SEVENTH FRAMEWORK PROGRAMME
Networked Media

Specific Targeted Research Project

SMART

(FP7-287583)

Search engine for MultimediA environment generated contenT

D2.1 Detailed Report on Stakeholders Requirements

Due date of deliverable: 30-04-2012

Actual submission date: 21-05-2012

Start date of project: 01-11-2011 Duration: 36 months

Summary of the document

Code:	**D2.1 Detailed Report on Stakeholders Requirements**
Last modification: **State:**	09/05/2012 Final
Participant Partner(s): **Author(s):** **Fragment:**	ATOS, AIT, IBM, Imperial, S3LOG, TELESTO, UG, PRISA, SDR Aristodemos Pnevmatikakis (AIT) Antonis Litke, Dimitris Drakoulis, Antonis Politis (Telesto) C. Macdonald, I Ounis (GLA) Josemi Garrido (Atos) Tomás García Fresno (SDR) Regis Riveret (Imperial College) No
Audience:	☒ public ☐ restricted ☐ internal
Abstract:	*This document is the SMART Deliverable 2.1.*
Keywords:	SMART Stakeholders, Requirements analysis, Functional requirements, non-functional requirements.
References:	DoW

Table of Contents

1 Executive Summary

1.1 Scope

The main objective of the SMART project is to build an open source multimedia search engine, which could provide scalable search capabilities over environment generated content i.e. content captured by the physical world via sensors. A main part of the project will be allocated to the processing of multimedia content derived from visual and acoustic sensors (notably cameras and microphones). The purpose of this processing is to extract pieces of information about the surrounding environment of the sensors based on leading edge audio/visual (A/V) processing components. Such components will be employed in order to allow the SMART systems to perceive the status of the surrounding environment and accordingly to make this context available to the search engine for (later) indexing and retrieval, as required by the SMART applications.

A key prerequisite for the design of the SMART system is the elicitation of requirements from the stakeholders that comprise its value chain. The purpose of the present deliverable is to elaborate on the requirements analysis as far as it concerns the desired functional and non-functional requirements as defined by the stakeholders.

1.2 Audience

The primary audience of this document consists of the people that will participate in the design of the SMART system. Primarily, the audience concerns members of the consortium who need to undertake the tasks of designing the components and modules of the SMART system. Additionally, the document is of wider interest to stakeholders that are active in multimedia search initiatives, including researchers participating and contributing to FP7 projects under the «'Networked Media and Search Systems» objective

1.3 Structure

The document is structured as following:

Section 2 provides the introduction of the deliverable.

Section 3 describes the requirements analysis methodology that has been followed for the elicitation of the requirements and the rationale behind it.

Section 4 gives an overview of the SMART system definition along with the stakeholders and how they form a consistent value chain.

Section 5 provides an overview of the project background of each partner involved in the consortium and its positioning with respect to the SMART value chain.

Section 6 presents shortly various other approaches from the state-of-the art technologies and projects that are currently running or have finished.

Section 7 is providing an analysis of the functional and non-functional requirements as elicited by the project partners.

Section 8 provides a consolidated and coded view of the requirements that have been derived in section 7.

Section 9 concludes the document.

2 Introduction

The main objective of the SMART project is to build an open source multimedia search engine, which could provide scalable search capabilities over environment generated content i.e. content captured by the physical world via sensors. A main part of the project will be allocated to the processing of multimedia content derived from visual and acoustic sensors (notably cameras and microphones). The purpose of this processing is to extract pieces of information about the surrounding environment of the sensors based on leading edge audio/visual (A/V) processing components. Such components will be employed in order to allow the SMART systems to perceive the status of the surrounding environment and accordingly to make this context available to the search engine for (later) indexing and retrieval, as required by the SMART applications.

A key prerequisite for the design of the SMART system is the elicitation of requirements from the stakeholders that comprise its value chain. The purpose of the present deliverable is to elaborate on the requirements analysis as far as it concerns the desired functional and non-functional requirements as defined by the stakeholders.

In order to have a consistent requirements analysis it is important to define the methodology that will be followed in order to come up with a consolidated requirements list that will be used to drive the design of the SMART system and its final evaluation. Furthermore, stakeholders' requirements will be taken into account in the development of the SMART architecture as part of later deliverables of WP2.

The present deliverable is devoted to the description, and documentation of key requirements associated with the development of the SMART search engine and associated open architecture for systems that search over repositories of environment generated content. The purpose of the deliverable is to present an analysis of the SMART requirements from the stakeholders' perspective. The deliverable will eventually lead to the overall specification of the SMART use cases that will be presented in the deliverable D2.2 SMART Use Cases Specifications.

3 Requirements analysis methodology

This section gives an overview of the requirements analysis methodology that will be followed in the SMART project. The following figure gives an overview of the approach.

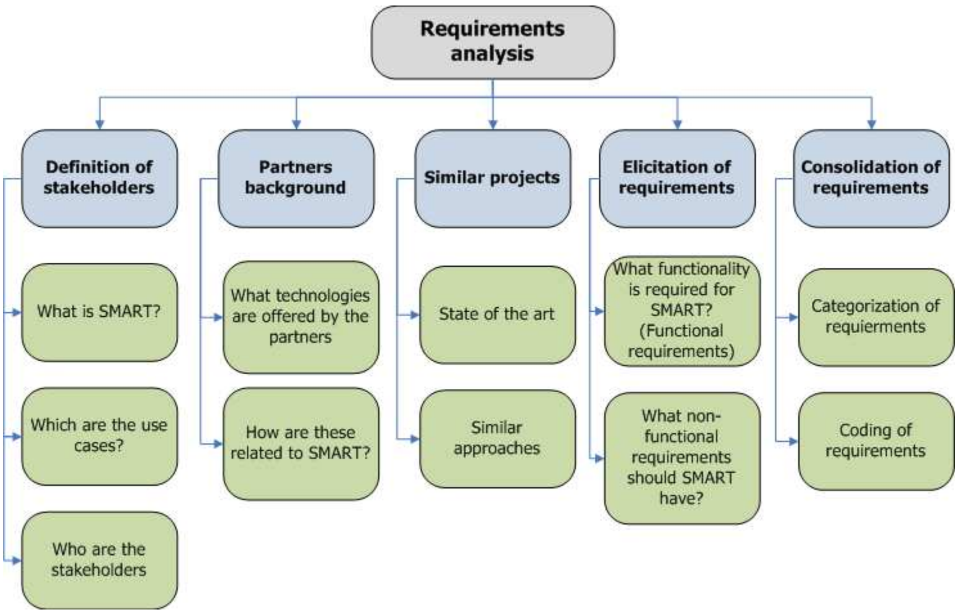

Figure 1: SMART requirements analysis approach.

The analysis starts with a definition of the SMART concept, including an overview of the use cases and the stakeholders involved so as to understand the context of the project and identify the various stakeholders. As a next step, the consortium partners give an overview of the technologies that are going to be involved in the project and the perspective of using them in order to implement the SMART concept. Similar projects and background from previous projects are also mentioned in order to present the state of the art and the previous achievements that can be used as a starting point. In the sequel the elicitation of requirements is derived from the definition of the desired functionality of the SMART system given from the perspectives of the functional components and the non-functional attributes that the end system has to expose. Finally, the requirements are gathered and consolidated into one list, grouped and coded accordingly in order to comprise the reference for the design, implementation and validation phases of the project.

4 SMART definition and stakeholders

This section provides a description of the SMART development platform and the search framework, so as to define the major technological components of the SMART platform and identify the stakeholders who are part of the SMART value chain.

4.1 SMART definition

The main goal of the SMART project is to research and implement a scalable open source next generation multimedia search engine that will be able to search information stemming from the physical world. The SMART multimedia search engine will be able to answer queries based on the intelligent collection and combination of sensor generated multimedia data. The matching of the queries with sensor and sensor processing algorithms (notably audio and video processing algorithms) will be based on the sensors' context and metadata (e.g., location, state, capabilities), as well as on the dynamic context of the physical world as the later is perceived by multimedia processing algorithms (such as face detectors, person trackers, classifiers of acoustic events, crowd analysis components and more). Furthermore, SMART will be able to leverage social network information in order to facilitate social queries over a multitude of sensor data.

The SMART engine will be able to answer to high-level queries by synthesizing results stemming from distributed sensors, in a scalable and intelligent way. Users of the SMART multimedia search engine will be able to submit their queries through high-level semantics and accordingly access a set of answers from all the sensors and sensor processing components that can contribute to the submitted query. Hence, towards answering a query, SMART will enable the creation of autonomous societies of collaborating networked sensors and multimedia processing algorithms, which can effectively add value to the results of the target query. With the formulation of such sensor societies, SMART will devise mechanisms (based on Web2.0/Web3.0 mashups) towards reuse the results of the queries across multiple application contexts.

SMART will be designed as an open framework for intelligently accessing environment generated content, while at the same time intercepting this content with information stemming from the emerging Web2.0 based social networks (such as Twitter and Facebook). In addition to the specification of an open framework for environment generated content, SMART will provide an open source implementation of the main components of the framework, with a view to achieving wide adoption and impact creation. The SMART open source implementation will be built on top of the Terrier (terrier.org) open source engine, which is supported/maintained by one of the project partners (GLA).

The main characteristics of the SMART search engine will be the following:

- Open and Open Source: SMART will be designed as an open framework, which will be extensible in terms of sensors (e.g., camera, microphone arrays, WSNs), ontologies and semantic structures (e.g. multimedia ontologies, sensor ontologies,), as well as multimedia processing components (notably video and audio processing algorithms). Furthermore, SMART will be to a large extent implemented based on open source technologies and royalty free standards. For example, the main components of the SMART engine will be implemented in the form of open source software over the Terrier.org search engine. SMART will attempt to create an open source community for sustaining and evolving the project‟s results beyond the end of the project.

- Multimedia: The SMART search engine will enable query answering based on the real-time processing of multimedia data stemming from the physical environment (such as audio and video). To this end the project, will research and adapt cutting edge audio and visual processing components, notably in the areas of acoustic event classification and visual scene analysis. These components will be used for the SMART proof-of-concept validation. However, SMART will be extensible in terms of audio and visual processing algorithms, beyond the components to be developed in the project.
- Participatory and Reusable: The very same sensor and multimedia processing algorithms will be able to contribute to multiple concurrent queries of the SMART system. Participatory sensing schemes will be researched along with ways of caching data and queries, while also dealing with mobility and sharing application contexts. Furthermore, a number of Web2.0/Web3.0 mashups will be implemented to allow reuse of sensor queries across multiple applications and searches.
- Smart and anticipatory: Based on machine learning and/or rule-based mechanisms, SMART should be able to anticipate the answers to certain queries. This will empower a level of intelligence, beyond self-learning and ranking algorithms used by existing search engines.
- Social: The SMART search engine will seamless leverage information and search results from (Web2.0) social networks (such as Facebook and twitter) in order to facilitate the interception of social networks with sensor networks, towards social applications and searches of environment generated content.
- Scalable and Dynamic: SMART will be designed to be scalable at internet scale. Hence, the project will research a scalable architecture for collecting, filtering, processing, caching and combining sensor data in a highly heterogeneous and distributed environment. Furthermore, SMART will dynamically provide up to date information sensed by the underlying sensor networks. To this end, it will deal with the changing context of sensors (e.g., in the scope of mobility scenarios).
- Context-aware: SMART will enable the context-aware orchestration of sensor data and metadata towards accessing data that pertain to a given context. To this end, metadata associated with time, space, location, goals, tasks and more will be used in order to trade/negotiated the contribution of a sensor to a particular query. To this end, the project will research sensor selection protocols/algorithms, along with collaborative protocols enabling the orchestration of sensors towards a joint task.

4.2 SMART uses cases overview

This Section provides an overview of the uses cases that will be deployed in the SMART project, namely Live News and Security & Surveillance in urban environments. The SMART use cases will in general complete applications comprising multiple configurable and dynamic queries over the SMART engine. They will combine multiple search queries into composite applications. Hence, they will comprising most of the technological elements/components of the SMART search engine including audio processing components, visual processing components, sensor edge servers deployed at multiple physical points within the smart city, identification of search terms in queries, scalable distributed processing/indexing/retrieval across multiple information/sensor sources, mashups for visualization of information and more. The SMART use cases will be deployed in the smart city of Santander, where cameras, microphones and other sensors will be deployed, with a view to validating all the above components in a realistic environment.

4.2.1 Live News

The objective of the live news use case is to create a container for local news and general information about the city of Santander. This container could be a website where a Santander’s citizen can find information and statistics about what is happening (and happened) around him, e.g. in his neighbourhood or his city. The objective is to build it using:

- Automatic event extraction by processing the streams from the deployed sensors.
- Social networks conversation, including user comments on Eskup.
  - City hall events agenda.
  - Breaking news from Eskup (from the ElPais Eskup's journalists).
  - News coming from other local news RSS feeds.

The data planned to be included on the live news use case container as well as the technologies required to extract these data are:

Data	Technologies involved
Local breaking news	Search Engine, Inferred news form sensors
Events registered	Search Engine, Audio and video processing of the streams from the deployed sensors (Crowd Analysis – Incident detection)
Color tendencies	Video processing (Color Analysis)
Social network conversation density	Search Engine
Filtered social conversation based on user's location and social networks	Search Engine
Relevant traffic incident with links to the traffic cameras of the city hall	Video & Audio Processing; Search Engines
Crowds detected on specific locations with inferred cause	Video Processing; Audio Processing
News and events published by Journalists and reliable sources using a Voice enabled Eskup voice mail	Audio Processing (Voice recognition and transcription)

Requirements Summary:

- *Social Conversation: Extraction, Filtering, Statistics, Geolocalization, Density, Sentimental analysis.*
- *Event recognition from video and audio data.*
- *Crowds and events extraction from sensors and Social Networks including inferred cause.*
- *Geolocalized color tendencies.*
- *Eskup Audio Input Voice Message: Audio Recognition and transcription.*

4.2.2 Security and Surveillance in urban environments

An important application for the SMART project is the automatic analysis and detection of abnormal events in public places or during public events. One particular class of public security issues are those involving a large number of people gathering together (crowding), such as in public assemblies, sport competitions, demonstrations (e.g., strikes, protests), etc. Because of the high level of degeneration risk, the security of public events involving large crowd has always been of high concern to relevant authorities.

Challenge for this experimentation is automatic detection of abnormal crowd behaviours during public events. We have divided crowd behaviour analysis into three tasks:

- *Task 1: motion information extraction;*
- *Task 2: noise information extraction;*
- *Task 3: abnormal behaviour modelling.*

We are looking for crowd behaviour such as:

- *flow divergence and convergence in the region of interest;*
- *emergence of a new crowd flow from an existing crowd flow having different characteristics;*
- *sudden changes in the crowd motion;*
- *people moving in counter direction;*
- *erratic motion beside the main crowd flow (as in the case of fighting);*
- *sudden change in the noise of the crowd.*

4.3 Stakeholders value chain

The following figure gives an overview of the SMART value chain.

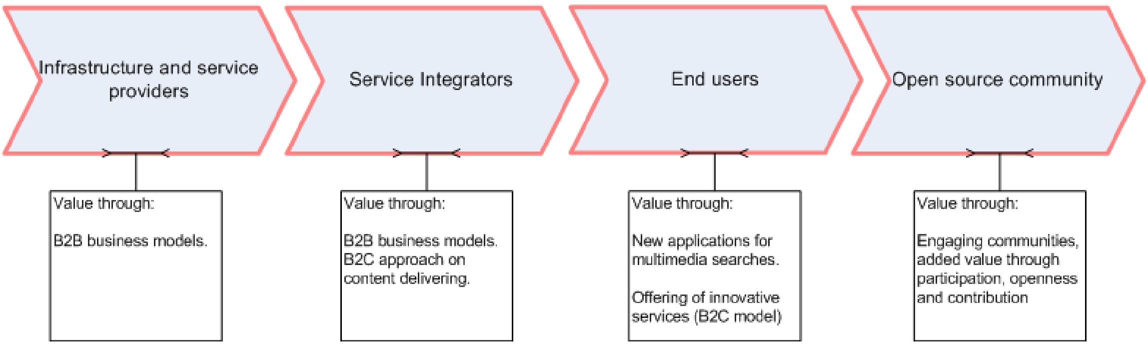

Figure 2: The SMART value chain.

Note that SMART consortium includes all necessary stakeholders of the SMART value chain. In particular, the consortium includes infrastructure providers (i.e. SDR), service providers (i.e. PRISA), application/service integrators (ATOS, S3LOG, TELESTO, IBM, AIT). This will allow for a credible validation of the SMART concept, along with different deployment configurations and services operations plans.

Following paragraphs illustrate the role of the various stakeholders in the SMART value chain.

4.3.1 Infrastructure Providers

Infrastructure Providers: also called also Environment Content Providers in the SMART context (e.g., municipalities, citizens, building owners), they provide their sensing infrastructure and environment generated content to the SMART search engine and ecosystem. Infrastructure Providers will offer or lease their infrastructures in exchange of some cost/fee, or based on other participation incentive.

4.3.2 Service Providers

Application Service Providers (ASPs) (e.g., Search Engines, Portals): ASPs will offer search services over the infrastructures of one or more infrastructure providers. To this end, ASPs will establish a number of Service Level Agreements (SLAs) with infrastructure providers and/or environmental content providers. In addition to agreements with infrastructure providers (such as smart cities and municipalities), ASPs could also provide incentive for citizens to participate in the platform, i.e. in order to provide access to citizen generated content (e.g., generated from mobile phones) ASPs will also manage user information to distinguish between premium users and final users. Also authorization for different services could be required and implemented. ASPs will benefit from the open source nature of the SMART platform, in order to integrate SMART and provided added-value sensor based search services. ASPs will endeavour to generated revenue streams based on subscriptions/fees of corporate end-users or individuals.

4.3.3 Service Integrators

Service Integrators and ISVs (Independent Software Vendors): The role of service integrators and ISVs in the SMART value chain is associated with the integration of the platform (on behalf of an ASP), as well as in the enhancement of the smart framework with additional components with a view to augment the services offered over the platform. For example, ISVs are likely to offer additional audio and video processing components for processing environment generated content, beyond the acoustic event classification and visual scene analysis components offered by the SMART consortium. Service integrators and ISVs will be paid for their technology provision services by ASPs. Note that this might be a modality particularly interesting for SMEs, who may work on targeted enhancements to the SMART platform.

Hence, SMART should provide APIs enabling service integrators to build added-value services over a multitude of readily available sensors. In several case, service integrators may also engage in the deployment of sensors and their subsequent integration in the SMART system. Overall, integrators of

SMART systems can use the software and/or middleware libraries of the project in order to build and deploy applications that leverage the SMART search engine. These applications can be classified into two main categories:

- Public search applications, in which repositories of sensor content are publicly available i.e. through the public internet and based on conventional web applications.
- Private search applications, in which sensors and sensor processing components are deployed in order to satisfy the specific business requirements of a specific customer / end-user.

Overall, several of the deliverables of the SMART project will target services integrators, in order to enable them to build a range of (private or public) search applications. The later will be used by end-users. The role of end-user in the SMART value chain is illustrated in the following paragraph.

4.3.4 End-Users (surveillance, news media)

Surveillance

From a “Surveillance and Security” point of view, the challenge of the project is the development of knowledge and innovative technical solutions, useful for the development of new markets related to security infrastructure. The project aims to produce concrete results:

- in the domain of automated processing of information flowing from different sources;
- in the development of search engines of multimedia content;
- in the management and sharing of knowledge related to such processing.

Moreover, the project aims to enrich the emerging market ofproducts and services for knowledge management, evaluating, right now, the innovative potential in areas where collaboration between public and private is essential to the welfare and safety of a territory.

For the SMART security domain, we have chosen to take reference to the supply chain management of multimedia data, for activities related to:

- public security;
- surveillance of the territory;
- administration of justice.

Behind these systems a complex constellation of public and private stakeholders operates They directly or indirectly contribute to services for homeland, citizens and businesses security.: private security, police, local authorities, but also banks, post offices, retail shops.

News Media

The most important outcome of SMART for a media company is the creation of an automatic source of reliable news and statistics.

The data extracted from the physical sensors and social networks could be used by a news media company as a news source, complementary to the conventional ones.

The statistics extracted and the inferred news from the data collected, presented to the final user on an adequate interactive web interface could have a great value for the final user, especially if these statistics take into consideration the location and social network environment of the users.

Other Individual Users

Consumers and individuals registering for a minimal capability search over the SMART platform. Individual users are less likely to employ subscription services, but they are likely to participate in the platform based on other forms of incentives (e.g., credits for using the platform as soon as they also contribute to the platform).

4.3.5 Open source community

SMART is in the most part an open source project. It will be implemented based in open source technologies (as much as possible), and the main components of the SMART engine will be implemented in the form of open source software. Many components will be integrated with the search engine of the consolidated open source project terrier.org.

So, a SMART objective is the creation of an open source community around it. The community is very important for sustaining and evolving the project results, especially after the end of the project.

Therefore the open source community is a stakeholder of the project, and the open source developers are also users. They have their own needs, and if SMART wants to create a healthy community, it is important to take the open developers specific requirements into account.

One of the most important outcomes of SMART will be an open framework, an extensible set of software libraries. The extensibility and openness of the SMART system, will enable application integrators to extend the SMART search framework with:

- New physical sensors (e.g., cameras, wireless sensor networks).
- New virtual sensors such as filters over a social network (e.g., a Twitter sentiment analysis filter)
- New visualization components or libraries (e.g., mashup components).
- New perceptual processing components (e.g., video processing algorithms) and sensor signal processing algorithms.
- New algorithms for reasoning and/or fusion over multi-sensor data.

The open source nature of the project should also enable the contribution of the above-mentioned extensions as open source components.

5 Project background

This section provides a description of the project background per partner and other projects in which the partner is involved in relation to the technological focus that is considered for the project according to its role. Furthermore, it identifies requirements associated with the (re)use, extension and enhancement of background technology components in the scope of the SMART project.

5.1 ATOS: application development and integration

Atos (formerly Atos Origin) is an international information technology services company with annual 2011 pro forma revenue of EUR 8.5 billion and 74,000 employees in 48 countries. Serving a global client base, it delivers cutting edge transactional services, consulting and technology services, systems integration and managed services. With its deep technology expertise and industry knowledge, it works with clients across several market sectors.

Atos Research & Innovation (ARI), is a key reference for the whole Atos group. Founded in 1987, ARI has developed about one hundred projects with the European Commission. Atos are the one of the key players of the Research in Europe, being members of initiatives as NESSI [45], Future Internet[46], NEM[47], EOS[48], or ISI[49].

In most of past research projects, Atos was responsible for the development and integration tasks. Atos has participated in several research projects with open project outcomes, and there is a specialised unit which dedicates they work to Open Source research.

Atos (Media unit) have gained a long experience in the development of integration activities in research projects about Multimedia, with projects like OLYMPIC[50], GAMA[51], iSCORE[52], LiVE[53], ALIS[54], or My-eDirector[55].

For instance, GAMA is a multimedia search engine with some points in common with SMART. Atos' Media unit worked on the integration of the whole system, using an Enterprise Server Bus and developed a Digital Rights Management system. In the project ALIS, Atos developed a web interface for the system based on Ajax techniques and worked in the integration with some advanced knowledge technologies. An example of good practices for requirements can be extracted from SeCSE project (Service

Centric Engineering).

Regarding open source, Atos participated in the Qualipso project [56], a platform for trust and quality in open source system, which provides us a good theoretical foundation for defining the requirements for an open source projects. On the practical side, the project PuppyIR gives a good experience for the requirements, because PuppyIR is an open source library related to the SMART intended functionalities.

5.2 AIT: video processing

Background modeling: Adaptive background schemes that learn the background as the persistent colours in every system have been implemented in AIT. These systems are based on the spatio-temporal adaptation of the learning rate, based on the targets being monitored [1]. Research in SMART is expected to improve on these systems, so they become a robust basis for the motion-based tracking systems and the background change detection systems.

Tracking systems: At AIT, tracking systems based on Kalman and Particle Filtering have been build and successfully tested in the CLEAR 2007 evaluations [2-4]. These systems operate on either two [2, 4] or three dimensions [3,5,6], utilizing a single or multiple cameras viewing the monitored space. 2D tracking does not require camera calibration. 3D tracking requires the calibration of the single or the multiple cameras. The systems operate both indoors and outdoors [4,7,8]. The trackers measure visual properties of the targets, either colour, motion, outline of object presence. All these different measurement cues are modeled by likelihood functions expressing the probability to measure given the state. The trackers are either utilizing a single of these modalities, or multiple ones, being fused with a variety of schemes [7-9]. The background on tracking systems and background modeling is detailed in [10]. Further research into the fusion schemes is expected to be carried out in SMART.

Recognition systems: Face recognition systems have been built at AIT, focusing on video-to-video recognition. A sub-class Linear Discriminant Analysis system with automatic sub-class partitioning [11] has been built and successfully tested in the CLEAR 2007 evaluations [12]. At SMART, the face recognition problem for suspect individuals is formed as a still-to-video one.

5.3 IBM: acoustic processing

Audio event classification: IBM has developed various audio analysis and classification tools focusing mainly in speech and voice audio. Examples for different projects include detection of voice pathologies from speech samples [13], emotion detection in speech samples in the HERMES FP7 project [14], and speech diarization [15]. Those different applications are based on similar classification process. The process starts by extraction of relevant features from the audio. Those features are then statistically modeled to compensate for variable lengths of the audio data. At the end, machine learning tools are used for learning a classification models from a labeled set of data and to apply this model to the test data.

Speech transcription: IBM is one of the top leaders in speech transcription technology. This technology is used for different application such as transcription of voice conversation [14], real-time speech translation [16], and speech search [14]. Speech transcription was also used in the HERMES FP7 project.

Speaker identification: state of the art voice biometric technologies are currently being developed at IBM [17]. Those technologies are used for application such as speaker authentication and speech diarization. The core technology is similar between the different application and models the speech data first info frames of MFCC and then models the frames with GMM into GMM super-vectors. Comparison, classification and clustering can now be performed in the GMM super-space. Dimension reduction techniques such as Joint Factor Analysis (JFA) are used for reducing the inter or intra speaker variability as needed.

5.4 Imperial College

Imperial College has a strong background in the logical, mathematical and computational foundations of multi-agents systems and social networks, as well as developing technology for pervasive and affective computing applications. The teamalso have experience in legal knowledge representation and reasoning, and policy-based computing.

In the field of multi-agent systems, the team has made substantial contributions to the theory of collective coordinated action [19], the theory of collective decision-making (especially computational social choice) [20], and both the theory and practice of self-organising and self-adapting systems [18]. Particularly relevant here is the use of logic-based action languages from Artificial Intelligence, especially the Event Calculus, for event recognition and the representation of institutionalised power (the performance of a designated action by an empowered actor, usually said to be occupying a privileged role, to create facts of conventional significance) in norm-governed multi-agent systems.

In the field of pervasive and affective computing, Imperial College has pioneered innovative sensor technology used to measure biometric signals, for example galvanic skin response, and the use of fuzzy logic to make some estimate of affective state. This information can be used to deliver a service or side-effect some display or interface in the environment. Affective information can be linked to policies, which are customised to individual users on the client-side, and can specify what events are of significance and what action to take on their detection, subject to user-centric specifications of how information may be used and stored [22]. This has also been related to legal reasoning about intellectual property in content management systems [21].

In the domain of social networks, we have developed algorithms which enable individual nodes embedded in a noisy, uncertain and partially perceived environment to get a more refined, or better-informed, view of the world through process of opinion formation [24], social influence [23], and machine learning [26].

5.5 S3Log: application development and integration

S3log is a consortium formed by "Selex Integrated Systems", a Finmeccanica company, and Vitrociset, with the aim to create the first Italian industrial sector ITLogistics for the Defense.

The main areas of activity S3log include consulting, systems development and information technology services for the logistic support and optimal management of assets, resources and processes of organizations and agencies of the Operational Defense.

The needs for innovative and integrated solute, capable of simplifying access to information and performing information mining, arise from the growing involvement of the italian Army in multinational joint scenarios to grant the operability and to support strategic decisions.

The involvement of S3log within the SMART project is born from the interest in increasing the current offering of services and products with services and products related to security issues integrating, where possible, the current assets with the SMART search engine.

Following, a brief list of R&D and related projects:

- **SG-162** *Distributed Simulation for Air and Joint Mission Training* (*NATO NIAG*)
- **MSG-096** *Consequence/Incident Management for Coalition Tactical Operations* (*NATO RTO*)
- **iLog4All** Infologistics platform for the management of general aviation traffic (Regione Lazio – POR)
- **iLogCS3** infoLogistic Combat Service Support System (Min. of Defense – PNRM 2011)
- **NIRVANA** Nucleo Interventi Robotici Virtuali Addestramento Natura Asimmetrica/Ambientale (Min. of Defense – PNRM 2011)

5.6 Telesto: non-A/V sensors and application mashups

An evolution of the Internet is the "Internet-of-Things" where the multitude of sensors operating in multiple environments will be inter-connected.

The Telesto Sensing Framework (TSF), is a unique approach to the acquisition and representation of data from heterogeneous sensors and sensor networks.

The combination of the abilities of distributed data processing and data transmission networks leads to increasing progress in the field of collection and management of data that originate from the natural environment and human activity.

Telesto Technologies invests on the field of collection and homogenization of data retrieved from heterogeneous sensor networks. We design and constantly develop the Telesto Sensing Framework (TSF, current version TSF v0.5) which relies on standard solutions, specifically the SensorML model of the Open Geospatial Consortium. The functionality of TSF is based on the development philosophy of Web Services (Service Oriented Architecture - SOA). This ensures the ability of using third party services without interoperability limitations.

Today, TSF includes the following subsystems, which are described in short below:

- The Sensor Data Collection Server (SDCS) which collects raw data from the sensors and transforms their observations in a standard, XML-type format
- The main server of the application (Sensor Application Server - SAS) that implements the functionality of client applications. It also exposes a service interface (WSDL) to third party applications, providing full interoperability with Geographic Information Systems (GIS).
- The Sensor Database Store which provides storage and retrieval services for the application data.
- The Sensor Application Registry which maintains a registry of available sensor monitoring applications. Client applications query this server to dynamically discover the available services.
- A client application (Sensor Web Client) which visualises the results on a Geographical Information System allows the TSF to operate autonomously.

Sensor networks can be employed in detection and monitoring of environmental changes in forests, oceans, crops etc as well as human-originated activities such as the occupation of parking slots in neighboring parking areas. The developing sensor network infrastructures lead to the implementation of applications such as:

- Meteorology, protection of environment and forest resources. Applications related to the monitoring and processing of natural environment parameters, visualization in a geographical platform and composition of information for supporting decision making systems.
- Precision farming. The main goal of these applications is to maximize the profit of producers by monitoring the characteristics of the soil, in order to interfere with precision to improve the production (for example activation of anti-freezing measurements when alerted from the sensor network, selective irrigation of part of the crop in order to save water and other applications).
- Hybrid solutions of telematics and telemetry. A vast number of applications utilize traditional concepts of telematics systems (monitoring and control of car fleets), combining them with those of sensor networks (monitoring and control of physical parameters), in order to formulate a complete monitoring environment of changes in critical quantities such as changes in temperature of products during the delivery chain, where in example the temperature of every palette of products is monitored from the factory to the delivery point.

5.7 GLA: Search engines

Real-time indexing: The Terrier information retrieval (IR) platform [38] is at the heart of all efficient and/or effective retrieval strategies developed by GLA in the last 10 years. Terrier deploys various indexing schemes to allow it to produce appropriate data structures to retrieve from large corpora, namely sort-based inversion, compressed in-memory inversion and MapReduce indexing [35]. In particular, the MapReduce indexing scheme uses Hadoop infrastructure to distribute indexing processing across multiple computing nodes, to achieve high throughput for massive datasets. However, as a Hadoop/MapReduce framework is not suitable for the low latency processing of real-time data streams, research in SMART will investigate suitable infrastructures (e.g. S4, Hbase co-processors) for real-time indexing, such that newly received updates are available for retrieval as soon as possible after they are received.

Query scoring: GLA has developed a plethora of retrieval models for search engines in different application areas, including, but not limited to searching of: the Web [42-44], intranets [41], email [30], people [31,33], blogs [32], Twitter [37], medical literature [29] and news [36]. Moreover, we have made theoretical contributions to the IR field in areas such as: document length normalisation [26], query performance prediction [27], relevance feedback [28], proximity [39], efficiency [34], and the diversification of search results (when the user’s query is ambiguous) [43,44]. Additionally, we are researching and deploying recent advanced techniques in learning to rank [33,36,44], where machine learning is used to appropriately combine many features in the retrieval process. Research in SMART will focus on additional developing novel features and advanced ranking models suitable for ranking sensor data, encapsulating aspects such as location, timelineness, and diversity, ensuring the search component responds to user queries in an efficient manner. This will permit an appropriate search engine for the use cases pursued within SMART.

5.8 Prisa: end user of smart application and social media

PRISA is the world's leading Spanish and Portuguese-language business group in the fields of education, information and entertainment. The PRISA Digital is the catalyst of online business growth across the PRISA Group by defining overall online strategy, by managing online monetization, by leading online innovation, by supporting the creation of new online businesses and providing online technology platform services to Business Units creating technology platform synergies across the Group.

The online news experience of Prisa managing more than 20 news channels either in online news radio or TV addressing Spanish and Latin markets gets 30 millions unique users monthly.

References and R&D news related projects

The Research & development department of Prisa Digital has been involved in different projects as Spanish “Plan Avanza2” collaborative project WebN+1 as well as many internal innovation projects to add new capabilities to existing Prisa’s Business Units products and launch new products in the forefront of technology.

Other Prisa R&D Projects include:

- Development of Eskup; real time news social network deployed in the online version of elpais www.elpais.com with more than 200k users;
- El viajero 2010 video player with interactive hotspot for travel recommendations http://elviajero.elpais.com/destinos-2010/;
- El pais Ipad (top 10 news tablet app Imonitor October 2011) http://mcpheters.com/2011/10/17/imonitor%E2%84%A2-releases-list-of-top-10-best-newspaper-apps/;
- La mar de noches (interactive visual player) http://tv.los40.com/lamardenoches/;
- We are currently developing multiscreen news channels on connceted TV (LG and Boxee) and Xbox http://www.elpais.com/tv/lge/.

5.9 SDR: end user on Smart City applications

Santander City Council has been characterized by his eagerness to provide an city management closer to the citizen through the use of new technologies. In this sense, the City Council has been very proactive in trying to become Santander as a smart city through the SmartSantander initiative.

Santander City Council is also running several research projects, funded by the European Commission under the Seventh Framework Programme (FP7), in the field of the smart cities: OutSmart (Street Lighting and Smart Metering), Burba (Intelligent Waste Management).

Furthermore, the City Council has participated in many other related projects among which may highlight the following:

- Deployment of proprietary networks combining different technologies, on a scale medium to high, for data transmission, traffic surveillance systems through TV Cameras.
- Monitoring and management of urban public transport vehicles routes such as buses and taxis using GPS positioning.
- Development of a Geographic Information System (GIS) for the catalogue and registration of municipal resources with the development of individualized records of each of the elements of water supply and sanitation networks, such as wells, pipes, connections, tanks, hydrants, suction cups, and so on. Each card includes a photograph of each item, its current status, anomaly detection, verification of operation, whether or not leakage, number of junctions or coming from it, surveying and geographic location and dates of each of the installations, modifications or repairs.
- Management of a payment system on public transport through RFID tags and Multilanguage Information System for Blind Persons (SIO), so that through a reader can receive radio signals indicating that a blind person on the bus is providing information about stops.

In relation with this project, on the one hand, the Santander City Council will provide access to the city infrastructures, facilitating either access to the public buildings and facilities where will be installed the devices needed in the Smart project, as well as providing access to the traffic surveillance cameras which will permit video processing and putting into practice some of the use cases. On the other hand, the Santander Council will be responsible of the purchasing process of the cameras and microphones selected by the partners, providing technical support in order to integrate the devices in the data local network.

6 Related projects

This section reports on state-of-the-art work performed by R&D projects (EU FP7) in the areas of Multimedia Search, Face/people tracking, Video Stream Annotation, Audio scene analysis, Heterogeneous sensors data fusion (A/V and non-A/V), or other R&D activities of relevance to the project. In the sequel we present shortly some of them.

Project name	Description	Relation to SMART
Terrier	Terrier is an open source platform for the research and rapid development of search engines. It is maintained by the University of Glasgow (GLA). Terrier's focus is on ensuring that the most effective document retrieval models can be easily deployed or developed within the platform, and its success has been assured by world-leading participations within international evaluation forums dedicated to textual search tasks, such as TREC (English), CLEF (European languages), NTCIR	The SMART open source framework will build upon the Terrier search engine. In particular, the consortium member GLA will research and develop new real-time indexing architectures tailored for the handling of multimedia data streams from sensors and social networks, as well as advanced and anticipatory retrieval models that can

	(Asian languages) and FIRE (Indian languages).	rank events by interestingness.
Socios	Socios is an EU FP7 project that provides cross-platform tools for developing applications analysing social networks, such as social graph mining and recommendation. The Socios Core API permits differences between social networks to be abstracted away from application-level code.	Socios could provide an additional component to the SMART framework providing enhanced social network analysis capabilities.
My-eDirector 2012	My-eDirector 2012: (Real-Time Context-Aware and Personalized Media Streaming Environments for Large Scale Broadcasting Applications) It is a R&D project that aims to provide a unique interactive broadcasting service enabling end-users to select focal actors and points of interest within real-time broadcasted scenes.	The person tracking and scene analysis knowledge is a good starting point for the visual processing systems developed in SMART. In the scope of SMART the my-eDirector 2012 background modelling technologies will be used towards the development of crowd and colour analysis modules.
LIVE	LIVE: (***intelligent television programming and services)***. The LIVE project improves on linear approaches to TV broadcasting of live events by providing digital technologies and content formats that enable viewers to shape their own personal and highly interactive viewing experience.	For SMART, it is interesting the knowledge in the development of semantic-based and user/content-aware systems.
DANAE	DANAE (IST 507113: multimedia content delivery in context-aware environments) addressed the dynamic and distributed adaptation of scalable multimedia content in a context-aware environment. Its specific objective was to specify, develop, integrate and validate in a testbed a complete framework able to provide end-to-end quality of service (for multimedia content) at an acceptable (usability) cost to the end-user.	Knowledge can be exploited on the quality of service for networking the sensors with the edge nodes.
NoGoSoN	NoGoSoN (UK EPSRC GR569252): was a Programmable Networks Initiative project investigating the logical and computational specification of norm-governed self-organising networks. The project showed how cognitive agents, equipped with appropriate reasoning capabilities, could shape and dimension a network to optimise with respect to prevailing operating conditions.	Knowledge can be exploited on Reasoning for the perceptual components of the edge node.

Table 6: Related projects

7 Functional and Non-functional requirements for technology elements

7.1 Functional requirements

The following subsections provide an analysis on the functional requirements as implied by the functionalities offered by the SMART partners.

7.1.1 ATOS: application development and integration

SMART is a complex project, with many different components that should work together to offer interesting services to the users. In order to construct an integrated system, the development process must fulfill some requirements. These functionalities requirements are intended for developers or integrators, not for final users.

Each component of SMART must be developed according accepted good programming practices, in order to reduce the complexity of code comprehension and difficulty of maintenance tasks. For instance, the code must be well documented, well indented and clear in general.
An important point related to good practices is the storing of configuration values for integration. No configuration values should be written in the source code. The components of SMART should separate their internal configuration options from the implementation, and this should be configurable easily.

The architecture must be modular with a clear separation between the modules. The components should be loosely coupled, allowing the development of new components that can be easily added.

The architecture of SMART must be layered, providing separation of concerns. For instance, SMART architecture can be structured in layers for Information Acquisition, Information Processing, and Retrieval.
SMART components should use common and open standard formats for data representation. This is especially important for data exchanged between different components.

Apart from these general requirements for design and developing, one of the main results of SMART will be an open source library. The open source developers and the open community source are stakeholders of the open source library, and they have their own specific requirements. The open source community is very important for the long-term success of the project, so these requirements should not be overlooked.

The first requirement for an open software project is to be delivered using an “open source” or “free source” license. It can be seem trivial, but there are several definitions about what is “free” or “open” according different institutes, foundations or standardization bodies, For the purposes of the document, we accept the definition and the terminology of the OSI (Open Source Initiative), so we call the software “open” and the library source code shall be published using a open license according OSI.

In practice, it is not enough for the success of an open software project being merely open source. For the long term success in the open source community, the SMART library shall use other open source libraries and tools when possible. For instance, it is a better idea to use MySQL or other open source database instead of a proprietary one. Proprietary databases (like Oracle or SQL Server) are common in companies, but very rare in the open source community. The requirement of a license for the database could be a serious barrier for open source developers.

Something related happens with the programming language. The open library must use a language with good support from the open source community. Fortunately, this requirement is less strict nowadays than in the past. Currently we have free implementations for almost any program language. But, in any case, it is a better idea to choose a well-accepted language, with good support from open source community. Apart from these considerations, the election of the programming language is very important from the developing and integration point of view. The SMART library must be written using a proven language, with modern characteristics and a mature library to build upon. Examples of good languages could be C, Java, Python or Prolog.

7.1.1.1 Requirements summary:

- The open source components of the SMART framework must be released under an Open Source License.
- SMART should reuse existing open source software and tools, where it is appropriate and possible according the license.
- SMART should use common and open standard formats for data representation, especially for data exchanged between different components.
- The architecture of SMART must be modular, with defined interface between modules.
- The architecture of SMART must be layered, providing separation of concerns

- The components of SMART should separate their internal configuration options from the implementation.
- The components of SMART must be developed according accepted good programming practice.
- The components should be developed using proven and trusted languages. The open components should use a language with good open source community support.

7.1.2 Media Data Management

As was stated previously, an event in SMART is a series of features occurring at a given point in time. The search engine will return a list of events, including links to relevant multimedia information, if is available. The Media Data component will be able to present the multimedia information to the final user.
Sometimes the relevant multimedia information about an event begins before the event itself. For instance, in the case of a car accident, we want to see the position and trajectory of the cars before the crash.
This make necessary to store a record of all media information for some time. This information usually will be discarded, unless there is an event. In the case of an event, the media information will be stored permanently, so the search engine will be able to present it to the users.
The multimedia information (especially, video) must be adapted to the terminal of the user, taking into account aspects like screen resolution or bandwidth. As the number of types of possible terminals is almost infinite, the Media Data will adapt the contents to the main types.
All data and video storing must obey the applicable legislation of the country and respect the privacy and ethics aspects.

7.1.2.1 Requirements summary:

- The Media Data component will be able to present multimedia information to the users.
- The video information will be adapted to the most common types of terminals.
- The Media Data component will be able to store video for a predefined time.
- Media Data component should respect the legal and ethical aspects.

7.1.3 AIT: video processing

Many video processing technologies are going to support the live news and security use cases. Central to these technologies is the tracking of foreground objects, usually not as individuals but as masses. Such algorithms are those for traffic and crowd analysis. This analysis can be:

- Static, yielding a crowd-ness factor (related to how many people are present).
- Persistent colours, yielding the trends of the city for automated posts in Eskup.
- Dynamic, yielding crowd behavior (concentrating at a point, dispersing away from a point, direction of flow).

The environmental conditions can also be visually monitored: Sunshine or rain can be detected, and such conditions can change the presentation style at the news player. Related to the presentation is the automatic capturing of clips (images or small audio-visual recordings) that represent incidents of interest.

These events of interest, like road accidents and crowd gatherings, are detected by a higher level of reasoning. The incidents are related both to the live news and the security use cases.

Tracking of individuals is more important at the security use case. Their loitering or isolated motion in restricted areas can be inferred when the areas are not crowded. The classification of social activity detection is by far the most difficult to establish, especially since such activities can easily be carried out in crowded places. The crowds pose problems also in recognizing suspect individuals.

On the other hand, change detection in the background can be carried out even in crowded spaces, since this involves detecting changes in the background. Such changes can range from the common emptying of a parking space, to the very suspicious appearance of a left object.

All the involved video technologies per use case are shown in Table 7.1.2. Highest priority will be given to the technologies whose difficulty is marked as "OK", implying algorithms that at the beginning of the project are deemed feasible and with adequate expected performance.

All these technology components report to the indexing mechanism of SMART. The frequency of the reports is also shown in Table 7.1.2, and spans from as often as once per minute, to as rarely as once per week, or in some cases upon the detection of an incident. The only component not reporting directly is the tracking of individual cars. This is used to reason about tracking incidents to report.

Use case	**Video processing**		**Difficulty**	**Frequency**
Live news	Crowd analysis	Density	OK	Minute
		Gender	Difficult	Minute
		Race	Difficult	Minute
		Age	Difficult	Minute
		Colours	OK	Week
		Incident	Reasoning	Minute
	Traffic analysis	Density	OK	Minute
		Tracking of individual cars	OK	Never
		Incident (accident, improper driving)	Reasoning	Upon incident
	Clip capture	For presentation	Reasoning	Upon incident
	Environment	Day/night, weather	OK	Hour
Security	Loitering	Tracking persistent person	Difficult in crowd	Minute
	Isolated motion	Tracking @ isolated places	OK	Minute
	Change detection	Background changes for parked cars, left objects, etc.	OK	1-5 minutes
	Clip capture after audio event	Triggered recording, target camera if DOA	OK	Upon incident
	Activity	Classification of social behavioursr	Difficult	Minute
	ID dangerous individuals	Track and ID faces	Difficult in crowd	Upon incident

Table 7.1.2: Video technologies supporting use cases.

7.1.3.1 Requirements summary:

- SMART system should support at least the technologies marked with OK in Table 7.1.2
- Video resolution should be at least VGA for visual scene analysis. Face and person tracking require much higher resolutions, which the selected camera can offer.
- Video processing algorithms in SMART should be real-time (there is no storing of data, unless a specific event is detected). The minimum frame rate for significant inter-frame correlation is 10 FPS, hence the algorithms need to process frames this fast.

- A constant latency of a few seconds is tolerated by the system.

7.1.4 IBM: acoustic processing

Detection and classification of audio event can be used for both the live news scenario and for the security scenario. In the live new scenario this can be used to classify different surrounding noises. For example: different crowd noises, outdoor and indoor events, and traffic noises. Different sub categories can be also identified for example: small crowd, large crown and cheering crowd. Another example can be no traffic, moving traffic, and traffic jam. Those audio events have continuous nature and therefore require some amount of time for proper analysis and classification. The amount of time may depend on the types of the events and on the required accuracy level. Higher report rate will mean less accurate classification. The report rate can vary between 10sec to several minutes.

The security scenario can use similar audio event classification and in addition it can also use short term event detection. This is useful for detecting specific audio event such as door opening and closing, glass breaking, and car noises. Since those are short term events, reports of their presents should be close in time to their occurrence.

Speech transcription can also be used for both scenarios. The main idea is to transcribe voice messages that will be recorded and the send the transcribed text for additional processing. In the live news scenario those messages can be news reports from various reporters. In the security scenario those can be security alerts from security personals.

Since speaker identification require speech sample it would be natural to combine it with the speech transcription. It can be used for example to identify or verify the person who leaves the message. The accuracy of this process depends on the type of speech which is used for the identification as compared to the training text. If the same text is used in both cases (e.g. a password) then the accuracy is higher than if different text is used.

Another approach for speaker identification would be to track a speaker in time or space (different microphones). This would probably be more useful for the security scenario. This approach is more difficult since it uses speech recorded away from the microphone and therefore, the accuracy might be reduced considerably.

7.1.4.1 Requirements summary:

- SMART system should support audio event classification.
- Audio input for the audio event classification should have a bandwidth of at least 16 KHz.
- The system should tolerated delays in the audio event classification of several minutes.
- SMART system should support speech transcription for voice messages
- Speech samples for the transcription should have a bandwidth of at least 16 KHz.
- The system should tolerated delays in the speech transcription which are similar in length to the length of the speech samples.
- SMART system should support speaker verification for voice messages
- The system should tolerated delays in the speaker verification which are similar in length to the length of the speech samples.

7.1.5 Imperial College

The SMART project requires some reasoning capabilities to infer higher-level events (or any other valuable pieces of information) from sensor outputs. However, traditional rule-based event/activity recognition is limited to the pertinence of declarative event/activity patterns inserted in the knowledge base, that is, an activity or a complex event not foreseen by designers cannot be recognized by a traditional rule-based recognition engine. In order to push these boundaries, SMART should be capable of learning patterns of significance from sensor data over varying timescales, using machine learning algorithms.

Furthermore, due to the existence of legal and user's policies, SMART has to implement a set of fair information practices, implemented as a policy on how user-generated content (UGC), and information derived from it, is to be collected, used and disposed of.

Finally, SMART should support UGC search in both push- and pull-modes in response to specific user queries and subscription mechanisms.

7.1.5.1 Requirements summary:

- It should be able map distributed sensor data from several data streams to perform complex event recognition, including atomic events, compound events, and situations.
- It should be capable of learning patterns of significance from sensor data over varying timescales, using machine learning algorithms.
- It should implement a set of fair information practices, implemented a user-centric policy on how UGC data, and information derived from it, is to be collected, used and disposed of.
- It should support UGC search in both push- and pull-modes, i.e. in response to specific user queries and subscription mechanisms.

7.1.6 S3Log: application development and integration

Technology has an increasing influence on daily surveillance activity for the purposes of ensuring security in the territory, sometimes completely redesigning the processes. Today security scenario requires a high capacity, by the staff, to evaluate risks or suspicious situations to better coordinate early operations coordinating all the stakeholders.

However, only some of the video and sensors streams are monitored by the operating centres. The most of them are registered h24 using VCR, HD, HD Servers, video server etc.. without connection with the security operators. This lack of connectivity with security operators, induces major technical and logistical problems in case you need to access or view the contents in case of crime, and represents a barrier to prevention.

In case of crime, for example, the institutions responsible of surveillance and central police are responsible of viewing dozen and dozens of hours of video streams to retrieve information and recognize the author of the crime. The cost of this operation is often relevant and the results do not justify the expenses.

This need for cooperation, although is perceived as necessary, requires additional resources and technological support both in analyzing and representing data. The human analysis does not always guarantee the best result. Operators need technologies to support decision making and data mining.

7.1.6.1 Requirements summary:

- The SMART platform should be modular and should be designed using most common design patterns;
- The SMART platform should be designed to facilitate the possibility of enhancing the user experience. The access to the SMART platform should be provided into two different ways: a low level API or through many pre-compiled widgets;
- SMART access widgets should be designed using languages usually adopted by the open-source community. To extends the use of the platform to the mobile handsets, HTML5 could be a good approach to SMART widgets programming;

- The SMART platform should be extensible by providing clear API to create specific add-ons;
- For security purposes, the SMART platform should distinguish signed and not signed add-ons;
- The SMART platform should be capable of managing different users profiles distinguishing between operators;
- The SMART platform, in accordance with privacy policies, should be capable to provide, if necessary, evidence of a crime. The platform should store privacy covered data in a protected way. Access to protected data should be possible only to authorized operators and only in case of a crime event;
- The SMART platform, in accordance to local security policies, should be capable of applying digital signature to content before storing them;
- The data stored should be accessible for post-processing operations. Post processing should be performed using existing SMART components or user created add-ons.

7.1.7 Telesto: non-AV sensors and application mashups

The Open Geospatial Consortium (OGC®) is an international industry consortium of 445 companies, government agencies and universities participating in a consensus process to develop publicly available interface standards. OGC Standards support interoperable solutions that "geo-enable" the Web, wireless and location-based services and mainstream IT. The standards empower technology developers to make complex spatial information and services accessible and useful with all kinds of applications.

To address the lack of interoperability between sensors, the application of Semantic Web technologies is proposed that can fill the gap of contextualization in sensors and enable a mashup in services provided on top of them. Recent efforts from the Open Geospatial Consortium go in the same direction by abstracting from XML-based serializations the Sensor Web Enablement standards i.e. SensorML and Observation-and-Measurement. Such standards have also been mapped into an ontology by the W3C Semantic Sensor Network Incubator Group. The advantages of applying Linked Data principles on sensors relies on the contextual information added by linking to the Linked Open Data (LOD) cloud. For instance a user can drive faster if he follows the routes suggested by his particular GPS car navigation system whose suggestions are based on crossing the information about the hilly surrounding area (from Geography LOD datasets) and the near both road works (from Government LOD datasets) and ongoing social events (from Media LOD datasets). The following figure gives an overview (mindmap) of the sensor observation service as defined in the frame of the OGC and the various operations that are applied in the sensors.

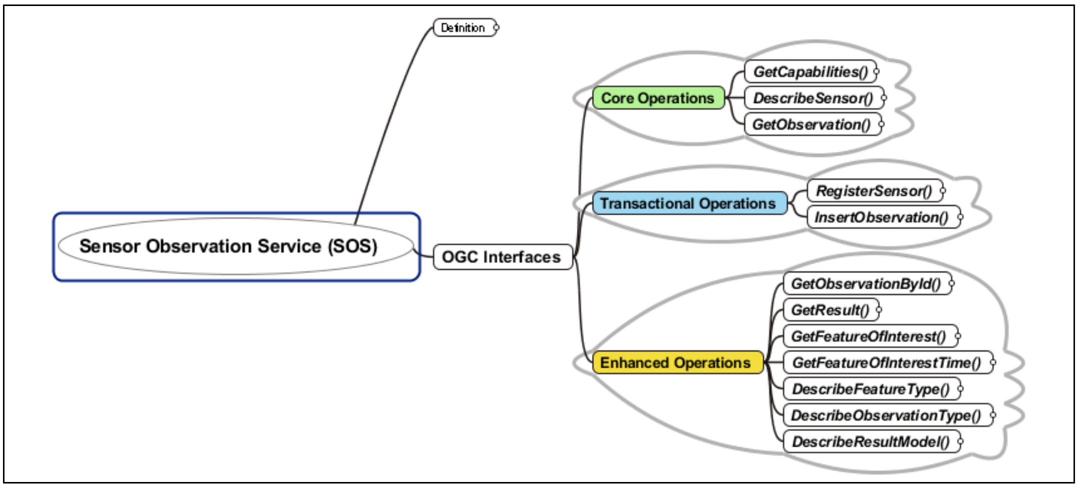

Figure 3: Operations of the Sensor Observation Service

Current applications providing Linked Sensor Data - e.g. Sensor Discovery On Linked Data - actually link only to a restricted set of LOD datasets that users cannot modify.

The main research goal in SMART is to enable a highly customizable annotation of the non-AV sensors deployed with respect to their context and values provided. It is then necessary to be able to use this measurements in a straightforward and standard way, in large scale and enable the content of the non-AV sensors to be subject to combined search techniques and mashup approaches for providing further added value services.

7.1.7.1 Requirements summary:

- SMART system should support a wide set of non-AV sensors;
- SMART sensors should adhere to the SensorML framework;
- Data captured by the SMART sensors should be able to be contributed to the Linked Data framework;
- SMART mashup technologies should be open and flexible to incorporate the functionalities implied by the SMART use cases;
- SMART should support for multiple social networks and for filters over social networks;
- SMART should support various "social sensors" (meaning: virtual sensors over various social networks).
- SMART should facilitate data linking and interoperation with the Linked Data initiative.

7.1.8 GLA: Search engines

Various search technologies are involved in the realization of the SMART vision. A central aspect of the SMART vision is the Terrier search engine, which will ensure that the latest sensor information is retrieved in response to a user's query.

It is proposed that information is displayed to users in terms of events: an event may appear on a map, or within a ranking for a search query. Internally, the search engine must then decide on the events to be shown, by ranking them by predicting their importance/interest to the user's query. Only those with highest predicted importance/interest will be shown to the users. The query may contain implicit or explicit hints to the search engine about the geographical location of interest – in such case, the search engine must select the appropriate locations to consider, perhaps focusing on results obtained from an appropriate subset of edge nodes.

It is proposed that an event is represented within SMART's search engine as a set of features occurring at a given point in time. The features will encapsulate the outcomes of video and audio processing technologies, as well as textual information about the location, and any geographically close textual information mined from available social streams.

For example, video features for a location might include the probability of observing a traffic jam or stationary cars. Audio features might include the probability that traffic is stationary based on noise, rather than moving. By combining these features, the system can infer that there is a traffic jam.

Time is a key concept within the SMART query scoring algorithms. In particular, for an event to be of interest, it should be surprising or unexpected. Stationary vehicular traffic at a set of traffic lights is expected, unless it hasn't cleared after a few cycles of the traffic lights. In this case, the novelty of the event, and the phase transition (normal events towards the unexpected) require that the search engine makes inferences about events over various timescales, and compare to similar timeframes in the past. Within SMART, we will monitor for one full month of events from the chosen locations, so as to gather a representative sample of historical background data. If the collection of data and legal aspects do not allow one month of full background history to be obtained, then a simulation of one month of background data will be created based on the expected distribution of events and features within such a period (e.g. 30 days & 30 nights; 20 days of morning rush hour traffic).

All of these attributes are implemented as features within the search engine. Using these features, an effective ranking model will be devised by research and integrating appropriate learning to rank techniques. Moreover, as the inputs of sensors, and the corresponding audio and video analysis is changing continually (e.g. perhaps every minute – see Table 7.1.2 above), the SMART search component needs to index the latest events and be in a position to provide their results in searches as soon as possible.

Input: The SMART search engine will be informed of SMART edge nodes to subscribe to. It will regularly poll all available edge nodes for updates on their processing information – indeed, as edge nodes are geographically distributed, they may be less reliable than the SMART search infrastructure, and hence the search engine should be robust to their unavailability. In particular, robustness should encapsulate two dimensions: *technical robustness* (the search engine does not hang on unavailability of the edge nodes), but also *result robustness* (results don't suddenly change just because an edge node becomes temporarily unavailable). There will be an API defined for the transferal of data between the edge nodes and the search infrastructure.

Output: The ability to obtain search results is provided through an API for use by the use cases/applications. Moreover, as the results of a query may change, the search system API should permit callbacks to the use case, to be provided with updated results, subject to appropriate triggers being activated within the search engine.

7.1.8.1 Requirements summary:

- The SMART search engine should produce rankings of events, possibly in response to a user's query.
- The search engine gathers new input from sensors via edge nodes, and must be robust to communication failures with edge nodes.
- Input from sensors via the edge nodes should be used to model the importance of these events. The current data from a sensor may be compared to the previous background history of that sensor.
- As recent events are likely to be more important, the search engine should regularly index new data arriving from edge nodes.
- These event rankings should be accessibly through an API available to use case applications.

7.1.9 Prisa: end user of smart application and social media

The system will integrate in a common interface all events (sensors, social and local news). Selecting a city map or a timeline or defined events type should be kept up to trackthe living pulse of the city.

End users will be classified as follow:

- Local citizens (Santander);
- Internet users talking about the city (Santander) issues or events (on social networks);
- Citizen journalist using (using Eskup) for local news (restricted group of user for first product test);
- Journalist: the system could be used by professional journalist to deliver real time local news

Eskup is the Prisa's Social network, used for publishing breaking news by authorized journalists, and for sharing user's comments and conversation about a topic or a news item.

An Eskup instance is going to be created for the project to be integrated on the live news use case as an input and publishing tool for:

- Local news;
- User's comments;
- Social conversation.

7.1.10 SDR: end user on Smart City applications

From the perspective of the end user who has to make use of applications or use cases that the project Smart aims to develop, these would have to distinguish essentially two types of users:

On the one hand or the end user itself, let's say the citizen, who can make use of applications and services developed by the project and from its use may obtain some useful information.

On the other hand would have to distinguish the citizen with a technical profile. It comes to technicians not related to new technologies, but related to technical areas about the use cases modelled in Smart (security or news). Areas such as traffic surveillance are vital in any city in the world and being able to use new technologies in surveillance would help greatly in the management of human resources available and would increase the quality of services offered to citizens. Other professionals will be able to consume the information provided by the use cases, both in the area of security as well in the media, in order to provide support in creating reports which will help to enhance services offered to citizens who indirectly will be recipients of these applications.

Taking into account user profiles described above, it is important that the applications and services developed by Smart accomplish with a number of key requirements:

- The Smart applications have to show a user interface suitable for the type of target user. The options and operations to be shown must be simple and adapted to the user needs that should be able to configure the interface to their preferences.
- The applications and services must be easily integrated into the existing infrastructure in the host city (in our case the city of Santander).
- The support technological core of SMART should be easily scalable so that a relatively simple way to be created new use cases or services in other areas not related to the cases to be developed by smart during the life of the project. But very important in order to achieve a sustainability model. The pursued goal is creating horizontal solutions in order to create a valid model in any city.
- The services to be developed must meet, inexcusably, with the regulations on the use of multimedia resources (audio and video) in public spaces so as not to infringe the privacy rights of citizens.

7.1.10.1 Requirements summary:

- The Smart applications have to be easy to use, especially for final users;
- These applications have to be adaptable to the different user profiles: technicians and final users;
- The information provided by the Smart applications has to be useful and interesting for all users;
- The Smart architecture has to be scalable and sustainable;
- The applications and technologies used in Smart must respect all regulations concerning the ethical aspects, especially those related with data protection and privacy.

7.2 Non-functional requirements

7.2.1 Scalability

It is essential that the SMART framework be scalable. Such scalability will permit the future expansive deployment in a variety of manners. In particular, SMART should be able to scale with respect to the volume of data from input sensors, and with respect to the number of queries answered in any period of time. This has a bearing on the design of different pieces of the SMART architecture, as detailed in Table 7.2.1 below.

Scalability Dimension	Effect	Mitigation	Measurement
More sensors at more locations	Audio/visual processing algorithms will need to process more data	More edge nodes can be added to handle audio/visual processing Sub-sampling of input data by audio/visual processing algorithms, or simpler implementations.	Latency of audio/visual processing algorithms. Achieved processing rate vs. incoming data rate.
	More indexing input to the search engine	More nodes can be deployed in the real-time indexing component of the search engine	Latency of an event being received until it can be presented in user search results
More users of SMART use cases	More user search requests	More nodes can be deployed in the retrieval side of the search engine	Response time of the search engine to a user query

Table 7.2.1: Scalability dimensions.

In the following we take each scalability dimension in turn, describing how SMART should be able to cope with increased scalability demands. Over both dimensions, one of the primary considerations of the usability of the SMART system is *overall latency* – the time elapsed between an event occurring and its reporting to the user. Of course, the latency requirements may vary for the applied use cases: 1-5 minutes might be sufficient for the news use case, but the security use case might have stricter requirements.

More sensors at more locations

SMART shall be designed such that more locations or more sensors can be added. Doing so, SMART will have to process additional input data from sensors. This will have two effects: increased processing for the audio/visual processing algorithms; and more data to be indexed by the SMART search engine. In both cases, SMART shall be capable of scaling horizontally (i.e. more computing resources, in contrast to vertical scaling, where larger computing resources are deployed):

- SMART's architecture shall permit the ability to add more edge nodes, each of which will apply the audio/visual processing algorithms to the input from a given number of sensors. The scalability of a SMART edge node will be measured in two manners: Firstly, by latency, i.e. the time from input being received from a number of sensors, until the time the audio/visual processing algorithms have processed that input and made it available to the SMART search engine to index. Secondly, the rate at which the incoming sensor data can be processed. In particular, if at any time, the processing rate is less than the rate the incoming data is arriving at, then the system is no longer real-time – and hence latency will increase over time. The latency of the audio/visual processing algorithms can be controlled in several manners: simpler implementations of algorithms, sub-sampling of the input data (esp. in the case of video processing), or by vertical scaling of the SMART edge node (adding more processing power). Of course, using simpler implementations or sub-sampling of the input data will reduce the accuracy of the algorithms – for instance, some video processing tasks deteriorate when sub-sampling amounts to less than 10 frames per second. Hence the overall targets are that one edge node of reasonable processing power can perform audio/visual processing for a number of sensors with latency less than one minute, such that the output rate does not fall below the input rate.

- SMART shall also permit the vertical scaling of the search engine component, by the addition of more search engine nodes – each of which will be able to process more input from a number of edge nodes. The scalability of the SMART search engine indexing system as the volume of input data increases, and as the number of search engine nodes increases will be measured by the indexing latency, which is the time between event data being received by the search engine, until the time when the event can be retrieved by the search engine. The target indexing latency is < 10 seconds.

More users of SMART use cases

SMART shall be designed to handle the interaction and searching by users in parallel. This will result in more queries being submitted to the search engine, possibly in parallel. The search engine should respond gracefully to increased query load. To deal with increased load, the search component shall support vertical and horizontal scaling, by permitting queries to be processed by multiple threads on the same machine, and/or adding more nodes to the retrieval process, thereby distributing the retrieval load. Response time latency between the issuing of a search query and the obtaining of the search results is a key measure of the scalability. The target search engine response time latency is < 10 seconds.

7.2.1.1 Requirements summary:

- The SMART system must be able to scale with respect to more input sensors. In this way, the number of edge nodes within a SMART system should not be limited, such that more edge nodes can be deployed to handle more sensors.
- The processing algorithms deployed by each edge node should contain be have sufficiently low latency, such that the edge node does not “fall behind”, and can always send the latest data to the search engine.
- The search engine should be able to scale to more input data from more edge nodes, by vertically scaling the search indexing processes across multiple search engine machines.
- The search engine should be able to respond to user queries from the use cases, and moreover should be able to handle multiple users of the use cases at once, such that results are obtained quickly even under load.

7.2.2 Performance

The accuracy (performance) of the SMART system is a key consideration. If the accuracy of the SMART system is poor, the end users will not be satisfied with the results, even if they are obtained very quickly. Table 7.2.2 below summarises the different dimensions of accuracy that may be evaluated within SMART, and the effect of their degradation.

Accuracy Dimension	**Degradation Effect**	**Measurement**
Accuracy of audio/visual processing algorithms	Reduced quality of feature data to search engine, thereby reducing the quality of results to users of use cases	On specific feature basis, using appropriate measures and ground truth data

Accuracy of predicted important events ranked by search engine	Reduce quality of results to users of use cases	Information retrieval (IR) standard/accepted measures such as Mean Average Precision (MAP), NDCG, Success@k

Table 7.2.2: Accuracy Dimensions.

In the following, we describe in further detail each accuracy dimension, detailing the manner in which each will be measured.

Accuracy of audio/visual processing algorithms

The various audio/visual processing algorithms applied are inexact, and may attain a variable degree of success. For instance, say that the crowd density detection algorithm becomes unreliable in low-level lighting conditions, such as night. This may result in false-negatives (e.g. crowd density detection algorithm predicts 'no crowds' when there are crowds) or false-positives (predicts 'crowds' when there are none). In the presence of a false-positive, the search engine may incorrectly show a crowd event to a user of the use case. To mitigate this, the accuracy of each deployed audio/visual processing algorithm will be carefully evaluated using appropriate and agreed evaluation measures. In these evaluations, performance will be measured by comparing to annotations. A sufficiently large sample of audio-visual content will be annotated by professionals, so as to obtain enough training data to allow the tuning of the processing algorithms and their accuracy. The annotations can be objective (actual tracks of people and cars marked on some video feeds) or subjective (the annotator describes a scene as crowded, normal or empty, doing so subjectively since in dense crowds, individual people cannot be counted). The consortium targets an accuracy improvement of 20% for each algorithm over the baseline algorithms. In many cases such systems exist (e.g. individual car or person tracking, speech transcription). In some cases, the systems are novel, therefore the early implementations will be used to gauge improvements as the used systems mature.

Accuracy of predicted important events ranked by search engine

Similar to the audio/visual processing algorithms, search engine rankings are inexact – for a given query, it is only possible to *predict* what searchers may be looking for. This is often based on training using existing ground truth data. We will measure the accuracy of the search engine component using evaluation measures that discern how similar the ranking is to the optimal. Such measures are commonly used in information retrieval (IR) research community as well as by commercial Web search engines to evaluate their systems, namely Mean Average Precision (MAP), Normalised Discounted Cumulative Gain (NDCG) and Success, each calculated to a given rank *k*. In particular, the target accuracies for the search components is Success by the 5^{th} ranked results of > 30%.

7.2.2.1 Requirements summary:

- The SMART system deploys various audio/visual processing algorithms that are inexact, uncertain processes. However, they should be regularly evaluated during development, such that they are shown to be accurate with real data.
- Similarly, the ranking of documents (events in the SMART case) by a search engine is also an uncertain process. The search engine will be appropriately trained and evaluated to ensure its effectiveness.

7.2.3 Reliability and availability

A system can be considered as reliable if it can operate normally even in the presence of some failures. It is evident that there can be no system considered as fault tolerant by design. However, it is important to anticipate forehands the various failures that might occur and try to eliminate their impact in the overall service provision. In this way it can be assured that failures will have no "fatal" impact for the system. Moreover, service workflows can be considered in the design phase so that alternative ways for service provisioning can be provided.

The relationship between reliability and Mean Time Between Failures (MTBF) is defined as follow:

$$\mathrm{Re}\,liability = e^{-\left(\frac{Time}{MTBF}\right)}$$

The above mentioned relationship shows that MTBF is one of important metric for reliability. MTBF itself is calculated as the average time between failures of the system and typically assuming that the failed system is immediately repaired. It should be noted that MTBF starts to be calculated after the first failure happens, thus the operating hours of the system before it fails is not taken into account. Mathematically, MTBF is calculated as the sum of the up time before the system goes down (t_{down} - t_{up}) divided by the total number of failures:

$$MTBF = \frac{\sum \left(t_{down} - t_{up}\right)}{\sum failures}$$

In principal, there are two basic definition of failure of such system:

- The termination of the ability of the product as a whole to perform its required function
- The termination of the ability of any individual component to perform its required function but not the termination of the ability of the product as a whole to perform

Availability of a system can be considered a specific time period when the system is up and running. Availability is a major issue for considering the delivered Service Level Agreement (especially for commercial services) and thus it can be measured to as to be maximum. It is a common metric to have approximately more than 99% uptime within a calendar year. 1% downtime is also including the schedule shutdowns for maintenance reasons.

This evaluation will assess how available the system is. This will be identified verifying whether the system is capable to perform according, or better, to the following thresholds:

- Acceptable level of system recovery time after a failure
- Acceptable level of battery lifetime
- Acceptable recharging procedure time duration

Availability of the system is determined by the MTBF (Mean Time Between Failures) and MTTR (Mean Time To Recover/Repair) with the following relationship:

$$Availability = \frac{MTBF}{\left(MTBF + MTTR\right)}$$

According to the above relation, MTTR is another important parameter besides MTBF that has been mentioned earlier. MTTR is the expected time to recover a system from a failure. This may include the time it takes to diagnose the problem, to get the technician onsite, and finally repair the problem physically.

7.2.3.1 Requirements summary:

- The SMART system should have a high availability and reliability (e.g. more than 99%) that can be monitored, measured and audited.
- In case of failures, measures have to be taken in order to overcome these in short notice and additional measures for preventing their occurrence.

7.2.4 Manageability and flexibility

This non-functional requirements define how easy is to manage the whole system and how flexible the system is for any changes. This will be identified verifying whether the system is capable to perform according to the following questions:

- How easy is it to add/delete new users?
- How easy is it to add/delete new services?
- How easy is it to release a backward compatible new version of the system?
- How easy is it to update any of the attributes of the system?
- How easy is it to add/delete new sensors?
- How easy is it to add/delete new virtual sensors?
- How easy is it to interface to more social networks?

7.2.4.1 Requirements summary:

- The SMART system should have a high manageability and flexibility even for users that are not considered experts.
- Common management attributes such as add/delete/update should be intuitive and easy to be performed.

7.2.5 Modularity

These non-functional requirements define the ability of the system to comprise several compents that are self-standing and consistent by design. This enables the replacement of compenents with new ones where changes in various features might be desirable. In general, SMART will have to address the following criteria:

- Is any SW and HW module consistent in its functionality?
- Can a module be replaced without influencing the functionality of the others as long as the interfaces remain unchanged?

7.2.5.1 Requirements summary:

- The SMART modularity level should allow enough independence of all modules so as if any module needs to be replaced, this will have no consequences to the other modules.

7.2.6 Security

SMART system has to be designed to be a secure system and protect that data and messages exchanged within the participants from any disclosure. Security is always considered as a set of aspects (security primitives) that comprise the whole notion of secure, safe and protected item.

The following aspects have been considered for technical evaluation:

- **Authentication** – what is the authentication mechanism used and how strong is it?
- **Authorisation** – access to services must be controlled based on authorisation policies attached to each service. It should accommodate various access control models and implementations. How easy/difficult is it to get permission which one is not supposed to get?
- **Privacy** – Privacy is considered as the aspect that when two peers exchange messages, those messages cannot be seen by others. In the same level, data that are referring one person cannot be accessed by any other person.

7.2.6.1 Requirements summary:

- SMART should cover with state of the art technologies all the aforementioned security aspects.

7.2.7 Open source requirements

As was stated in the section **Error! Reference source not found.**, the open source community is a key element for the success of SMART in the long term. The open source SMART framework will be successful if it can easily be used and reused in the future, both within and outwith of the SMART consortium. For instance, organizations not involved in the SMART project may wish to expose an edge node to an existing SMART search engine on the Internet. On the other hand, an organization may wish to deploy their own instance of the SMART search engine. In both cases, the SMART framework must be able as open source, such that is can be deployed by any organization. Indeed, such organization may use different operating systems that differ from the standard SMART development environment, therefore the SMART framework should aim to be operating system-agnostic. As companies may develop applications built upon the SMART framework, the open source license used should support their commercialization aims.

Hence, key non-functional requirements are that the SMART framework must be available as open source and portable. The license must be compatible with the commercialization plan of the SMART project.

7.2.7.1 Requirements summary:

- The edge node component of the SMART framework should be open source.
- The search engine component of the SMART framework should be open source.
- The components of SMART must be reusable in the development of new multimedia search frameworks.
- The components of SMART should be portable across major operating systems.
- The license of the open source framework should support the commercialization aims of SMART.

7.2.8 Openness and Extensibility Requirements

Unforeseen applications may be built using the SMART framework. For instance, an organization may connect to an edge node new types of sensors that have not been foreseen. They may develop additional A/V processing algorithms, or event detection approaches that are not part of the open source SMART framework. Furthermore, an edge node may be deployed that does not deploy the standard SMART open source software – for this reason, the API that defines communication between the edge node and the search engine should be an open standard, built upon existing open standards where possible.

On the search engine component, a deployer of SMART may wish the search engine to identify different events by using different models of interestingness. Furthermore, a use case application developer may wish to build unforeseen applications upon a SMART search engine, so the query language of the search engine should be rich enough for complex queries to be supported. As the use case applications interact programmatically, the API of the SMART search engine should be an open standard, built upon existing open standards where possible.

7.2.8.1 Requirements summary

- The edge node component of the SMART framework should be extensible to new unforeseen types of sensors.
- The edge node component of the SMART framework should be extensible to unforeseen A/V processing algorithms.

- The edge node component of the SMART framework should be extensible to unforeseen event detection approaches.
- The edge node to search engine API should be an open standard, built upon other existing open standards where possible.
- The search engine component of the SMART framework should be extensible to unforeseen search models of interestingness.
- The search engine component of the SMART framework should be extensible to input from edge nodes exposing unforeseen types of sensors, A/V processing algorithms or event detection approaches.
- The search engine component of the SMART framework should support a rich query language, such that unforeseen queries can be easily supported.
- The search engine component of the SMART framework should export an API for use case applications that is an open standard, built upon existing open standards where possible.

8 Requirements Consolidation

This Section provides a consolidated view of the aforementioned described requirements so as to conclude to a unique coding of all individual requirements in order to be tracked throughout the lifecycle of the project. The following table gives an overview of the elicited requirements of the SMART project.

Category: R1 Functional Requirements				
Group 1: Integration and application level functional requirements				
Code	**Description**	**Mandatory**	**Dependencies /Comments**	**Relevant Stakeholders**
R1.1.1	SMART should reuse existing open source software and tools, where it is appropriate and possible according the license.	Yes		Service Integrators and Open source community
R1.1.2	The architecture of SMART must be layered, providing separation of concerns	Yes		Service Integrators
R1.1.3	The components of SMART must be developed according accepted good programming practice.	Yes		All
R1.1.4	The components should be developed using proven and trusted languages. The open components should use a language with good open source community support.	Yes		Service Integrators and Providers Open source community
R1.1.5	The open source components of the SMART framework must be released under an Open Source License.	No		End Users Open source community
Group 2: Video processing functional requirements				
R1.2.1	SMART system should support at least the technologies marked with OK in Table 7.1.2	Yes	See table 7.1.2	Infrastructure Providers
R1.2.2	Video resolution should be at least VGA for visual scene analysis. Face and person tracking require much higher resolutions, which the selected camera can offer.	Yes	Camera technical specifications. Dependency with DPA	Infrastructure Providers / End users

R1.2.3	Video processing algorithms in SMART should be real-time (there is no storing of data, unless a specific event is detected). The minimum frame rate for significant inter-frame correlation is 10 FPS, hence the algorithms need to process frames this fast.	Yes		Infrastructure Providers / End users
R1.2.4	A constant latency of a few seconds is tolerated by the system.	Yes	Dependency with physical networking	Infrastructure Providers / End users
Group 3: Acoustic processing functional requirements				
R1.3.1	SMART system should support audio event classification.	Yes		Infrastructure and Service Providers, End users
R1.3.2	Audio input for the audio event classification should have a bandwidth of at least 16 KHz.	Yes		Infrastructure and Service Providers, End users
R1.3.3	The system should tolerated delays in the audio event classification of several minutes.	Yes		Infrastructure and Service Providers, End users
R1.3.4	SMART system should support speech transcription for voice messages	No		Infrastructure and Service Providers, End users
R1.3.5	Speech samples for the transcription should have a bandwidth of at least 16 KHz.	Yes		Infrastructure and Service Providers, End users
R1.3.6	The system should tolerated delays in the speech transcription which are similar in length to the length of the speech samples.	Yes		Infrastructure and Service Providers, End users
R1.3.7	SMART system should support speaker verification for voice messages.	No		Infrastructure and Service Providers, End users
R1.3.8	The system should tolerated delays in the speaker verification which are similar in length to the length of the speech samples.	Yes		Infrastructure and Service Providers, End users
Group 4: Sensor data processing functional requirements				
R1.4.1	It should be able map distributed sensor data from several data streams to perform	Yes	R.1.4.5	Infrastructure and

	complex event recognition, including atomic events, compound events, and situa-tions.			Service Providers
R1.4.2	It should be capable of learning patterns of significance from sensor data over vary-ing timescales, using machine learning algorithms.	Yes	Number of pat-terns must be limited though.	Infrastructure and Service Providers
R1.4.3	It should implement a set of fair information practices, implemented a user-centric policy on how UGC data, and information derived from it, is to be collected, used and disposed of.	Yes		Infrastructure and Service Providers, End users
R1.4.4	It should support UGC search in both push- and pull-modes, i.e. in response to spe-cific user queries and subscription mechanisms.	Yes		Infrastructure and Service Providers, End users
R1.4.5	SMART system should support a wide set of non-AV sensors.	Yes	R.1.4.1	All
R1.4.6	SMART sensors should adhere to the SensorML framework.	Yes		Infrastructure and Service Providers
R1.4.7	SMART should support for multiple social networks and for filters over social net-works.	Yes		Service Providers, End users
R1.4.8	SMART should support various “social sensors” (meaning: virtual sensors over vari-ous social net-works).	Yes	R1.4.8	Service Providers
Group 5: Widgets, mashups and APIs functional requirements				
R1.5.1	SMART access widgets should be designed using languages usually adopted by the open-source community. To extend the use of the platform to the mobile handsets, HTML5 could be a good approach to SMART widgets programming.	Yes		Infrastructure and Service Providers
R1.5.2	The SMART platform should be extensible by providing clear API to create specific add-ons;	Yes		Infrastructure and Service Providers, Integrators.
R1.5.3	The data stored should be accessible for post-processing operations. Post pro-cessing should be performed using existing SMART components or user created add-ons.	Yes		Infrastructure and Service Providers, Integrators.
R1.5.4	Data captured by the SMART sensors should be able to be contributed to the Linked Data framework.	Yes		Infrastructure and Service Providers, Integrators.
R1.5.5	SMART mashup technologies should be open and flexible to incorporate the func-	Yes		Infrastructure and

	tionalities implied by the SMART use cases.			Service Providers, Integrators, End users
R1.5.6	SMART should facilitate data linking and interoperation with the Linked Data initiative.	Yes		Infrastructure and Service Providers, Integrators, End users
Group 6: Search engine functional requirements				
R1.6.1	The SMART search engine should produce rankings of events, possibly in response to a user's query.	Yes		Service Providers and End users
R1.6.2	The search engine gathers new input from sensors via edge nodes, and must be robust to communication failures with edge nodes.	Yes		Service providers
R1.6.3	Input from sensors via the edge nodes should be used to model the importance of these events. The current data from a sensor may be compared to the previous background history of that sensor.	Yes	Dependency with the selected machine learning algorithms.	Infrastructure and Service Providers
R1.6.4	As recent events are likely to be more important, the search engine should regularly index new data arriving from edge nodes.	Yes		Service Providers
R1.6.5	These event rankings should be accessibly through an API available to use case applications.	Yes		Service Providers
Group 7: Media Data management				
R1.7.1	The Media Data component will be able to present multimedia information to the users.	Yes		Infrastructure and Service Providers, Service Integrators
R1.7.2	The video information will be adapted to the most common types of terminals.	Yes	Dependency with the terminals.	Infrastructure and Service Providers, Service Integrators
R1.7.3	The Media Data component will be able to store video for a predefined time.	Yes	Dependency with the storage capacity.	Infrastructure and Service Providers, Service Integrators

R1.7.4	Media Data component should respect the legal and ethical aspects.	Yes		Infrastructure and Service Providers, Service Integrators.
Category: R2 Non-Functional Requirements				
Group 1: Security & Privacy				
R2.1.1	The SMART platform should be capable of managing different users profiles distinguishing between operators;	Yes		All
R2.1.2	The SMART platform, in accordance with privacy policies, should be capable to provide, if necessary, evidence of a crime. The platform should store privacy covered data in a protected way. Access to protected data should be possible only to authorized operators and only in case of a crime event;	Yes		All
R2.1.3	The SMART platform, in accordance to local security policies, should be capable of applying digital signature to content before storing them;	Yes		All
R2.1.4	The applications and technologies used in SMART must respect all regulations concerning the ethical aspects, especially those related with data protection and privacy.	Yes		All
R2.1.5	SMART should cover with state of the art technologies all the aforementioned security aspects.	Yes		All
Group 2: Scalability				
R2.2.1	The SMART system must be able to scale with respect to more input sensors. In this way, the number of edge nodes within a SMART system should not be limited, such that more edge nodes can be deployed to handle more sensors.	Yes	R1.2.x and R1.3.x	Infrastructure Providers, Service Integrators, End users
R2.2.2	The processing algorithms deployed by each edge node should have sufficiently low latency, such that the edge node does not “fall behind”, and can always send the latest data to the search engine.	Yes	R1.2.x and R1.3.x	Infrastructure Providers, Service Integrators, End users
R2.2.3	The search engine should be able to scale to more input data from more edge nodes, by vertically scaling the search indexing processes across multiple search engine machines.	Yes		Infrastructure Providers, Service Integrators
R2.2.4	The search engine should be able to respond to user queries from the use cases, and moreover should be able to handle multiple users of the use cases at once, such	Yes		Infrastructure Providers, Service In-

	that results are obtained quickly even under load.			tegrators, End users
Group 3: Performance				
R2.3.1	The SMART system deploys various audio/visual processing algorithms that are inexact, uncertain processes. However, they should be regularly evaluated during development, such that they are shown to be accurate with real data.	Yes	R1.2.x and R1.3.x	Infrastructure Providers, Service Integrators
R2.3.2	Similarly, the ranking of documents (events in the SMART case) by a search engine is also an uncertain process. The search engine will be appropriately trained and evaluated to ensure its effectiveness.	Yes		Infrastructure Providers, Service Integrators
Group 4: Reliability and Availability				
R2.4.1	The SMART system should have a high availability and reliability (e.g. more than 99%) that can be monitored, measured and audited.	Yes		Infrastructure Providers, Service Integrators
R2.4.2	In case of failures, measures have to be taken in order to overcome these in short notice and additional measures for preventing their occurrence.	Yes		Infrastructure Providers, Service Integrators
Group 5: Manageability and flexibility				
R2.5.1	The SMART system should have a high manageability and flexibility even for users that are not considered experts.			Infrastructure Providers, Integrators, End users
R2.5.2	Common management attributes such as add/delete/update should be intuitive and easy to be performed.	Yes		Infrastructure Providers, Integrators, End users
Group 6: Modularity				
R2.6.1	The SMART modularity level should allow enough independence of all modules so as if any module needs to be replaced, this will have no consequences to the other modules.	Yes		Infrastructure Providers, Service Integrators
Group 7: Open source				
R2.7.1	The components of SMART must be reusable in the development of new multimedia search frameworks.	Yes	R1.1.x	Infrastructure Providers, Service Integrators, Open source Community

R2.7.2	The components of SMART should be ideally extensible beyond their original design and purpose.	Yes	R1.1.x	As above
R2.7.3	The components of SMART should be ideally portable across major operating systems.	Yes	R1.1.x	As above
R2.7.4	The components of SMART should be interoperable with other services implementing common and open standards	Yes	R1.1.x	As above
Group8: Openness and Extensibility Requirements				
R2.8.1	The edge node component of the SMART framework should be extensible to new unforeseen types of sensors.	Yes	Minor adaptation might be required	Infrastructure and Service Providers, Service Integrators,
R2.8.2	The edge node component of the SMART framework should be extensible to unforeseen A/V pro-cessing algorithms.	Yes	As above	Infrastructure and Service Providers, Service Integrators,
R2.8.3	The edge node component of the SMART framework should be extensible to unforeseen event detection approaches.	Yes	As above	Infrastructure and Service Providers, Service Integrators,
R2.8.4	The edge node to search engine API should be an open standard, built upon other existing open standards where possible.	Yes		Infrastructure and Service Providers, Service Integrators,
R2.8.5	The search engine component of the SMART framework should be extensible to unforeseen search models of interestingness.	Yes	As above	Service Providers, End users
R2.8.6	The search engine component of the SMART framework should be extensible to input from edge nodes exposing unforeseen types of sensors, A/V processing algorithms or event detection approaches.	Yes	As above	Infrastructure and Service Providers, Service Integrators
R2.8.7	The search engine component of the SMART framework should support a rich query language, such that unforeseen queries can be easily supported.	Yes		Service providers, End users, Open Source Community
R2.8.8	The search engine component of the SMART framework should export an API for use case applications that is an open standard, built upon existing open standards where possible.	Yes		As above

9 Conclusions

Requirements analysis plays an important role for the whole lifecycle of the SMART project. It is the input for the SMART specification and overall architecture as well as for the validation of the final system and its evaluation against to the desired functionality.

In this document we have presented the requirements analysis methodology along with the defined functional and non-functional requirements of the SMART system.

The analysis starts with a definition of the SMART concept, including an overview of the use cases and the stakeholders involved so as to understand the context of the project and identify the various stakeholders. The stakeholders in SMART represent a holistic value chain: Infrastructure Providers, Service Providers, Service Integrators, End users and the Open Source Community.

As a next step, the consortium partners give an overview of the technologies that are going to be involved in the project and the perspective of using them in order to implement the SMART concept. Similar projects and background from previous projects are also mentioned in order to present the state of the art and the previous achievements that can be used as a starting point. In the sequel the elicitation of requirements is derived from the definition of the desired functionality of the SMART system given from the perspectives of the functional components and the non-functional attributes that the end system has to expose. The SMART system should have specified functionality to facilitate the use cases and a rich set of non functional requirements such as scalability, reliability, availability, manageability, flexibility, modularity, openness etc.

The document has concluded with a consolidated and coded requirements list that will be the reference for the design activities and the implementation and validation of the SMART system. The list has been divided into groups of requirements, indicating dependencies between the various requirements but also the relevant stakeholders that are affected.

The current deliverable will be one of the main reference documents for the design and specification of the SMART system. In particular, the deliverables D2.2 SMART Use Cases Specifications and D2.3 Multimedia Search Framework Open Architecture and Technical Specifications are the ones that will take this document most into account for the further development of the project.

10 BIBLIOGRAPHY AND REFERENCES

[1] A. Pnevmatikakis and L. Polymenakos, 'Robust Estimation of Background for Fixed Cameras', 15th International Conference on Computing (CIC2006), Mexico City, Mexico, Nov. 2006.

[2] A. Stergiou, G. Karame, A. Pnevmatikakis and L. Polymenakos, 'The AIT 2D face detection and tracking system for CLEAR 2007', in R. Stiefelhagen, R. Bowers and J. Fiscus (eds.): *Multimodal Technologies for Perception of Humans (CLEAR 2007)*, Lecture Notes in Computer Science 4625, Springer-Verlag, Berlin Heidelberg, pp. 113-125, June 2008.

[3] N. Katsarakis, F. Talantzis, A. Pnevmatikakis and L. Polymenakos, 'The AIT 3D Audio / Visual Person Tracker for CLEAR 2007', in R. Stiefelhagen, R. Bowers and J. Fiscus (eds.): *Multimodal Technologies for Perception of Humans (CLEAR 2007)*, Lecture Notes in Computer Science 4625, Springer-Verlag, Berlin Heidelberg, pp. 35-46, June 2008.

[4] A. Stergiou, A. Pnevmatikakis and L. Polymenakos, 'The AIT outdoor tracker for vehicles and pedestrians in CLEAR 2007', in R. Stiefelhagen, R. Bowers and J. Fiscus (eds.): Multimodal Technologies for Perception of Humans (CLEAR 2007), Lecture Notes in Computer Science 4625, Springer-Verlag, Berlin Heidelberg, pp. 148-159, June 2008.

[5] G. Karame, A. Stergiou, N. Katsarakis, P. Papageorgiou and A. Pnevmatikakis, '2D and 3D Face Localization for Complex Scenes', *IEEE Int. Conf. on Advanced Video and Signal based Surveillance (AVSS 2007)*, London, UK, Sept. 2007.

[6] M. Andersen, R. Andersen, N. Katsarakis, A. Pnevmatikakis and Z.-H. Tan, "Three-dimensional adaptive sensing of people in a multi-camera setup", special session on Person tracking for assistive working and living environments, EUSIPCO 2010, Aalborg, Denmark, Aug. 2010.

[7] A. Pnevmatikakis, N. Katsarakis, P. Chippendale, Cl. Andreatta, S. Messelodi, C. Modena and F. Tobia, "Tracking for Context Extraction in Athletic Events", International Workshop on Social, Adaptive and Personalized Multimedia Interaction and Access (SAPMIA 2010), ACM Multimedia, Florence, Italy, Oct. 2010.

[8] Ch. Patrikakis, A. Pnevmatikakis, P. Chippendale, M. Nunes, R. Cruz, S. Poslad, Z. Wang, N. Papaoulakis, P. Papageorgiou, 'Direct your personal coverage of large athletic events', IEEE Multimedia, DOI 10.1109/MMUL.2010.69, November 2010.

[9] F. Talantzis, A. Pnevmatikakis, and A. G. Constantinides, 'Audio-Visual Active Speaker Tracking In Cluttered Indoors Environments', Journal of Systems, Man, and Cybernetics–Part B: Cybernetics, pp. 799-807, June 2008.

[10] F. Talantzis, A. Pnevmatikakis, and A. G. Constantinides, 'Audio-Visual Person Tracking: A Practical Approach', World Scientific Publication Company, Imperial College Press, Dec. 2011.

[11] A. Pnevmatikakis and L. Polymenakos, 'Subclass Linear Discriminant Analysis for Video-Based Face Recognition', Journal of Visual Communication and Image Representation, Vol. 20, Issue 8, pp. 543-551, Nov. 2009.

[12] A. Stergiou, A. Pnevmatikakis and L. Polymenakos, 'The AIT multimodal person identification system for CLEAR 2007', in R. Stiefelhagen, R. Bowers and J. Fiscus (eds.): *Multimodal Technologies for Perception of Humans (CLEAR 2007)*, Lecture Notes in Computer Science 4625, Springer-Verlag, Berlin Heidelberg, pp. 221-232, June 2008.

[13] Z. Kons et al. "On Feature Extraction for Voice Pathology Detection from Speech Signals", Proceedings of Afeka speech processing conference 2011, June 2011.

[14] Sorin, A. et al, "Speech processing and retrieval in a personal memory aid system for the elderly". Proc. ICASSP 2011, May 2011

[15] Aronowitz H., "Unsupervised Compensation of Intra-Session Intra-Speaker Variability for Speaker Diarization", in Odyssey, 2010.

[16] J. M. Huerta et al, "RTTS: Towards Enterprise-Level Real-Time Speech Transcription and Translation Services", Proceedings of INTERSPEECH 2009, Sept 2009.

[17] Aronowitz H. et al, "New developments in Voice Biometrics for User Authentication", Proc. Interspeech, 2011, Aug 2011.

[18] J. Pitt, J. Schaumeier, A. Artikis: The Axiomatisation of Socio-Economic Principles for Self-Organising Systems. SASO 2011: 138–147 (2011)

[19] J. Pitt, D. Ramirez-Cano, M. Draief, A. Artikis: Interleaving multi-agent systems and social networks for organized adaptation. Computational & Mathematical Organization Theory 17(4): 344–378 (2011)
[20] J. Pitt, L. Kamara, M. J. Sergot, A. Artikis: Voting in Multi-Agent Systems. Comput. J. 49(2): 156–170 (2006)
[21] T. Regner, J. A. Barria, J.. Pitt, B. Neville: Governance of digital content in the era of mass partici-pation. Electronic Commerce Research 10(1): 99–110 (2010)
[22] J. Farrer, P. Goulev and J. Pitt, The Emotive Episode: An Investigation into User Re-sponse to Sustainable Issues in Fashion/Textiles and Affective Computing, Proceedings of Sustainable Inno-vation 06: Global Challenges, Issues and Solutions, Chicago, pp54–60 (2006).
[23] H. Amini, M. Draief, M. Lelarge, Marketing in random Networks, Conference on Network Control and Optimization (Net-Coop), Paris, LNCS 5425, pp 17-25 (2009)
[24] M. Draief and M. Vojnovic, Convergence Speed of binary Interval consensus, IEEE Int. Conf. Computer Communications (INFOCOM), pp 1792-1800 (2010)
[25] M. Draief and M. Vojnovic, Bargaining in Exchange Networks, Allerton Conference on Com-munication, Control and Computing, pp 1303-1310, (2010)
[26] B. He, I. Ounis: A study of parameter tuning for term frequency normalization. In Proceedings of CIKM 2003: 10-16
[27] B. He, I. Ounis: Inferring Query Performance Using Pre-retrieval Predictors. In Proceedings of SPIRE 2004: 43-54
[28] B. He, I. Ounis: Combining fields for query expansion and adaptive query expansion. Inf. Process. Manage. J, 43(5): 1294-1307 (2007)
[29] N. Limsopatham, R. Santos, C. Macdonald and I. Ounis. Disambiguating Biomedical Acronyms Us-ing EMIM. In Proceedings of SIGIR 2011.
[30] C. Macdonald and I. Ounis. Combining Fields in Known-Item Email Search.In Proceedings of ACM SIGIR'06
[31] C. Macdonald and I. Ounis. Voting for Candidates: Adapting Data Fusion Techniques for an Expert Search Task. In Proceedings of ACM CIKM'06.
[32] C. Macdonald and I. Ounis. Key Blog Distillation: Ranking Aggregates. In Proceedings of CIKM 2008.
[33] C. Macdonald and Iadh Ounis. Learning Models for Ranking Aggregates. In Proceedings of ECIR 2011.
[34] C. Macdonald, N. Tonellotto and I. Ounis. Upper-bound Approximations for Dynamic Pruning. ACM Transactions on Information Systems. 29(4), 2011.
[35] R. McCreadie, C. Macdonald and I. Ounis. MapReduce Indexing Strategies: Studying Scalability and Efficiency. In Information Processing & Management J. 2010.
[36] R. McCreadie, C. Macdonald and I. Ounis. A Learned Approach for Ranking News in Real-time us-ing the Blogosphere. In Proceedings of SPIRE 2011.
[37] R. McCreadie, C. Macdonald, R. L. T. Santos, I. Ounis. University of Glasgow at TREC 2011 - Ex-periments with Terrier in Crowdsourcing, Microblog, and Web Tracks. In Proceedings of TREC 2011.
[38] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald and C. Lioma. Terrier: A High Performance and Scalable Information Retrieval Platform. In Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006).
[39] J. Peng, C. Macdonald, Ben He, Vassilis Plachouras, I. Ounis. Incorporating Term Dependency in the DFR Framework. In Proceedings of SIGIR 2007. July 2007. Amsterdam, the Netherlands. 2007.
[40] J. Peng, Craig Macdonald, Ben He, Iadh Ounis. A Study of Selective Collection Enrichment for En-terprise Search. In Proceedings of CIKM 2009.
[41] J. Peng, C. Macdonald, I. Ounis. Learning to Select a Ranking Function. In Proceedings of ECIR 2010.
[42] V. Plachouras, I. Ounis: Usefulness of hyperlink structure for query-biased topic distillation. In Pro-ceedings of SIGIR 2004: 448-455
[43] R. Santos, C. Macdonald and I. Ounis. Intent-Aware Search Result Diversification. In Proceedings of SIGIR 2011.
[44] Exploiting Query Reformulations for Web Search Result Diversification R. Santos, C. Macdonald, I. Ounis. In Proceedings of WWW 2010.
[45] http://www.nessi-europe.com

[46] http://www.future-internet.eu/
[47] http://www.nem-initiative.org/
[48] http://www.eos-eu.com
[49] http://www.isis-europe.eu/
[50] http://cordis.europa.eu/search/index.cfm?fuseaction=proj.document&PJ_RCN=5269020
[51] http://www.gama-gateway.eu/
[52] Roel J. Wieringa and R.B. Feenstra (editors). "Information Systems: Correctness and Reusability - Selected Papers from the IS-CORE Workshop", World Scientific Publishing Co Pte Ltd, Singapore (1995)
[53] http://www.ist-live.org/
[54] http://www.alisproject.eu/
[55] http://www.myedirector2012.eu/
[56] http://www.qualipso.org/