releases will also reflect on the progress made in T5.2 "Smart Reduce" Engine and will include improved efficient techniques for partitioning the index.

5.1.4 **Plan for the Multimedia Data Manager (MDM)**

The next release will feature an integrated version of the Multimedia Data Manager. For suitable events, audio/visual data will be selected, adapted to several formats and stored. The required URL to access the media will be integrated in a data feed, so links to the multimedia content can be retrieved from the search engine and presented to the client applications.

Note that the MDM implementation will broaden the scope of functionalities that are supported by the SMART platform, while also strengthening the (multi)media dimension of the project. In this way, it boosts the broadening of the SMART community of stakeholders [SMART-D2.1] and encourages at the same time the participation of more end-users and contributors.

5.1.5 Demonstrations

In order to attract the interest of application developers, end-users and the open source community at a large scale, an open source project should provide demonstrations (ideally one-click demonstrations) that are easy-to-download, easy-to-install, easy-to-deploy and easy-to-use. In this direction, the SMART project aims to complement future releases with interesting motivating demonstrations that illustrate the concept of the SMART search framework, as well as its functionalities. In terms of functionalities, these demos will cover:

- The ability to integrate and search feeds from popular social networks such as Twitter.
- The ability to integrate and search feeds from non A/V sensors (such as meteorological stations).
- The ability to integrate and search feeds from A/V sensors and perceptual components (including video and audio processing components).
- The ability to reason over multiple data feeds stemming for A/V perceptual components, non A/V sensors and social networks.

In addition, the demos will reflect on the outcome of the deliverable D5.4 by incorporating the developed visualisation and web 2.0 mashup libraries, which would make them more appealing to end-users.

5.2 Tools for Engagement and Development

We will continue to improve our open source portal to devise more tools for engaging with the SMART community. In the following, we list some of the tools we already have in our portal and discuss how they are useful for engaging with the SMART community:

- **Documentation:** Speaking in general terms, one of the main barriers to get involved in an open source project is the low quality of documentation. The consortium is trying to overcome this problem, so we have put great efforts in documenting all the aspects of SMART (currently, more than 130 wikipages of documentation). Moreover, the documents in SMART are in a wiki, so the users can collaborate interactively in writing or correcting the documentation.
- **Git:** the Git repository allows the developers within the SMART community to share code and collaboratively develop the various components of the framework.
- **Code browser:** Git is a good tool for development, but it is neither “visual” nor a “web application”. The code browser allows accessing the code stored in the Git repository graphically from a web browser. Using the code browser, users can examine the code from the web portal, without actually downloading the code or installing Git.
- **Tickets:** The open portal features a ticketing system for reporting issues to developers such that they can be fixed and kept track of for reference in the future.
- **Milestones:** Some project management tools are available. For instance, it is possible define milestones, and to mark some bugs corrections and features as objectives for a milestone.
- **Forums:** a very valuable tool in order to help less experienced users and construct a community are the forums. In the last months, we have added the forum functionality to the portal.

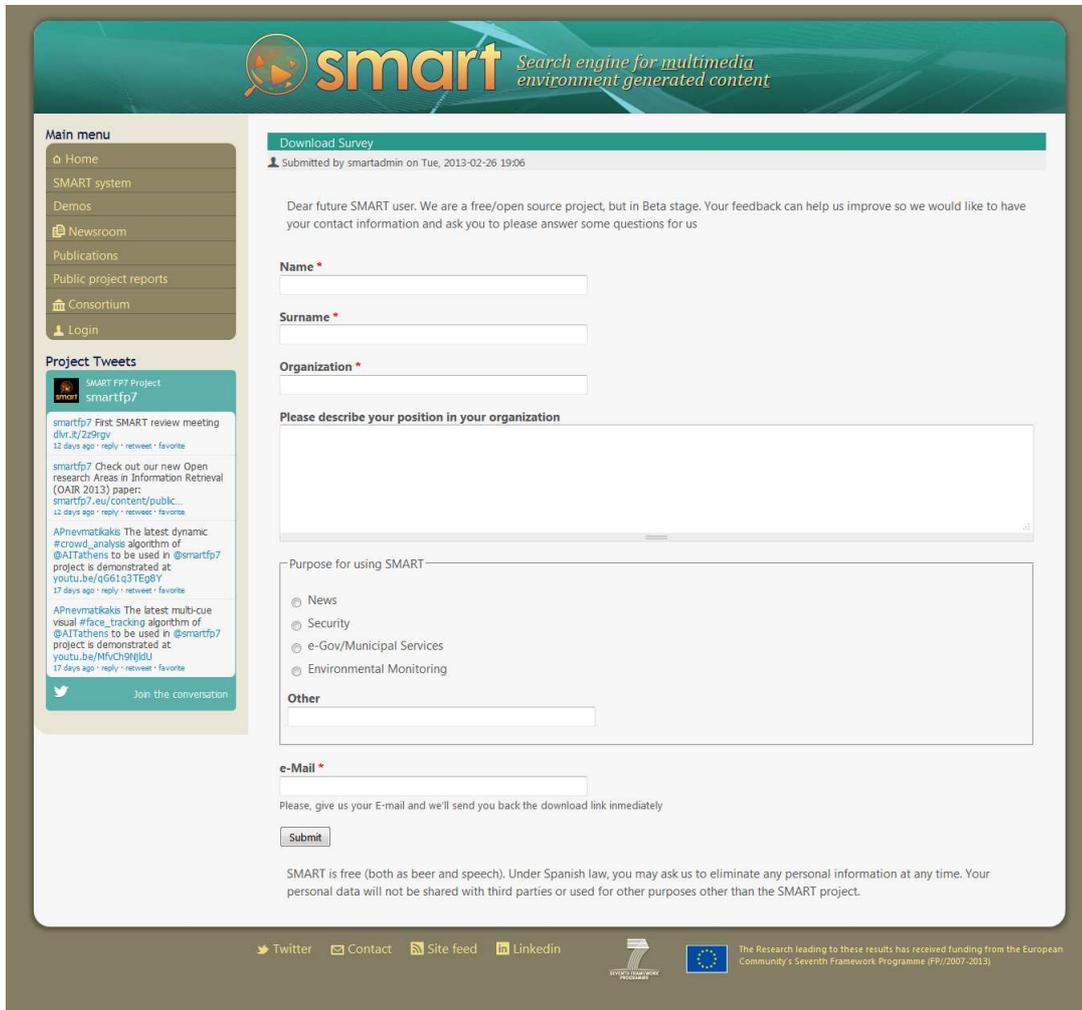
The above tools provide a baseline set that could be used by the SMART community. In future releases we will investigate additional tooling support in order to facilitate installation, deployment and use. These tools will be for example targeting: (a) the management of dependencies between SMART and third-

party libraries, (b) the integration with integrated development environments (e.g. Eclipse) and related visualisation tools.

Some parts of the portal are currently restricted to registered users. Until now, the registered users were the developers from the consortium members. After the public release of the first version, we plan to add external developers who can involve in the development process. Finer-grained permission policy will be implemented, as this is a common practice in these environments. Non critical parts like the forum will be open for all, whereas critical aspects like writing access to the code will be kept restricted.

A new section in our web site is under construction, where all the active Edge Nodes contributed by SMART partners and the public will be shown on a map, together with information on what sort of metadata they are producing. This will help our potential users to understand how SMART can be useful to them, especially when the map starts filling with edge nodes, our live information sources.

Finally, in order to keep in contact with the people interested in SMART and construct the SMART community and database (as it is explained in section 5.3), we have created a survey¹² for downloaders who are wishing to obtain the first release (Figure 1). After filling in the survey, we send the download link to the person interested.



The image shows a screenshot of a web browser displaying a survey titled "Download Survey" on the SMART website. The page has a green header with the SMART logo and the tagline "Search engine for multimedia environment generated content". On the left, there is a "Main menu" with links to Home, SMART system, Demos, Newsroom, Publications, Public project reports, Consortium, and Login. Below the menu is a "Project Tweets" section with several tweets about SMART FP7 project updates. The main content area contains the survey form, which includes a greeting, a "Name" field, a "Surname" field, an "Organization" field, a text area for "Please describe your position in your organization", a section for "Purpose for using SMART" with radio buttons for News, Security, e-Gov/Municipal Services, and Environmental Monitoring, and an "Other" text field. There is also an "e-Mail" field and a "Submit" button. At the bottom, there are social media links for Twitter, Contact, Site feed, and LinkedIn, along with logos for the European Union and SMART FP7 project.

Figure 1 The survey for downloaders

¹² <http://www.smartfp7.eu/content/download-survey>

5.3 Monitoring progress

Following this initial open source release, the SMART consortium will actively monitor and keep track of users and contributors to the open source project. The development of a database of users, contributors and other interested parties will be a key for maintaining and sustaining the project's open source community. Apart from monitoring the evolution of the community (e.g. in terms of its size, composition, expertise and more), the progress monitoring will extend to the specific actions undertaken by the members of the SMART open source community (e.g. integration of new Edge Nodes, development/integration of use cases, development/integration of new demos, development of new tools, using SMART in the scope of other projects and research initiatives and more). In addition to reporting the activities of the SMART community (e.g. using SMART in the scope of other projects), the consortium will take advantage of the SMART database in order to update the community members about new releases, new developments, improvements on existing components, plans & roadmaps, news, announcements and more. In this way, the project will attempt a more active and continuous engagement of stakeholders and interested parties.

As it was stated in section 5.2, we expect to create the community around the developers section in the web page (opensoftware.smartfp7.eu). By now, the developer's page is used mainly by consortium members, but after the public release, it will be important for the community. A good indicator for the success can be the number of visits to the page, especially to the pages regarding the information for setting up new nodes and creating applications around SMART.

Monitoring the developers' page raises some difficulties, because the web page is not created directly. We (and the future developers) create the documentation, tickets, forum posts etc using Trac. Currently, we use Google Analytics for monitoring the rest of the website, but the integration of Trac and Google Analytics is not straightforward. However, since Trac is an application over the Apache Web server, it is possible to monitor the access to the different pages in the developers' website using one of the many tools created for Apache Web Server analysis, like Apache Log Viewer.

Finally regarding source contributors, the project will adhere to the (master-planning) governance scheme illustrated in [Smart-D7.4]. The initial planning/roadmap is briefly outlined in earlier paragraphs. Third party contributions will be organized/structured in-line with this governance scheme as well.

6 Conclusions

The SMART consortium has made the first step towards building an open source community, which is delivering the first release of the open source SMART integrated framework. On the basis of this integrated framework, the consortium will attempt to attract a wide range of early technology adopters, including application developers, end-users and special interest groups. These stakeholders could provide invaluable feedback that will be used to improve subsequent releases of the SMART software. In this respect, the open source release of the project is a very critical milestone within the project's wider roadmap.

In this document, we provided a comprehensive accompanying report on the first release of the SMART open source integrated framework. In particular, we identified the various SMART user groups and explained our objectives for each user group. Furthermore, we described the released components and the third-party software that is used to build these components. Finally, we devised a plan the next releases of SMART and the procedures for effective monitoring and engagement to sustain and grow a SMART open source community.

Subsequent versions of the SMART open source framework are expected to provide more sophisticated functionalities, while at the same time being more robust. Furthermore, as part of deliverable D6.1, application developers within the SMART consortium (i.e. Atos, PRISA, S3LOG) will integrate the present release of the SMART open source framework into more sophisticated applications associated with live news and security/surveillance use cases. The experience of developing these applications will also provide insights in fine-tuning and enhancing the open source SMART framework as part of its subsequent releases. This is yet another factor that indicates the high importance of this first release in the scope of the wider roadmap of the SMART open source project. Overall, the release of SMART results as open source paves the ground for continuous improvement, but also for the sustainability and the wider dissemination of the SMART research results.

7 References

[SMART-D2.1] SMART FP7 consortium. Deliverable D2.1, "Detailed Report on Stakeholders Requirements", 2012

[SMART-D4.1] SMART FP7 consortium. Deliverable D4.1, "SMART Distributed Knowledge Base and Open Linked Data Mechanisms", 2012.

[SMART-D5.1] SMART FP7 consortium. Deliverable D5.1, "SmartReduce Engine", 2012.

[SMART-D7.4] SMART FP7 consortium. Deliverable D7.4, "Open Source Software Portal", 2012.