PrivAcy pReserving Infrastructure for Surveillance

Deliverable D4.3
SALT Compliant Processes Guidelines for Use Cases

Project:	PARIS
Project Number:	SEC-312504
Deliverable:	D4.3
Title:	SALT Compliant Processes Guidelines for Use Cases
Version:	v1.2
Date:	13/06/2014
Confidentiality:	Public
Authors:	Antonio Maña (UMA) Francisco Jaime (UMA) Fernando Casado (UMA) Marioli Montenegro (UMA) Domingo Perea (UMA) Zhendong Ma (AIT) Christophe Jouvray (Trialog) Mathias Bossuet (Thales) Denis Butin (INRIA) María Cinta Saornil (Visual Tools)

Part of the Seventh Framework Programme
Funded by the EC - DG INFSO

Table of Contents

Document History

Version	Status	Date
v0.1	Document structure and initial content	18/03/2014
v0.2	Added section 2 and substitution of the FP7 logo	24/03/2014
v0.3	Added sections 3.1 and 3.2.1	07/05/2014
v0.4	Added section 3.5 and section 2 revised	15/05/2014
v0.5	Added introduction and typos correction	16/05/2014
v0.6	Added content to section 3.2	20/05/2014
v0.7	Rearrangement of figures. Added section 3.4	26/05/2014
v0.8	Added section 3.3	27/05/2014
V0.9	Chg. doc. structure, added section 4.3, doc. revised	29/05/2014
v0.10	Added section 4.2	02/06/2014
v1.0	Ready for internal review	05/06/2014
v1.1	Final version	12/06/2014
v1.2	Minor revision	13/06/2014

Approval		
	Name	**Date**
Prepared	Francisco Jaime	13/06/2014
Prepared	Marioli Montenegro	13/06/2014
Reviewed	Fanny Coudert	12/06/2014
Reviewed	Zhendong Ma	11/06/2014
Authorised	Antonio Maña	13/06/2014

Circulation	
Recipient	**Date of submission**
Project partners	13/06/2014
European Commission	13/06/2014

Executive Summary

This document provides a refined and expanded explanation of the SALT general process, focusing on its application to the two main use cases covered by the PARIS project, i. e. video search technology and biometrics technology.

It begins with the description of a practical application of a SALT process, showing the key parts involved within the process, the connections among them, and what are their functionalities according to their inputs, the expected results and the type of user who is following the process guidelines.

How this SALT process has been refined, and what new parts and functionalities have been included, is also covered in the following text. Due to its importance within the whole project, a previous and brief explanation of the SALT framework is provided for the sake of clarity. Then we address the SFMT (SALT Framework Management Tool) and the way it deals with the information gathered within the SALT framework, just before providing a view of the SALT general process and how its use can affect to surveillance systems based on video search and biometrics technologies.

The last part of the document is dedicated to the possibility of specializing the SALT general process to a given type of surveillance system (video search or biometrics). That is, look for possible aspects or steps of the process that could be specifically adapted in a way or another to better fit video search systems or biometrics systems.

List of Figures

List of Tables

Abbreviations and Definitions

Abbreviation	Definition
API	Application Programming Interface
DNA	Deoxyribonucleic acid
EDPS	European Data Protection Supervisor
EU	European Union
FOV	Field of Views
HMI	Human Machine Interface
HTML	HyperText Markup Language
IBM	International Business Machines
JSP	JavaServer Pages
LODP	Data Protection Organic Law (Ley Orgánica de Protección de Datos)
MILS	Middleware for Large Scale Surveillance
NVR	Network Video Recorder
ÖBB	Austrian Railway Organization (Österreichische Bundesbahnen)
OCL	Object Constraint Language
PAERIS	PrivAcy-by-design EngineeRIng aSsistant
PARIS	PrivAcy pReserving Infrastructure for Surveillance
PC	Personal Computer
PET	Privacy Enhancing Technology
PIA	Privacy Impact Assessment
S3	Smart Surveillance Suite
SALT	Socio-contextual, ethicAl, Legal, Technological
SF	SALT Framework
SFMT	SALT Framework Management Tool
SSE	Smart Surveillance Engine
UML	Unified Modeling Language
VCA	Video Content Analytics
VMS	Video Management System
WP	Work Package
XML	eXtensible Markup Language

1 Introduction

We have already provided a description of the SALT compliant process in the PARIS project deliverable D4.2 “SALT Compliant Processes Definition”. We described the process lifecycle and the guidelines describing all the steps covered from beginning to end. However, that was the first version of the SALT compliant process, and although the process main behavior and interests remain the same, after a few more months of work we have produced a refined version of the general SALT compliant process, which is presented and described in this document.

Even though the process objectives are kept, with the SALT framework located at its core, this deliverable shows how we have refined some functionalities and how some others have been added, like the possibility of automatically check that privacy and accountability concerns haven been properly taken into account at design time (or whenever it is not possible, the system ensures that the decisions and actions taken are documented to assist a human user to check it).

In order to keep the reader of this document focused and provide him with an overall view of the SALT compliant process, we start by describing how a practical application of the general process would look like, what steps it covers, their objectives and how they are addressed following the SALT process guidelines.

The SALT framework is at the heart of the SALT compliant process, as mentioned above, hence even though it was described in details in the deliverable D2.2 “Structure and Dynamics of SALT Frameworks”, we provide a brief description to keep the reader informed with the fundamentals. We then follow with the introduction of the SFMT (SALT Framework Management Tool), the application in charge of interacting with the SALT framework. Thanks to this tool, future users will be able to add new content to the framework (this is the point of view of experts) and to retrieve it when necessary (this is mainly the point of view of system designers).

The SFMT has been addressed in two different ways: a web-based approach integrated with the SALT repository (SALT repositories are the hosts where the SALT framework is physically stored), and a stand-alone version that can work offline. Both solutions are introduced in this deliverable, focusing in the system designer’s point of view, which is the tendency of this document due to its implication with use cases. Moreover, the expert’s point of view has been approached in the deliverable D2.2.

Example use cases are provided in order to illustrate the usage of the SALT compliant process regarding surveillance systems based on video search technology, as well as surveillance systems bases on biometrics technology. Here we will be able to see how surveillance systems of both types of technology are affected by the use of the SALT compliant process, and how privacy and accountability aspects are taken into account in order to improve their system designs.

Finally, we also get into how the general SALT compliant process could be specialized regarding the two types of surveillance technologies covered by the PARIS project (video surveillance and biometrics), what type of information/concerns are more relevant for each type of system and

whether we should consider or not, different (specialized) ways of applying the SALT process guidelines to each one. Due to their impact on the process, accountability concerns are also taken into account in this specialization guides.

This document assumes the 3-stage process, defined in WP2 and shown in Figure 1, as the base for this work.

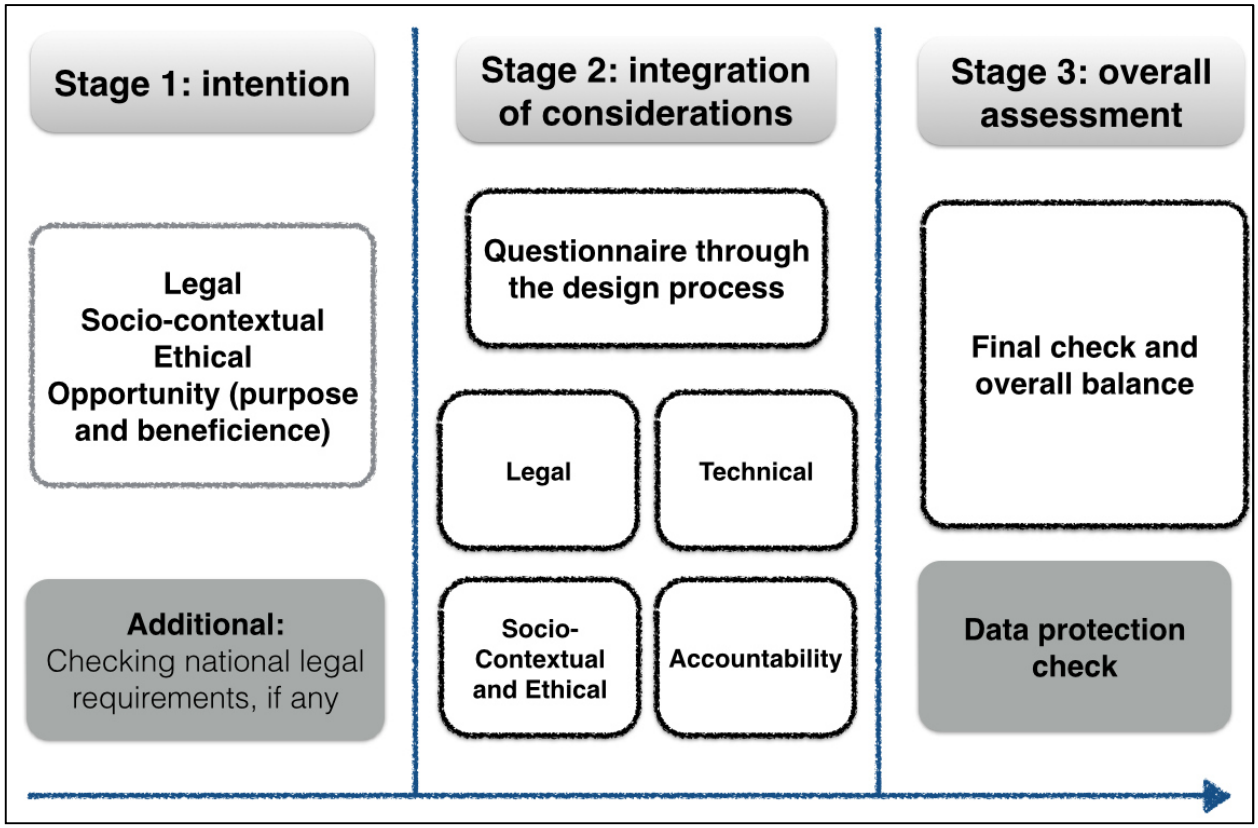


Figure 1Three stage process for SALT Framework

2 Practical application of the SALT process

As it has been described in the PARIS project deliverable 4.2, the SALT compliant process includes not only the surveillance system design, but also the information acquisition and the information representation processes. Therefore, for a description of a practical application of the SALT compliant process, we will consider the whole process lifecycle.

Let us briefly remember the concepts of SALT reference and SALT framework. On one hand, the SALT reference is an entity where privacy-related and accountability-related information regarding surveillance systems is included. Thereby, each SALT reference contains a number of concerns from four different categories: socio-contextual, ethical, legal and technological. On the other hand, the SALT framework is the set of all SALT references, thus we can say it is the entity that contains all the knowledge. From a practical point of view, the SALT framework, and hence all the information that it contains, is stored into a repository (one or several repositories).

Therefore, the information contained within the SALT framework is crucial, since it is what it will be used in order to embed privacy and accountability features into the design of the surveillance systems under development. This means that, in first place, and independently of any particular surveillance system, the SALT framework has to be populated with the proper information.

The privacy-related and accountability-related information is typically provided by experts in one (or several) of the four possible categories. The methods used to gather this information will vary depending on the nature of the knowledge: analysis of data obtained from questionnaires given to a population, legal documents, technical reports, official interpretations of the legal framework for a given context, etc. Then, once the information is at hand, it is supplied to the SALT framework via the SFMT (SALT Framework Management Tool). This tool will guide experts, who do not need to know how the SALT framework is implemented, in the task of adding their knowledge to the SALT framework.

Practically speaking, all the information is digitally stored within a repository, which means that we need a digital representation of the knowledge in order to make it manageable (understandable) by a computer. This representation is ruled by a template, called the SALT template, created by someone with a proper knowledge of the SALT framework implementation, and guided by the experts' requirements. Initially, members of the PARIS consortium will be in charge of creating the SALT template. Besides, it is also possible to update the SALT template according to emerging needs for representing new knowledge provided by experts. That is, the SALT template, and hence the SALT framework and the information it contains, will be able to evolve as required.

At this point, once the repository containing the SALT framework is running and the privacy-related and accountability-related information is available, system designers can make use of it when creating a design of the surveillance system under development. For this to happen, system designers use the SFMT to provide the context of a particular surveillance system. These parameters will then be used by the SFMT as a filter to search those SALT references susceptible of being applied to the current system design. The repository will output the selected SALT references and the SFMT will show to the system designer the information

obtained. Thanks to this, system designers have access to privacy-related and accountability-related information before making a system design, having the possibility of applying that information (guided by the SALT framework) during the design phase of the surveillance system. By guiding system designers to take privacy and accountability aspects during the system design, this solution achieves the goal pursued by the PARIS project, i.e. bringing the system design close to be "SALT compliant" (that is, the design of the surveillance system takes into account a series of privacy and accountability requirements from the socio-contextual, ethical, legal and technological points of view). The process does becomes "SALT compliant", meaning that the actions and decisions taken by designers with regard to privacy and accountability aspects are made explicit. It however does not guarantee compliance with legal and ethical aspects, since this goes beyond the realm of technology and it may require an external check by an auditor. Nevertheless, the SALT process intends to facilitate the work of the auditor by contributing to the production of the relevant documentation of actions and decisions made by designers.

However, the possibilities of the SALT framework do not end here. The SFMT shows to system designers a textual description of the concerns contained within a SALT reference. This is a human-readable description that can be understood (and applied) by system designers regardless of their previous knowledge about the four possible areas of expertise. But in addition to this textual description, the concerns can also be provided by means of OCL (Object Constraint Language) rules. We will now see how these OCL rules can be useful.

The design of the surveillance system can be expressed by a UML diagram. At this point, in order not to force system designers to know about UML, we can provide higher level mechanisms that help them create the design, such as a UML profile. In this way, what system designers can see is a list of elements they are used to work with, such as cameras, NVRs, fingerprint scanners, servers, etc. Thanks to this solution they can produce a surveillance system design that has an inner UML representation, even though they may not know about this representation.

Having the UML representation of the system design, together with the OCL representation of the privacy-related and accountability-related information from the SALT references, it is possible to use an automated process (which can be implemented by a second tool) to check that the system design actually fulfills the OCL rules. In the case of all rules being fulfilled, we could say that the system design properly implements the SALT concerns, otherwise a message can be displayed informing the user about what rules were not fulfilled. Figure 1 depicts a graphical description of an application of the SALT process. The tool for checking the concerns implementation has been named PAERIS: PrivAcy-by-design EngineeRIng aSsistant.

Here we have a challenge: the creation of the OCL representation for the concerns included within the SALT references. It is clear that most of the SALT experts will not know how to produce the OCL representation for the concerns they include in the SALT references that they create (they may not even know about OCL at all). For this reason, we need the inclusion of an OCL expert who can translate the textual descriptions of the concerns provided by SALT experts to their corresponding OCL representations. This is what we have called *OCL translator* in Figure 1. However, we cannot guarantee that the OCL translator will always be available for the translation of all SALT references, since it will also need a good understanding of the concerns to be translated.

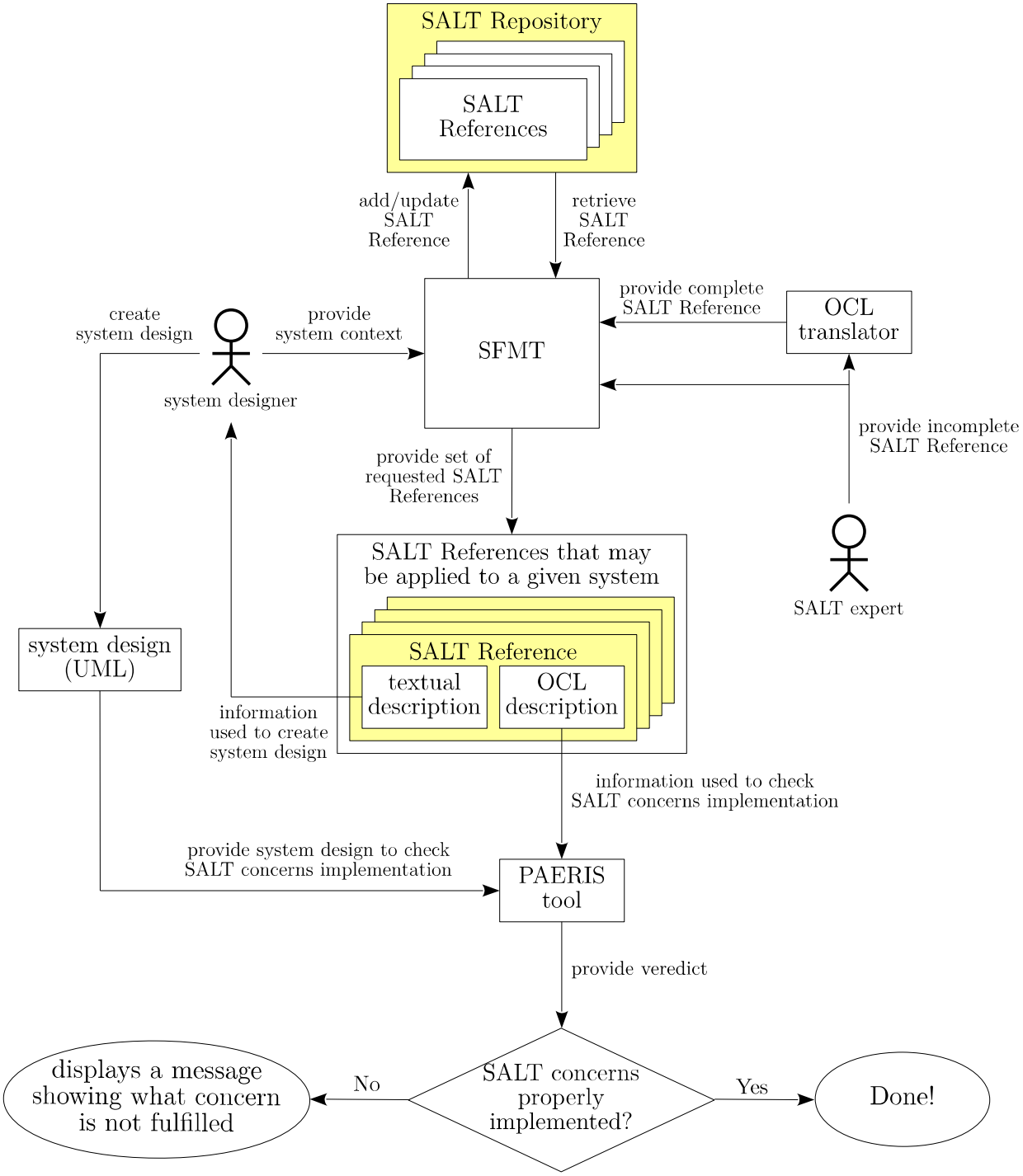


Figure 2. Practical application of the SALT process

This means that the SALT repository will store two types of SALT references:

1. Incomplete SALT references: these are references provided by SALT experts with a textual description of concerns, but lacking of an OCL representation.
2. Complete SALT references: these references are provided by the OCL translator (who previously received them from SALT experts). They include the textual description of concerns, as well as their OCL representation.

Let us analyze the two possibilities concerning the SALT process: in case system designers receive incomplete SALT references, they still can count on the textual descriptions. This is the information they really need at design time, and hence the process still fulfills the privacy-by-design and accountability-by-design approaches. On the other hand, in case complete SALT references are available, the PAERIS tool will be able, in addition, to check that privacy and accountability concerns have been properly integrated in the current system design.

As for the SALT compliance, we could say that a system design created using complete SALT references, whose concerns have been properly implemented and checked by the PAERIS tool, is a SALT compliant design. However, the use of incomplete SALT references and not being possible of using the PAERIS tool, does not necessarily means that the system design is not SALT compliant. In this case, the designer may have also properly taken into account the corresponding privacy and accountability concerns, hence resulting in a SALT compliant design, too. The only difference is that in this case it is a human user, assisted by the information provided by the SALT process, who checks the SALT compliance of the system design instead of a tool.

3 Refinement of a SALT general process

This section provides a brief description of the SALT framework, which has been slightly improved from the previous version with the inclusion of OCL rules (when possible). The SALT Framework Management Tool (SFMT) is also introduced, showing its main functionalities and the two implementations that are intended to be developed.

There are also a couple of use cases for surveillance systems based on video search and biometrics technologies, indicating how the SALT process affects both cases. And then, a description of a general SALT process follows.

3.1 Concept of SALT framework

The SALT framework can be considered as the core of the PARIS project, since it gathers all the privacy and accountability information coming from experts in the four different areas of expertise attached to the PARIS project: socio-contextual, ethical, legal and technological. This information is provided and retrieved via the SALT framework management tool (SFMT) and it is used during the SALTed design process (introduced in the project deliverable 4.2). Therefore, as it can be seen, the SALT framework is a key part of the PARIS project.

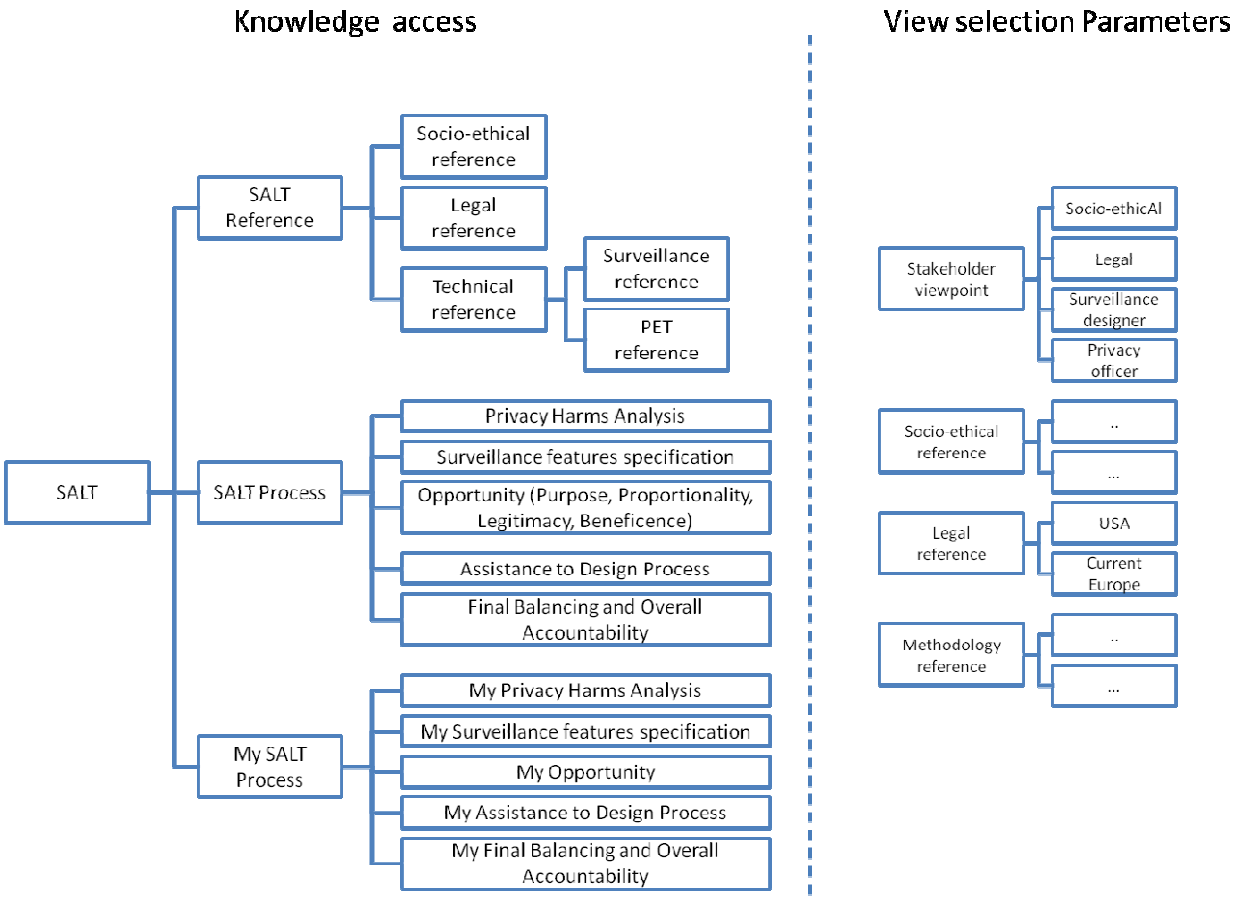


Figure 3. Example of SALT framework content

Consequently, the addition of relevant and comprehensive information to the SALT framework is a crucial task, because the privacy and accountability measures taken into account during the design phase of a surveillance system will rely on this information. And not only the

identification of the proper information is important, but also how it is represented and stored into the framework. We focus on developing an implementation that allows for a quick and efficient access to the data and a digital representation that properly contains the information given by the experts. Figure 3 shows an example of the SALT framework content and how can be seen by users.

The format of data storage is given by the nature of the information itself and the usage that is going to have. At this point we have to remark two main functionalities for the information contained within the SALT framework:

- Show privacy-related and accountability-related concerns to system designers at design time. This fact aims to achieve the privacy-by-design approach.
- Help to check that privacy and accountability requirements have been properly taken into account in a given surveillance system design. This fact aims to achieve the verification of the SALT compliance.

For the first functionality, the SALT framework provides, via the SFMT, text-based information that can be easily read by system designers. Thanks to this information system designers will be able to better make appropriate decisions for their designs to include privacy and accountability measures.

The second functionality is more difficult to achieve. In this case it is necessary to find some type of information representation that is objective enough, in a way that it allows for a later matching between the represented information and the system design, thus we can check whether privacy and accountability concerns have been properly taken into account or not. The format (or formal language) chosen to perform this representation is OCL (Object Constraint Language).

Therefore, the text-based information provided by experts will also be translated into what we call OCL rules. Due to the existence of these rules, the PARIS project envisages the development of another tool that helps to "automatize" (as much as possible) the process of checking the compliance, within the system design, of such OCL rules. We detail these aspects in Section 3.5.

3.2 SALT framework management tool

The SALT framework management tool (SFMT) is the element used to interact with the SALT repository. Experts in socio-contextual, ethical, legal and technological areas will use it in order to provide their knowledge to the framework, whereas surveillance system designers will use it to retrieve the information they need.

3.2.1 Overview of the SFMT

One of the main challenges managed by the project is to address the multi-domain concern of each stakeholder. To provide the same interface to all users sounds unreasonable since it will not take into consideration their respective needs. For this reason, the project decides to develop a set of tool. Figure 4 summarizes the tool set in order to manage a SALT Framework.

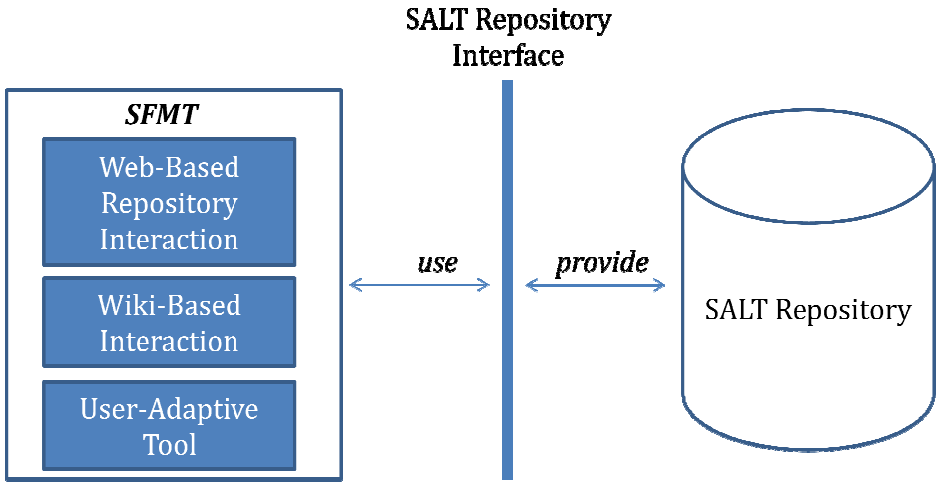


Figure 4. Overview of the tool set of the SMFT

In order to manage a SALT reference, the PARIS project considers the following tools:

- A web-based tool
- A wiki-based tool which converts the repository into a wiki. This wiki is easy to use and is more targeted to socio, ethical and legal experts.
- User-adaptive tools can be used by SALT experts or surveillance system developers. For instance, there is a tool which runs a questionnaire.

In the next section, the web-based tool and a user-adaptive tool are described. Deliverable D3.2 provides a more in depth description of the architecture and the development process.

3.2.2 Example of the Web-based tool

This tool shows an interface that can be used through any common web browser. Thanks to this web interface users can access and interact with the SALT repository without having to install any third party application. Figure 5 shows a screenshot of the search interface of the web based SFMT.

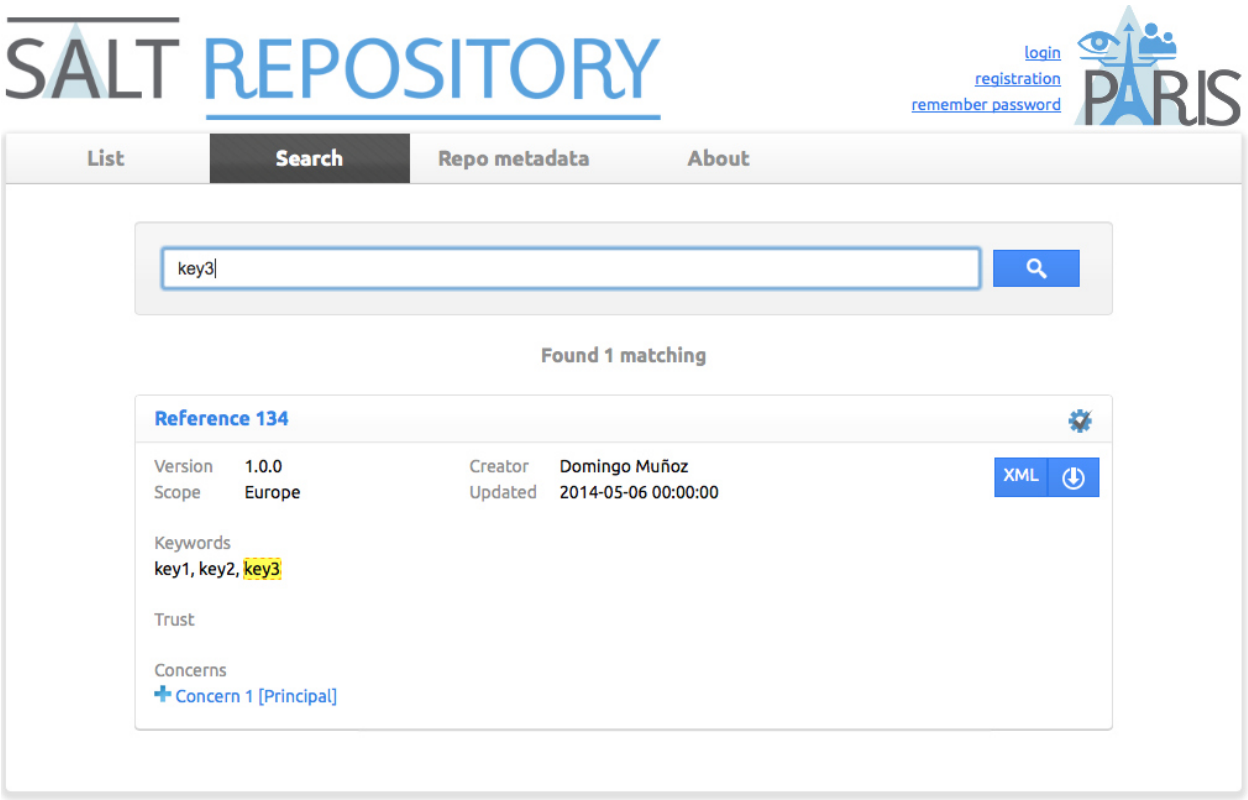


Figure 5. Search interface

As it can be seen, each SALT reference has a number of keywords, which can be used to help in the task of searching information specific to a restricted group of surveillance systems. The search functionality is proactive, i. e. the tool starts to look for the appropriate SALT references as soon as the user types in some words. Besides, keywords are not present just at the reference level, but they are also located at the concern level (each concern may contain its own set of corresponding keywords).

At this moment of the development, a user of the tool can see the text description of the privacy and accountability concerns within a SALT reference, as well as other relevant information such as, version number, scope, creator, etc. Besides, there is also the possibility of downloading the whole reference and watch it in XML format (this last representation is more technical oriented, since surveillance system designers or expert knowledge providers do not necessarily have to know about XML. For this reason, it is not sure that this functionality will be kept for future versions of the tool).

Each concern may also be described by means of OCL rules (one or several). If OCL rules are present, we say the SALT reference is a complete reference, otherwise we call it incomplete reference. Since OCL rules have to be provided by OCL experts, not all SALT references will have

them available, that is why this information is not mandatory for the creation of a SALT reference.

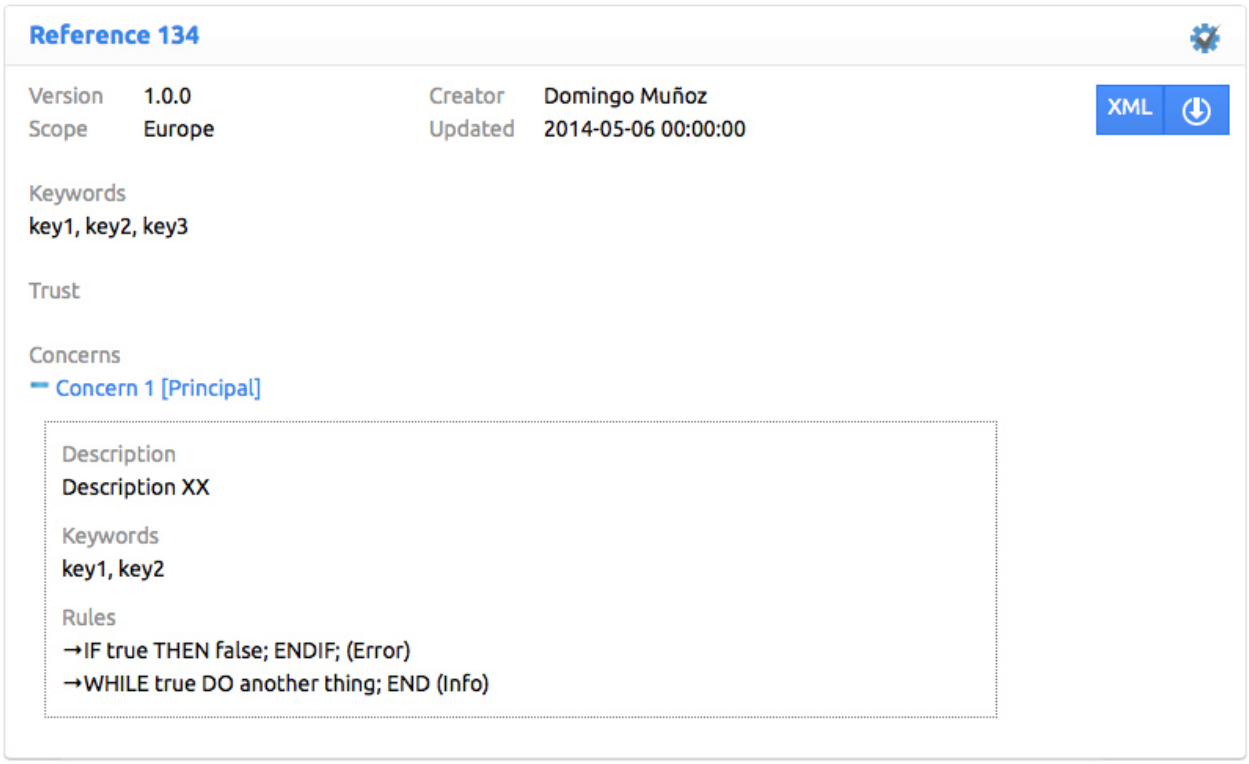


Figure 6. SALT reference with concern description

Figure 6 shows a complete SALT reference, which is indicated by the icon that appears at top right corner. As it can be seen, in this case not only the description and the search keywords of the concern are provided, but also the OCL rules, which are accompanied by a corresponding level that indicates the importance of not fulfilling the rule: error, warning or info.

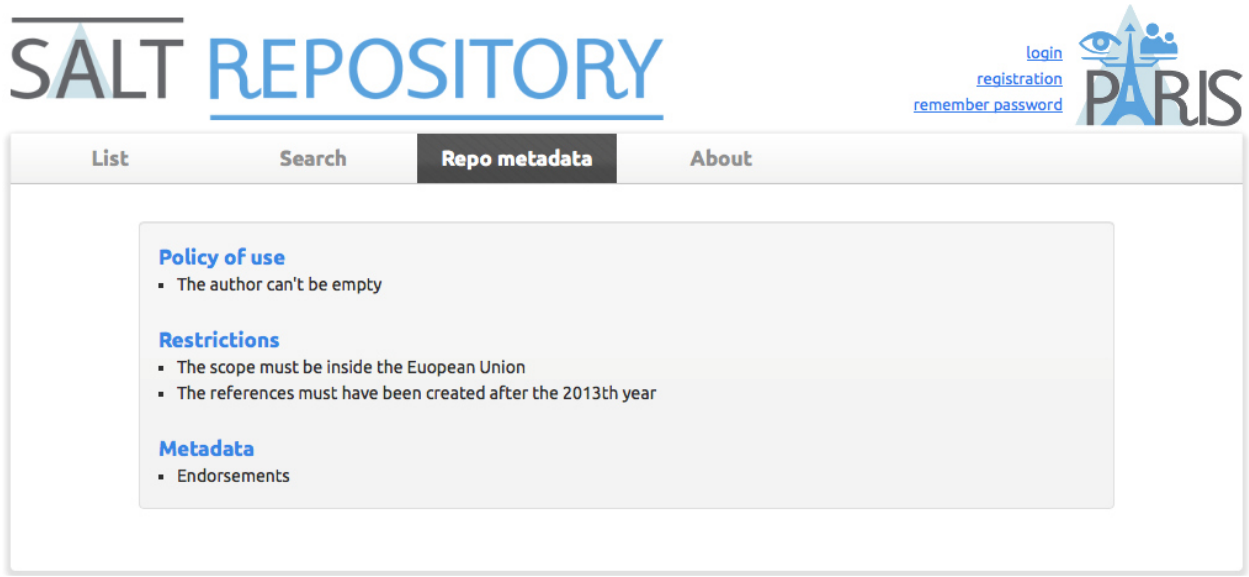


Figure 7. Repository metadata

Apart from this functionality, the tool also provides the option of listing all SALT references already stored in the repository, although this possibility will have to be reanalyzed in future versions when the repository is populated with many more references.

There is also a space where any type of information regarding each particular repository can be added. This information is not inherent to any SALT reference in particular, but to the repository itself. Figure 7 shows an example.

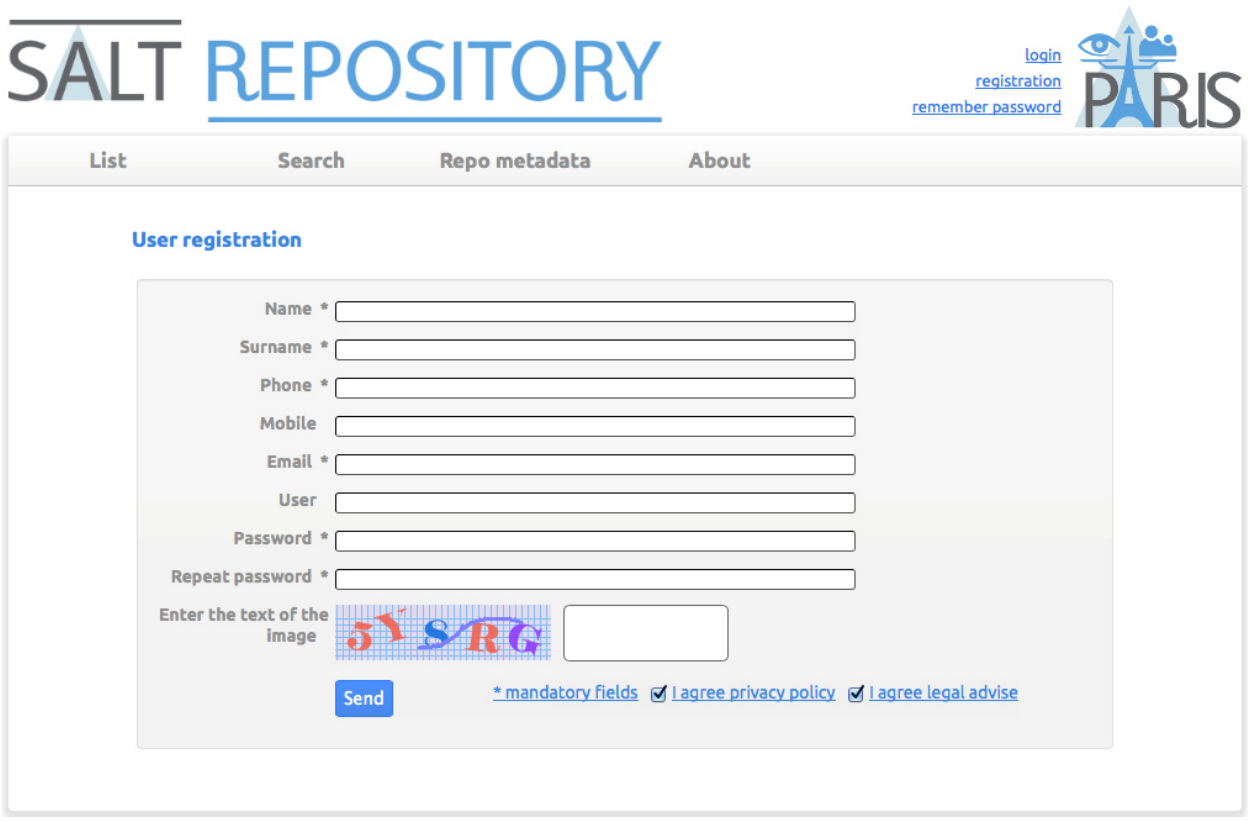


Figure 8. SALT Repository registration form

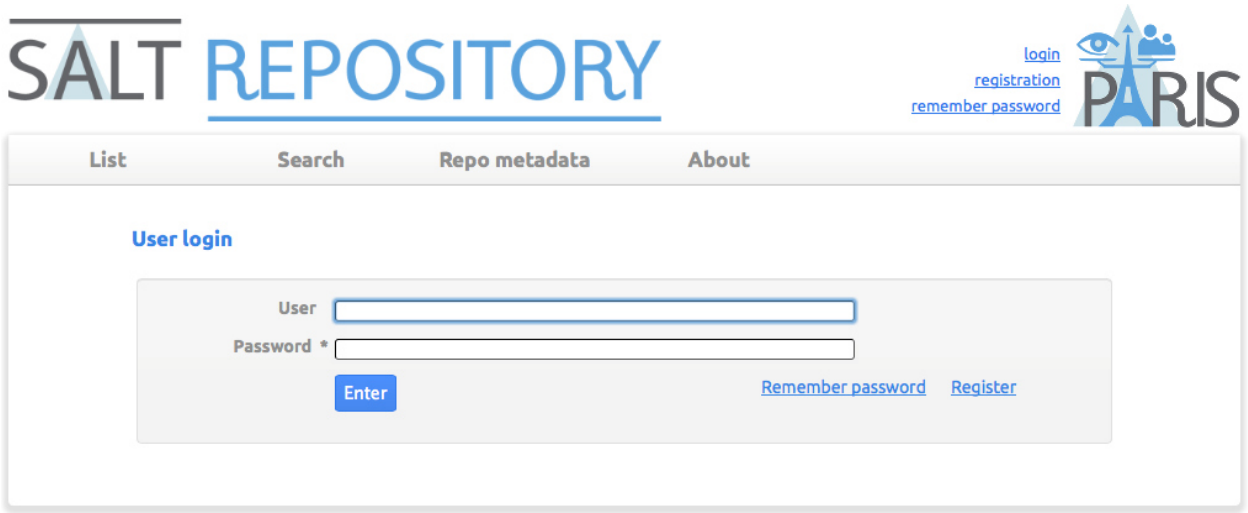


Figure 9. SALT Repository login interface

Another important feature is the possibility of adding new SALT references. However, this option does not appear unless the user is registered and logged into the system. Figure 8 and Figure 9 show the registration form and the login interface, respectively.

Once the user is logged into the system, he is allowed to create a new SALT reference and add as many concerns as necessary within it. Figure 10 and Figure 11 illustrate how the SFMT handle with this task.

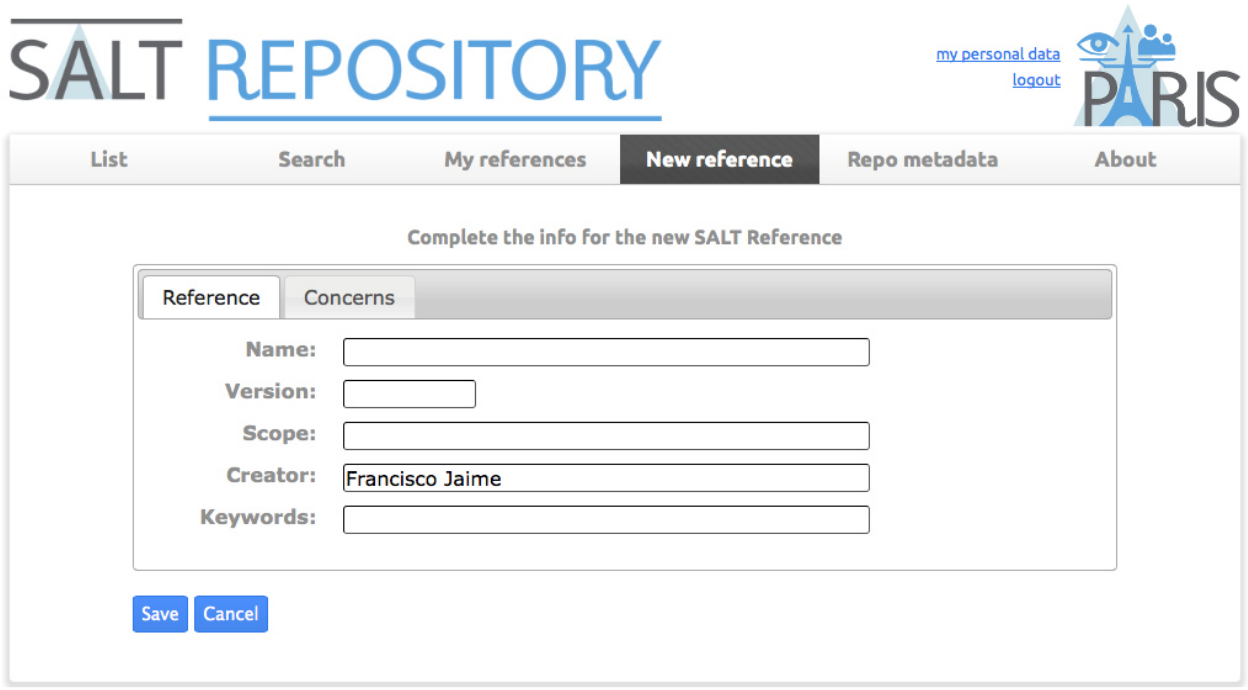


Figure 10. Creation of a new SALT Reference

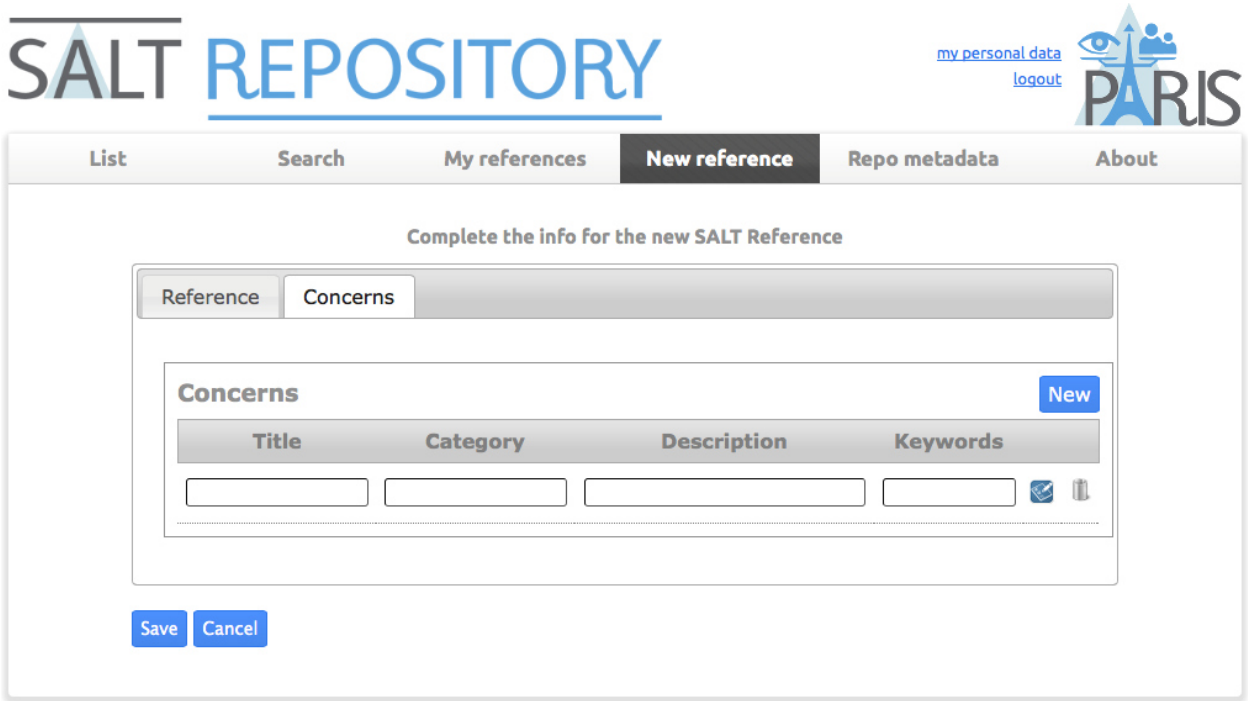


Figure 11. Addition of concerns to a SALT Reference

Finally, a last option provides brief and general information of the tool, together with the partners involved in the PARIS project. See Figure 12.

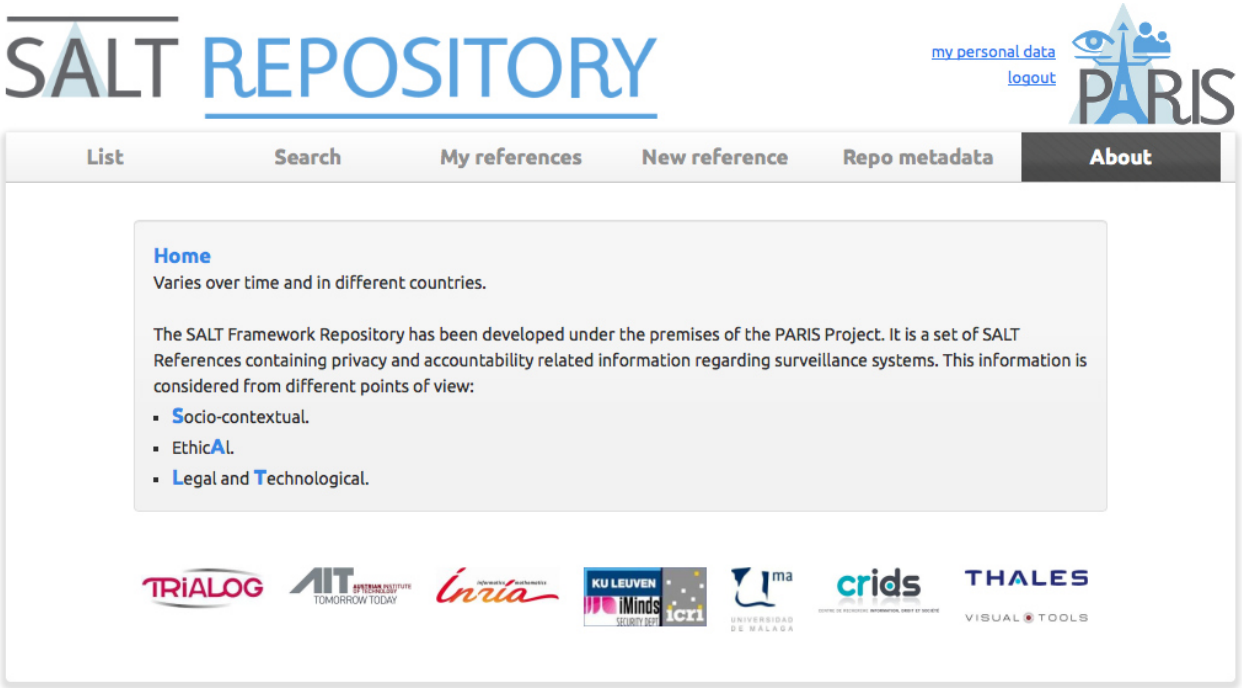


Figure 12. SALT Repository general information

3.2.3 Example of a Questionnaire-based Tool for Surveillance System Developers

The SALT Framework will contain some questionnaires among all content. The surveillance system will have to run them in order to check if he addresses properly privacy concern.

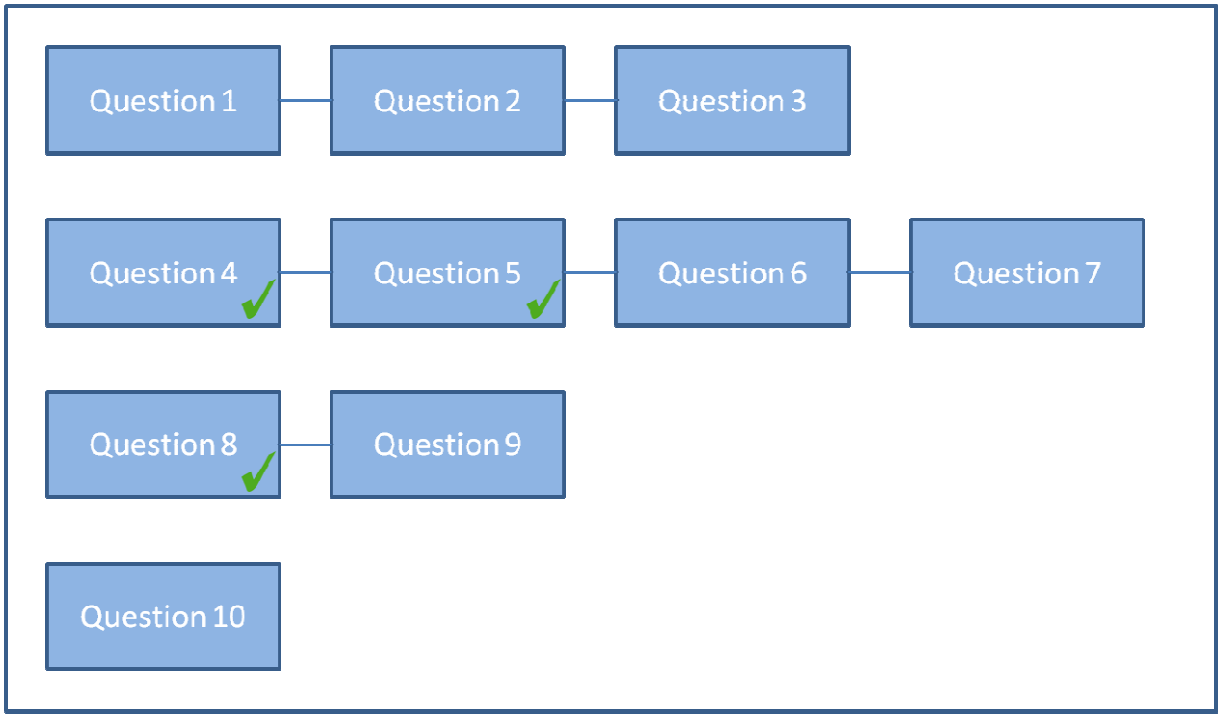


Figure 13. Example of a the execution of Questionnaire

A questionnaire is composed of a set of questions. Of course, the system developer can run the questionnaire at the beginning of the project and resume later. Moreover, the tool has to be flexible since all questions are not sequential. An example is given in Figure 13.

The tool will also be able to provide details of the questions, to show the terminology used by the questionnaire, and to store the answers of system engineer.

3.3 SALT general process

The SALT general process covers a wide range of tasks, from the addition of information to the SALT framework (the experts' knowledge) to the check of the SALT concerns (privacy and accountability aspects that appear in SALT references have been taken into account) of a given surveillance system design. The following are the main tasks concerned to the SALT general process:

- Add/update information to the SALT framework (via the SFMT).
  - Textual description of the experts' knowledge.
  - Textual description indicating how to implement (take into account) the above information in a system design.
  - OCL rules, which provide a formal representation (without ambiguities) of the above information. These rules are not mandatory to be included into a SALT reference, since they need an OCL expert to formulate them, who will not always be available.
- Retrieve SALT references from the SALT framework.
  - Each reference will contain a series of concerns with the privacy and/or accountability related information.
- Create a surveillance system design.
  - Taking into account the information from the SALT references (privacy-by-design and accountability-by-design).
  - Using an UML profile, which contains a set of modelling artefacts for representing the elements that can appear in a surveillance system.
- Verify that SALT concerns have been properly implemented in the system design.
  - With an automated process that checks the OCL rules against the system design (in case the SALT references used are complete, i. e. OCL rules are provided).
  - By a human user (in case the SALT references used are incomplete, i. e. OCL rules are not available).

Here we are going to focus on the last two points, since the SALT framework has already been introduced in section 3.1 and better detailed in the PARIS project deliverable D2.2 (Structure and Dynamics of SALT Framework).

The materialization of a surveillance system design is an UML model, since this type of models is very flexible and allows for representing nearly anything that we need. However, surveillance system designers may not know about UML (and surely they will not), and it is unviable to make them all learn how to work with UML. For this reason, the PARIS project has developed an UML profile. This UML profile is a set of modeling artifacts covering all elements that can be part of a surveillance system, together with the main characteristics of each element. Thanks to this

profile, system designers will just need to drag (from a list) and drop the elements they need for their systems designs and fill in the values for their respective characteristics, without knowing about the UML description that lies underneath. Figure 14 shows an example with some modeling artifacts from the UML profile that represent elements from a surveillance system (this still is a work in progress, we expect future versions to have a more friendly interface, showing nice icons for each represented element instead of UML alike boxes).

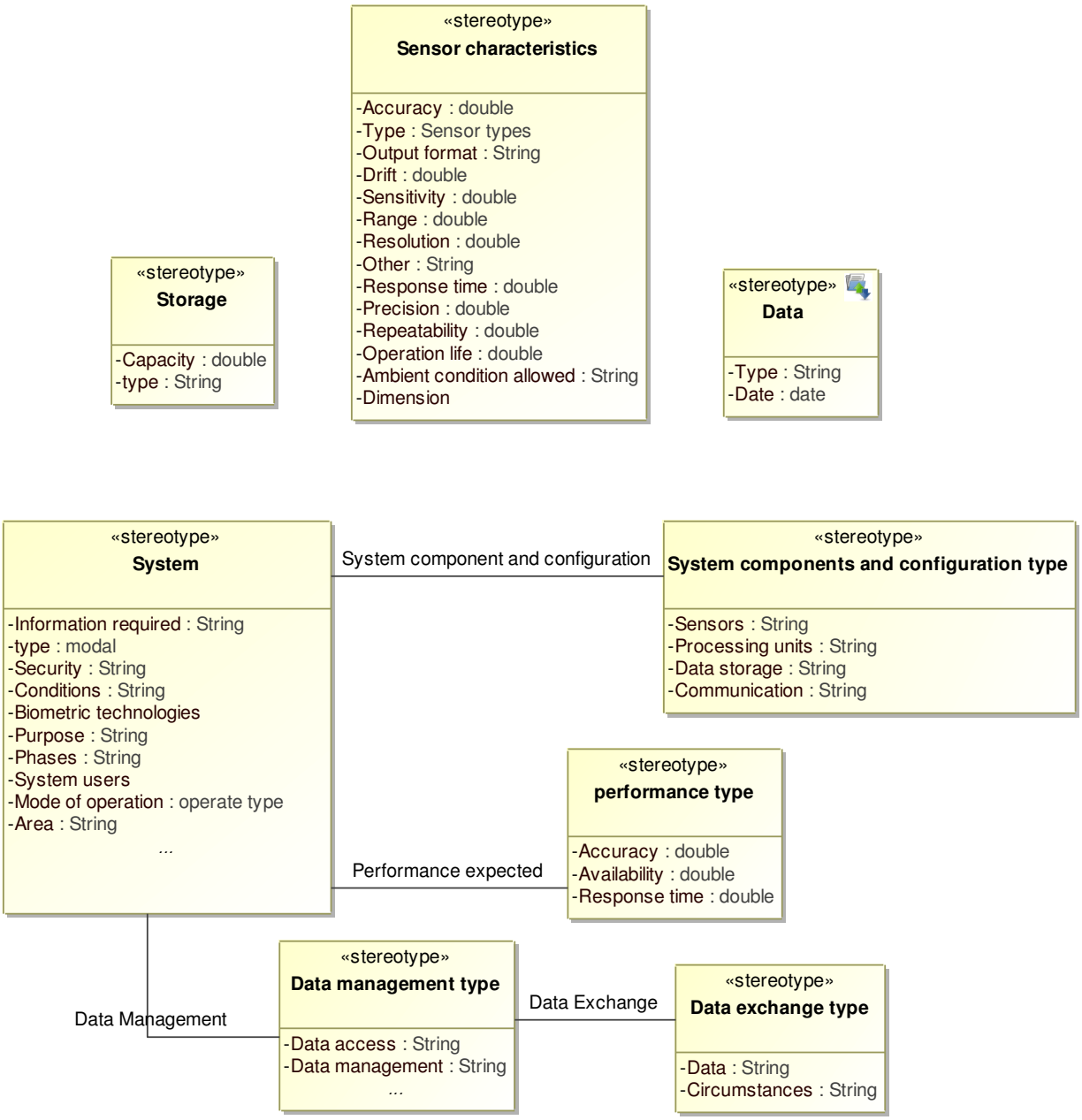


Figure 14. Example with UML profile elements

The UML model that corresponds to the surveillance system design is created by the system designer taking into account the information retrieved from the SALT framework. This information tells the designer about the privacy and accountability concerns that may be taken into account for a particular surveillance system, but it also shows a way about how these concerns could be implemented in the design. Here we would like to remark the expression "a way": each concern may be implemented in many several ways, but the SALT reference shows a

possibility. In fact, the SALT framework can have different SALT references expressing different possible ways of implementing the same concern.

At this point, once the system design is available, the last step of the SALT general process regards to checking the implementation of the privacy and accountability concerns, i. e. whether the system designer properly applied these concerns to the system design or not. This task can be performed in two ways:

- Automatically: the SALT references are complete, thus they include OCL rules that formally represent the privacy and accountability concerns. An automated tool can check the correspondence between these rules and the UML model that represents the system design. Initially, this tool is called PAERIS.
- By a human user: the SALT references are incomplete, which means they lack of the OCL rules. In this case the PAERIS tool cannot be used, but if the SALT process has been followed, it can provide valuable information to assist a human user to check the privacy and accountability aspects of the system design.

The translation of privacy and accountability concerns into OCL rules is a difficult task. Let us not forget that these concerns may correspond to the categories socio-contextual, ethical, legal and/or technological. This means that sometimes the concerns will be ambiguous, vague, too wide or general. Because of this the creation of OCL rules is a difficult task and sometimes it is not even possible. This is the reason why the SALT framework may also contain incomplete SALT references.

Finally, another point to take into account relates to the surveillance system operation. Usually, a considerable amount of privacy and accountability concerns arise once the surveillance system has been deployed, i. e. during its operation time. However, these concerns could be avoided, or at least reduced, with a consistent system design that takes into account privacy and accountability aspects from the beginning and provide solutions at design time. Once again, this is what we call privacy-by-design and accountability-by-design, the goals pursued by the PARIS project.

3.4 Using SALT for surveillance based on video search

This section mainly addresses the use of the SALT approach applied to video archive search. Video archive search can be considered as a capability subset of the more generic “video surveillance systems” domain. An overview of the of the SALT approach applied to the whole video-surveillance domain is proposed within Section 3.4.4.

3.4.1 Video archive search

Surveillance video is used either in *real-time* or for searching events of interest *afterwards*. Video-surveillance systems often combine the two approaches (real-time use of the system, and off-line use of the system often referred to as “forensics” usage). Video-archive search is a specific category within forensics methods which is based on the use of Video Contents Analytics (VCA) applied to recorded streams.

Within powerful state-of the art video archive search modules, the video-surveillance footages are usually indexed along multiple dimensions. As a function of video surveillance, video

archive search is usually integrated as a part of the surveillance system. Using IBM Smart Surveillance Suite (S3) as an example, the red box in Figure 15 indicates the location of the video archive search in the whole surveillance system architecture.

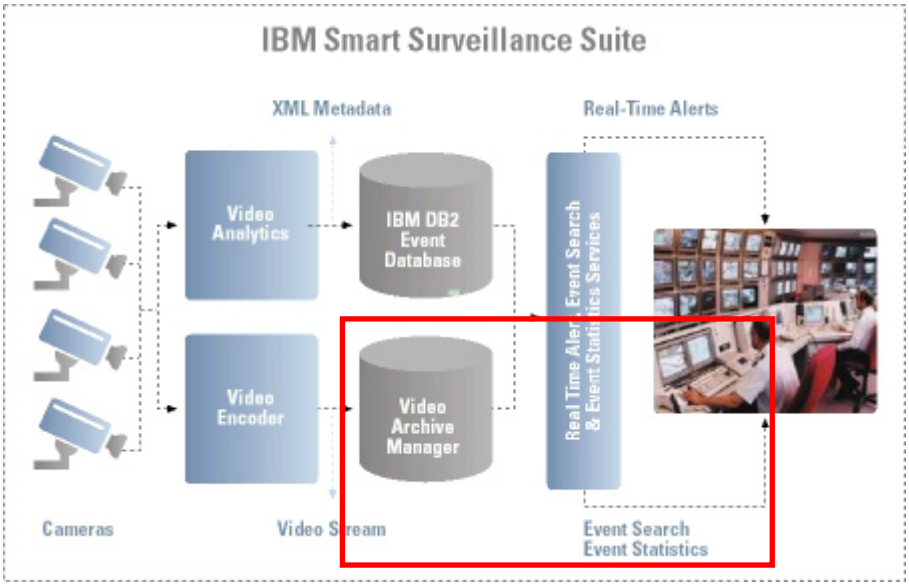
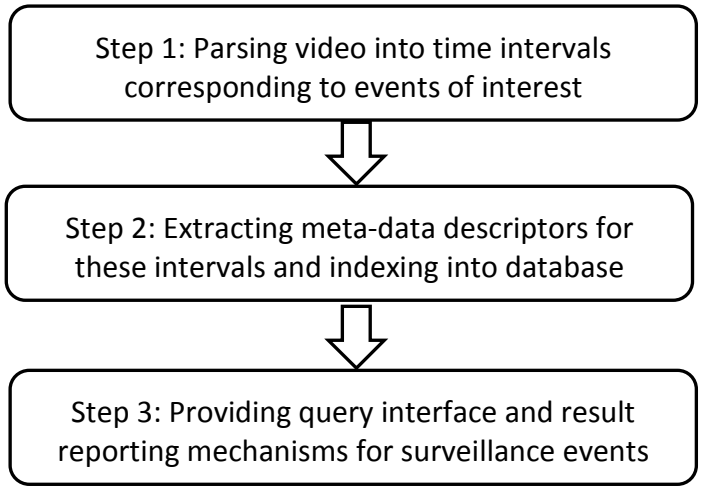


Figure 15. Example of placement of video archive search in IBM S3

Video archive search typically focuses on two key search criteria [5]:

- Specific search for people and objectives (events, patterns).
- Generic search for objectives (events, patterns) and events of interest.

Software architecture for video archive search usually has the following steps:



In general, key considerations in the design and development of video archive search includes:

- Search goals in the context of surveillance, for example, motion detection, intrusion detection, virtual line crossing detection, abandoned luggage detection, person counting, license plate recognition or face recognition, as well as system and performance requirements.

- Computer vision algorithms and video analytics to perform searches according to the surveillance goals.
- Video archive search software architecture and system integration. Software architecture for video archive search and how to integrate search functions into system operation and other system components. Figure 16 shows an example of the video search software architecture of IBM S3.
- Video archive search operator HMI. This component enables the full interaction with the user of the system, whoever the user is. It implements controls and commands which provide possibility to select videos of interest, algorithms of interest, and to display the archive search results in the best possible user-friendly way.

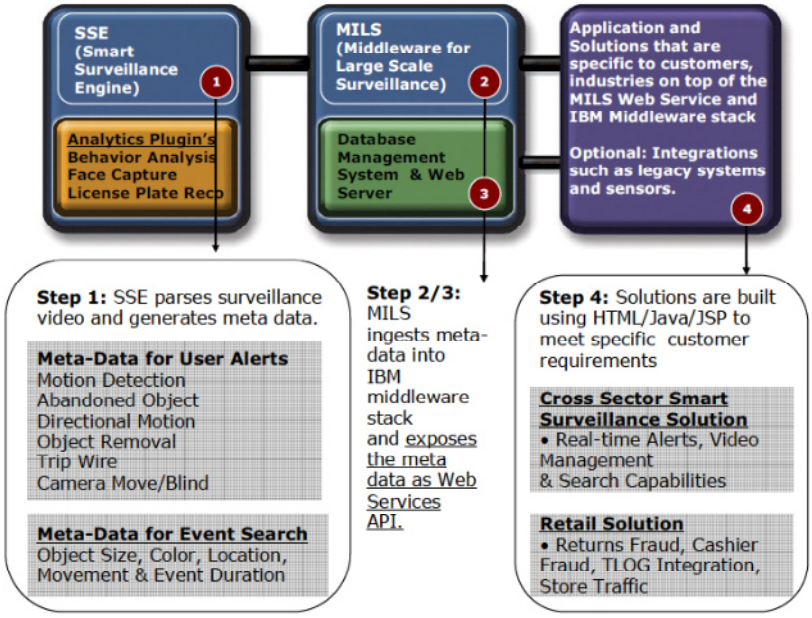


Figure 16. Software architecture of video search for the IBM S3 [source: 1]

A common generic architecture for the core of the archive search, namely the algorithms processing, is proposed below:

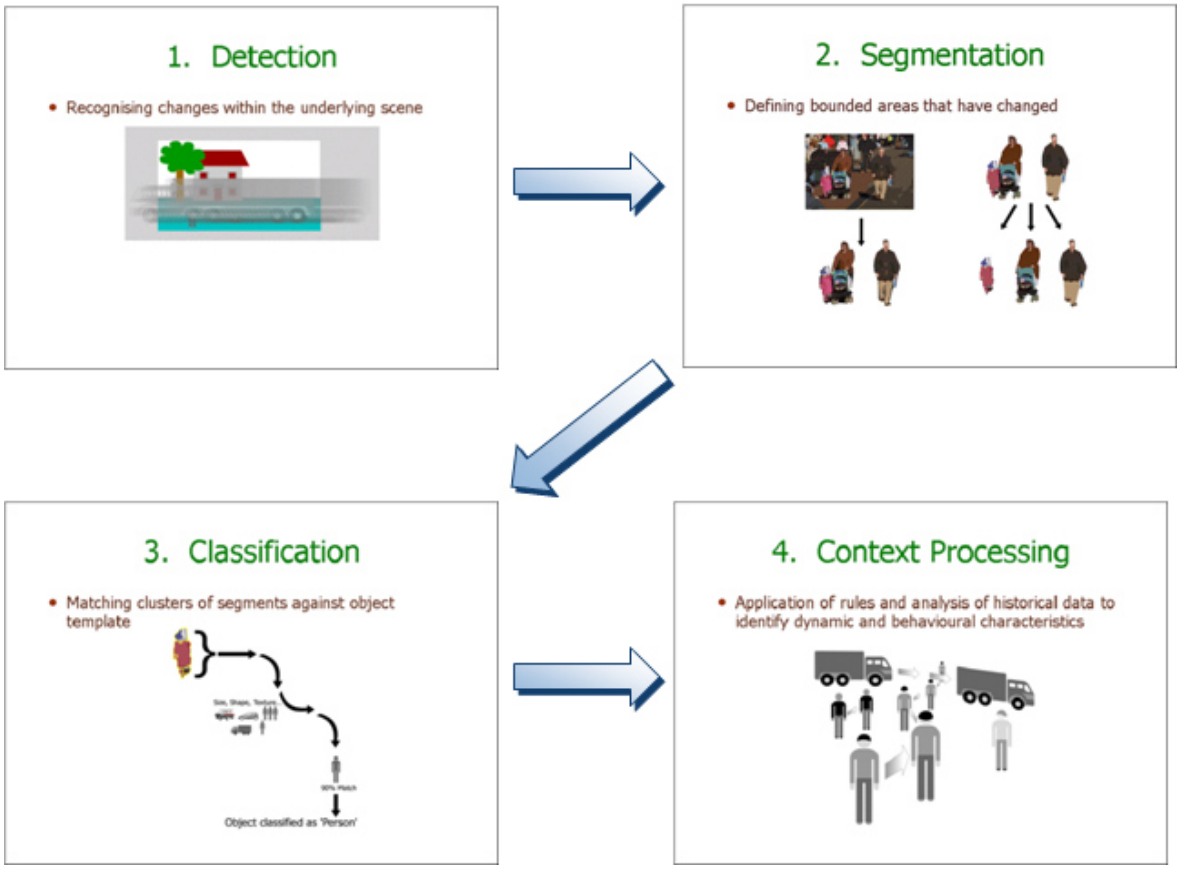


Figure 17. Generic decomposition of VCA algorithms within steps

3.4.2 Engineering process

The design and development of video archive search is basically a software engineering process. Typical software engineering process can be represented by the V-model (see Figure 18).

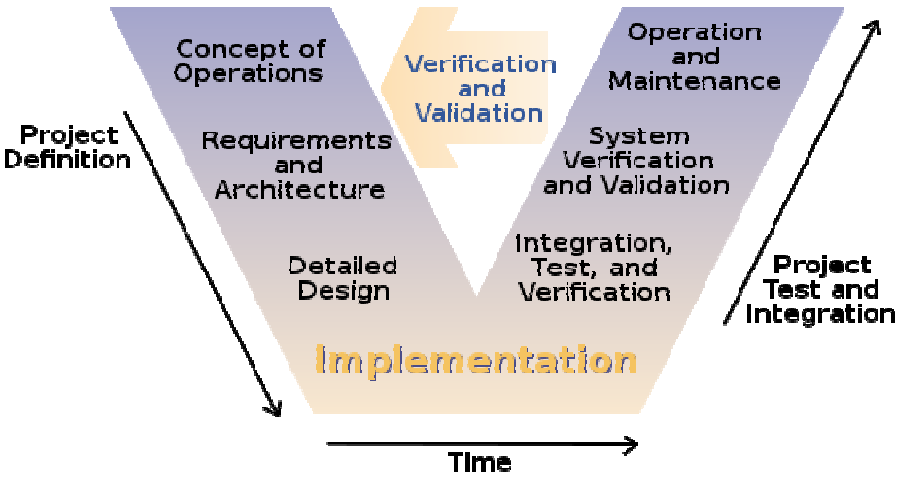


Figure 18. Software engineering V model [source: Wikipedia]

Most design decisions are made in *detailed design* within the project definition phase, in relation with the requirements and architecture step, that should embed as framework of requirements most of the functional prescriptions issued from the SALT framework.

3.4.3 Integrating SALT framework

We envision that the SALT framework is integrated into the requirement engineering and design of video archive search. Figure 19 illustrates how the integration can be realized.

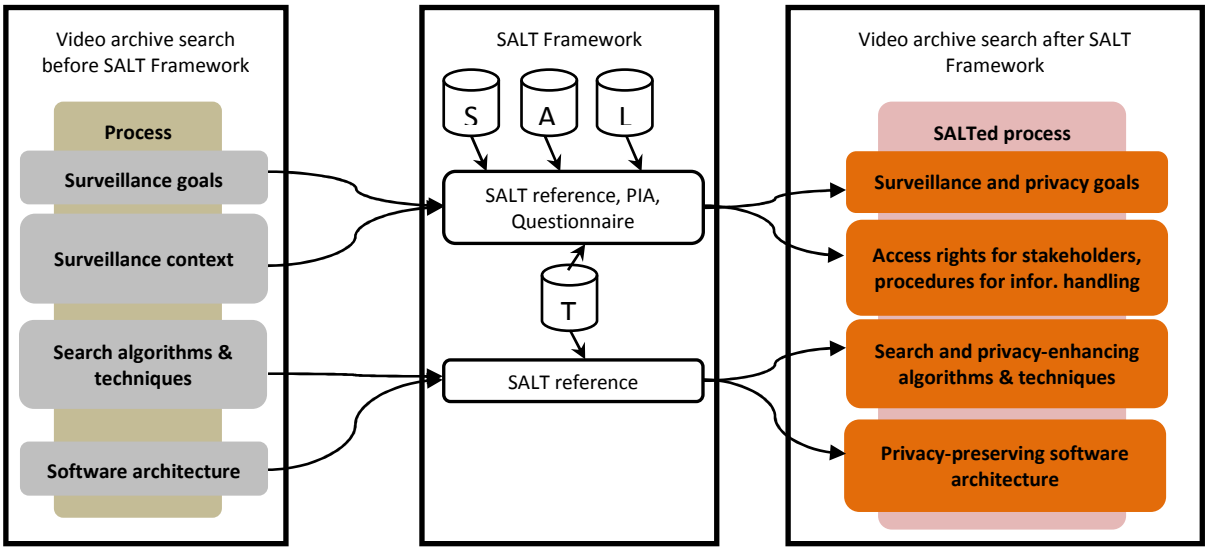


Figure 19. Integrating SALT framework for video archive search

Initial design artefacts (video archive search before SALT framework) will go through SALT framework to produce privacy-compliant design artefacts (video archive search after SALT framework). Details related to these specific artefacts are given below:

- The surveillance goals and the surveillance context are leading to a subset of functional requirements and to requirements about its use (who, how, which limits). The surveillance goals will be scrutinized by consulting SALT references and privacy assessment such as PIA and questionnaires, to ensure that all surveillance goals are balanced against specific social, ethical, and legal requirements. The result is a combination of requirements (functional and non-functional) including the scrutinized surveillance goals and additional privacy requirements.
- The search algorithms and techniques will go through the SALT framework by consulting SALT references to ensure that they support both the functional requirements and privacy goals. Additional privacy-enhanced technologies specified in SALT references can be added.
- The software architecture will be scrutinized and additional software components can be added to produce privacy-preserving software architecture.
- The process can be seen as the mechanisms to guide and glue all activities related to the design that leads to the software architecture. Thus the original process will be modified to accommodate the changes due to the addition of privacy into the design process.

Note that we will validate this approach in WP5 and refine it when necessary. Moreover, to ensure accountability, the integration with the SALT framework addresses how the compliance of the search system with privacy requirements can be demonstrated. The framework provides pointers relevant to video search systems, both legal and technical. Legal aspects can be more or less domain-specific depending on national regulation. Technical accountability considerations focus on techniques to promote verifiability of compliance. In practice, archiving specific video images may be prohibited, depending on the circumstances and national law. The demonstration of legal compliance can feature a side-by-side analysis of data capture and the relevant legal framework. The level of detail at which the demonstration of technical compliance can be done depends on the capabilities of the video search software. If detailed logs are available, it may be possible to check them against policies specifying which categories of data can be archived and searched. Another aspect of accountability features clear communication towards the individuals subject to the system, providing them with precise information about which images are recorded and how they are processed, as well as a means to access recordings of images of themselves.

3.4.4 Overview of the SALT approach to a global video-surveillance system

A typical video-surveillance system embeds software components, such as video archive search, video management system (VMS) software, Network Video-software, but also hardware components, such as cameras, servers, network components. This implies that the design of a video-surveillance system includes a wider range of design choices compared to a stand-alone software module.

A typical and demonstrative example is given by the position of the cameras within the space under surveillance. The choice of the 3D locations and angular attitudes of the cameras is part of the system engineering choices. This choice typically impacts both the surveillance performance of the system and the privacy performance of the system (cf. e.g. 2010 EDPS video-surveillance guidelines which insist on this point especially for its impact on the privacy axis). This kind of choice has an impact on accountability. While it may not be relevant for data subjects to know about specifics such as camera positions, these choices have consequences on the richness of recorded information and on what can be inferred from it. Accountability of policy should focus on functional configurations rather than on the physical settings generating them. Similar considerations are true for parameters such as image resolution and pan-tilt-zoom capabilities.

More and more tools are available on the market or on the web to deal with this design issue. They provide accurate and efficient means to image the impact of cameras FOV (Field of views) on the monitored space. Figure 20 and Figure 21 show and example of such a tool.

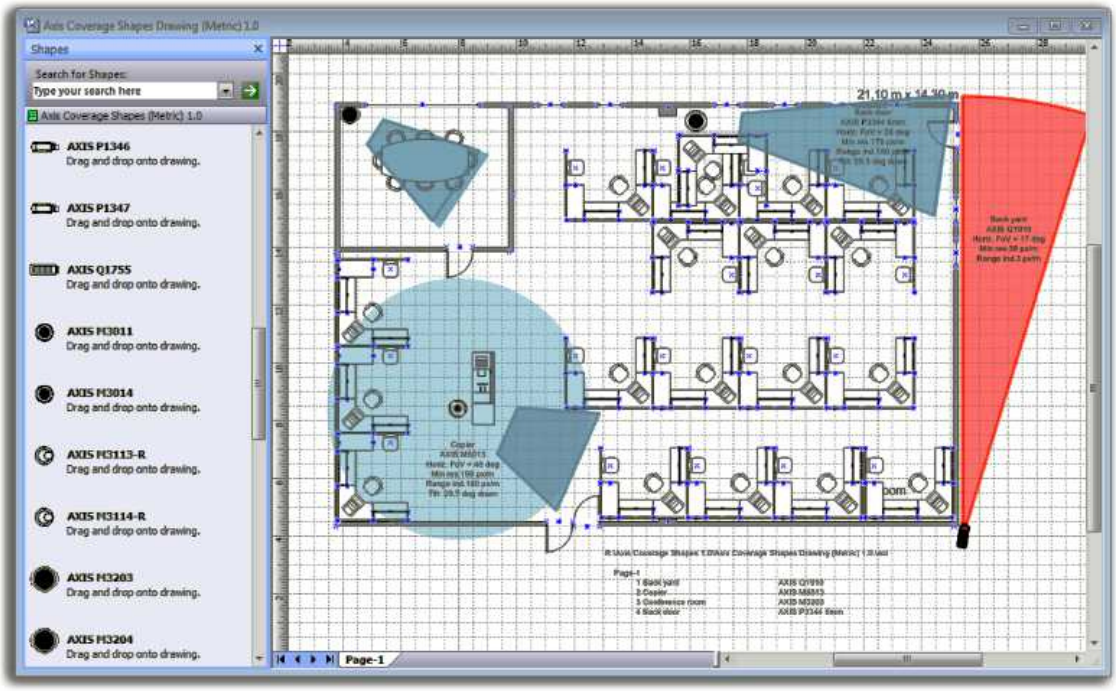


Figure 20. Example 2D tool for cameras field of view evaluation [from www.3dvisworld.com, designing video-surveillance systems by simulation]

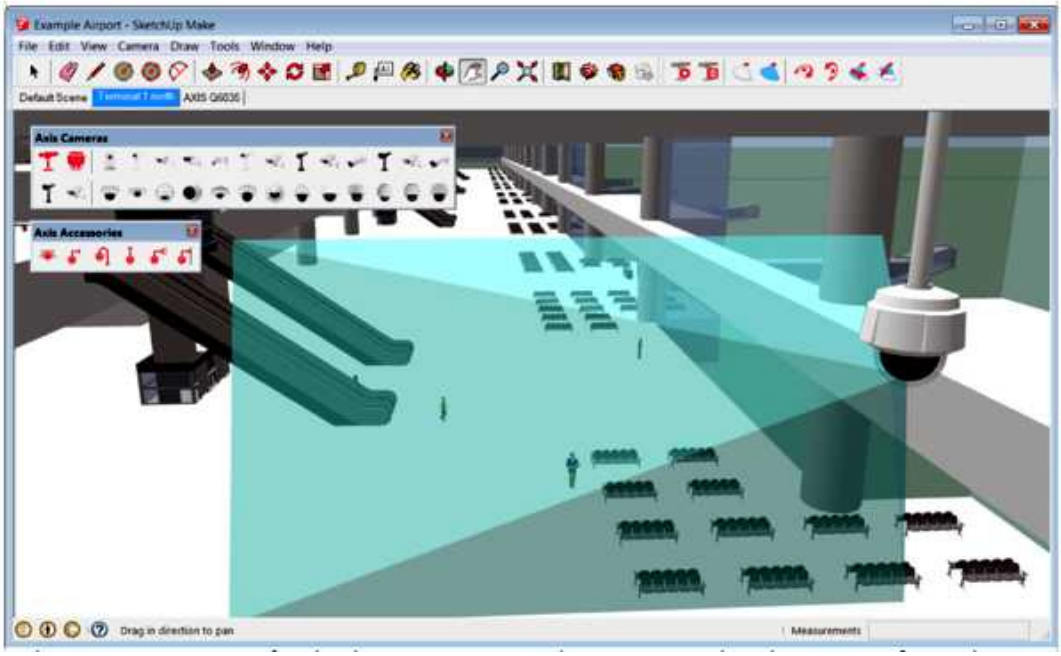


Figure 21. Example 3D tool for cameras field of view evaluation [from www.3dvisworld.com, designing video-surveillance systems by simulation]

More generally speaking, *system engineering* differs from *software engineering*, as it has to deal in addition to software engineering with all the additional constraints and phenomena linked to physical space. As an example, the following properties and phenomena are specifically addressed by system engineering:

- Position,
- Orientation,

- Speed, acceleration,
- Mass, weight,
- Color,
- Electrical power,
- Radioelectric emissions,
- Optical properties,
- Chemical reactions,
- Electromagnetic compatibility,
- …

As a result, system engineering uses classically three different architectures to provide a full system design:

- Functional architecture,
- Logical architecture,
- Hardware architecture (see Figure 22).

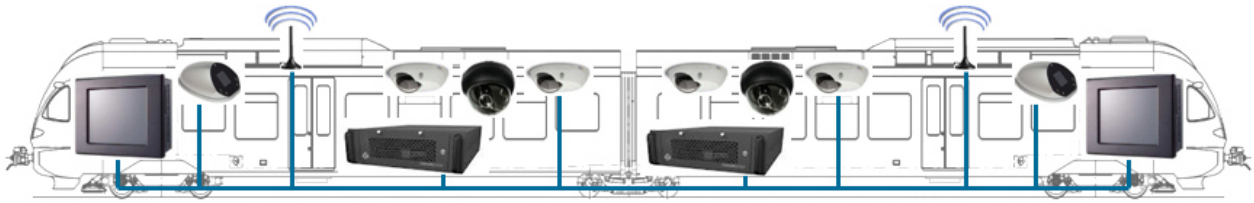


Figure 22. Example hardware architecture of an onboard video-surveillance system

These differences between system and software engineering are huge. Nevertheless, the design approach presented in Section 3.4.3 using the SALT framework remains applicable to a full video-surveillance system: a SALT reference embeds in this case wider information, but the process to obtain a SALTed design process remains the same. The application of the SALT process to video-archive search is for this reason demonstrative and generalizable.

3.5 Using SALT for surveillance based on biometrics

As described in the previous sections, the SALT Framework provides guidelines for the implementation of Privacy by Design and Accountability by Design, and a tool that can be used, in some cases, to verify if a design addresses the SALT concerns (SFMT). These features can be harnessed by the designers of biometric systems and by the service providers to build systems that balance the recognition capabilities with privacy protection and integrating the concept of accountability.

Taking all the capabilities of the SALT Framework into account, during the lifecycle of a biometric system, it can be used for three main tasks, as explained in deliverable D6.1:

- **Extraction of concerns and recommendations** on privacy and accountability for the design of a system according to the SALT principles. These guidelines can be used to complete the initial specification of the system at the "Requirements" phase.

Accountability requirements directly arise from privacy concerns: every privacy enhancing aspect must be demonstrated both declaratively, legally and technically. The SALT Framework facilitates this aspect of verifiability by suggesting how evidence of privacy preservation can look, e.g. by providing relevant legal texts that justify the legality of the system, or by mentioning methodologies to check compliance on the technical level, for instance log analysis.

- **Validation of the design of the system according to the SALT principles**, which can be performed at the design phase but also anytime the system is modified, for instance after the system testing or for maintenance purposes.
- **Consulting references related to a specific system for auditing purposes**, which is normally performed by the Data Protection Officer in order to verify that the system complies with the current regulations on privacy and data protection. The core idea is that relevant evidence is both available and easy to verify

Figure 23 depicts the role of the SALT Framework at the different stages of the lifecycle of a biometric system.

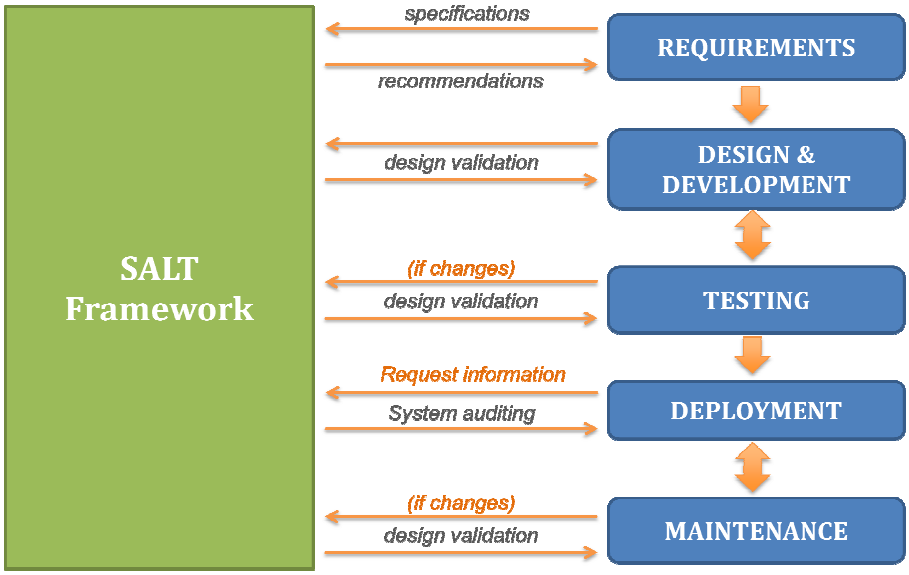


Figure 23: Role of the SALT Framework at the different stages of the system lifecycle

The first stage in the design of a biometric system is the collection of requirements from customers and services providers. This initial specification covers some functional, environmental, operational and technical requirements for the system, as well as business constraints or legal requirements specifically requested by service providers. Using the SALT framework it is possible to add other requirements to this list that should be fulfilled by the system to take into consideration privacy and accountability from the start. In particular, the legislation may allow the recording or use of biometric data only when specific conditions are fulfilled. Rules may vary according to which category of biometric data is captured. One role of the SALT Framework is to guide designers by pointing them to the regulations that apply in their specific setting. The setting is input to the framework through questionnaires.

The difference between using and not using the SALT Framework is especially crucial at the *Requirements* stage. The guidelines provided by the SALT Framework at this first stage allows to

create a more solid foundation for the design of the system in terms of privacy and accountability, which will raise the level of privacy and protection and will facilitate the future system audits by Data Protection Officers, Data Protection Agencies, and law enforcement authorities in case it is required. In that sense, the new Data Protection Package contains new obligations in terms of accountability, so in many cases taking it into account is not a bonus but a duty. Otherwise, the designers may not have in mind all the privacy and accountability concerns while designing and implementing the system. The consequences of this range from increasing the impact of the system on individuals' privacy, or having to deal with emerging problems related to the incorrect treatment of personal data with more difficulties and at a higher cost, to diminishing the trust of customers.

Before the implementation, the design of the system should be reviewed in order to check if it addresses the concerns provided by the SALT framework about privacy and accountability. The SFMT provides a mechanism for the validation of designs that can be applied to those systems that obtained complete SALT references in the first stage. Other systems cannot be verified automatically with this tool of the SALT Framework, which does not mean they are not SALT compliant, they just have to be validated in a different way. Thus, the SFMT facilitates the validation of the system and speeds up the process of design.

During the testing phase, some modifications may be required to improve the performance of the system. Anytime a change is made in the system, the resulting design should be validated again to check if it addresses the concerns provided by the SALT Framework.

Once the system is tested and verified as compliant with the SALT concerns, it shall be properly deployed and configured to work under the conditions defined in the initial specification, after which the system is ready to be used for the purpose it was built.

Biometric systems are subject to audit at any moment by Data Protection Officers or Data Protection Agencies to check if they comply with the current regulations on privacy and data protection. The systems addressing the SALT concerns should already meet the legal requirements, and hence they should get a positive evaluation report. In case the auditor disagrees with the implementation, he/she can consult the SALT Framework references used in the design of the system in order to check the basis of the different design decisions. As it is not possible to determine if the recommendations provided by the SF have been correctly implemented, a system where its design has been verified as SALT compliant, may not fulfill all the SALT principles properly once implemented. The SF helps the designers during the design stage, but the responsibility for the implementation is up to them. Anyway, it is an advantage for the designers to have in mind all the concerns regarding privacy and accountability from the beginning of the design process.

The last phase is the maintenance of the system, which may also require to perform changes in the system, that shall also be validated.

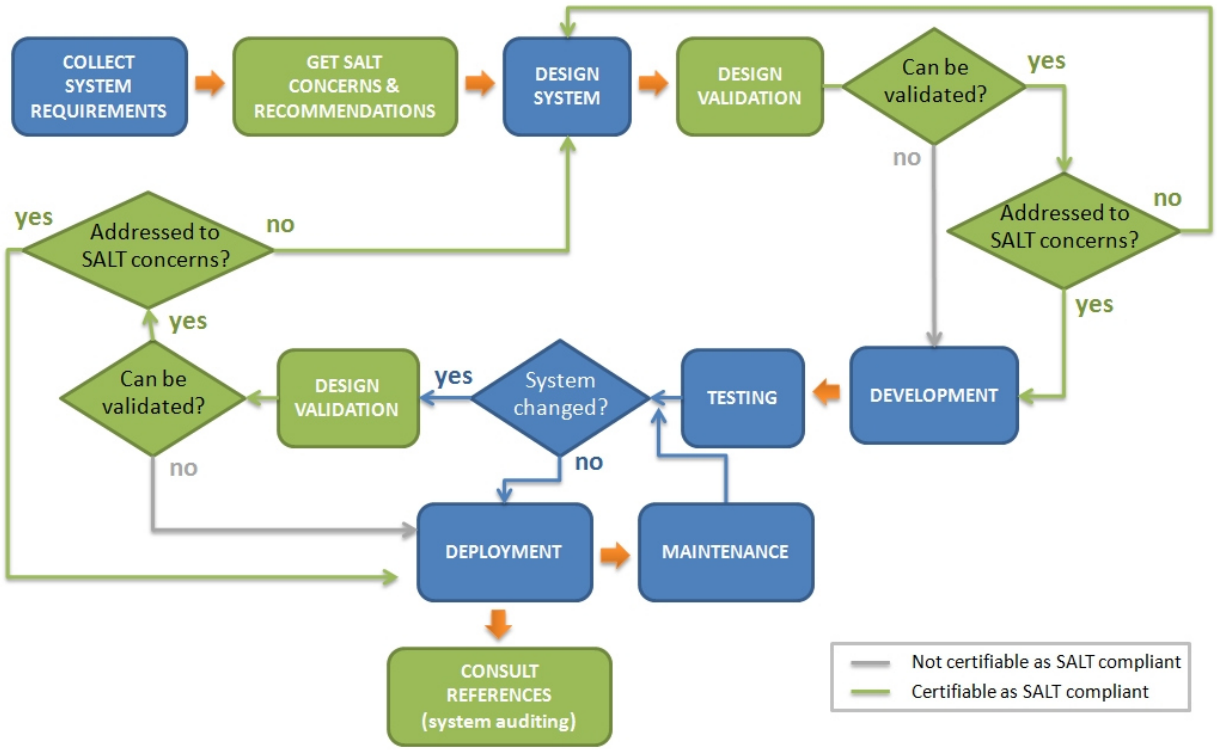


Figure 24: Design process of a SALT compliant biometric system

Figure 24 describes the design process of a SALT compliant biometric system. If the design can be validated through the OCL rules, it is possible to certify with the SALT Framework if the system is SALT compliant or not. Otherwise, if the system cannot be validated automatically with the corresponding tool of the SALT Framework, it may comply with the SALT principles, but that cannot be granted by the SALT Framework.

4 Specialisation of the SALT general process

This section discusses possible specializations of the SALT general process to the two types of surveillance technologies considered: video search and biometrics. Here we see whether some aspects of the process (or the information itself) can be specifically adapted depending on the type of the surveillance system under design.

4.1 Specialisation to video search technology

4.1.1 Interactive forensic search in large video data

Technological approaches in the topic of video surveillance often follow the principle “surveillance now and every time”, this has to be investigated seriously. Data protection and privacy rules have to be fulfilled. More data does not automatically mean more security or more information. A wide-area deployment of more and more surveillance cameras shows this dilemma dramatically, because a specific search for cars or even persons or special situations gets more complicated. The task is even more complex. (e.g. the new terminal in the Vienna Airport contains 1800 cameras, the ÖBB - Austrian Railway Organization operates 3000 cameras, 700 new cameras for the new terminal). As long as the exploiting system is not more intelligent than the users it will be left to the operator finally to conclude findings and draw conclusions from the mass data to find the relevant scenes. Tremendous advances in the fields of algorithm, hardware and software in the last years had given hope for a complete automatic and intelligent surveillance system. But requirements and reality differ more than ever, so the scientific community had some sort of self-reflection and their consciousness is now much more attracted to the development of partial intelligent modules, working together as a tool-set and having the user acting as a director of interactive tools which are helping to structure data in a fast way.

Search and image processing algorithms through interactive usage by the cognitive knowledge of operators would unfold their effectiveness to a high degree. A 100% automatic online detection of critical events would rather be a strategic roadmap or a vision. Improvement in terms of automatic surveillance for Special Forces is unrealistic at the moment. The concept of interactivity supports an important use-case: “forensic search in video material”. When investigation in the material after the event is necessary, the knowledge of the user can be incorporated in the “search question” which is not possible in the online case. We take the human in the loop, meaning that the user knows what effects a certain parameter change will have, thus he can immediately react to adjust the parameter in a way for the best results of the software.

Systems and prototypes (already developed in several national and international projects) aim for improvement on semantic or algorithmic level, or with the help of the installation of more and overlapping cameras for a specific scenario. Even with improved cameras, algorithms or setups, only one false alarm per day would be sufficient to put the complete system efficiency into question. In large installations with more than 500 cameras only one false alarm per day would be equivalent to a false alarm every three minutes. For sure, algorithmic improvements would decrease the false alarm rate, but in the online scenarios there will be always a remaining rate of false positives, not acceptable by the users.

The interactive video forensic search approach does not solely aim for improvement of detection rates through advances in algorithmic, semantic or setup topics. The innovation is concealed in the way of how the search requests are done. The normal video player (on PC) has to take a back seat. An interactive tool is taken instead of, allowing a refinement of search questions to organize amounts of data and filtering or sorting by selectable criteria given by available algorithmic modules. As an advantage, at this point the complete knowledge of the user is incorporated into the formulation of the search, meaning that during parameterization (sensitivity of an algorithm, setting of locations in the image, selection of a specific color) the user already does an intelligent pre selection. This pre selection depends on the context and would never be available before video footage already has been acquired. At the same moment this “tool” has to be configured interactively. The effect of a parameter change has to be presented to the user immediately. With such a tool it is not more in the focus to develop the best image processing algorithms or improve them. Even a “90% detection rate” would reduce all the relevant data by 90%, shrinking the whole material to a more handsome size, or – in other words – speed up search by this factor.

Imagine a scene in a surveillance video (see Figure 25), where a red car has done some harm or damage. We don’t know when this had happened, we only know the exact day – which would be equivalent of a search of 24 hours of video. With the help of some tracking algorithms we can reduce the video footage of one day to the relevant scenes of “moving objects”, still maybe some thousands. But the user knows that we are searching for a car, so he can redirect the size filters of some additional algorithms to exactly the size we are looking for, excluding persons, bicycles and motorcycles. In addition a “color selector algorithm” can further reduce the relevant scenes when all previous events are sorted by color and displayed in a ranked list. The system would give a thumbnail view of possible results.

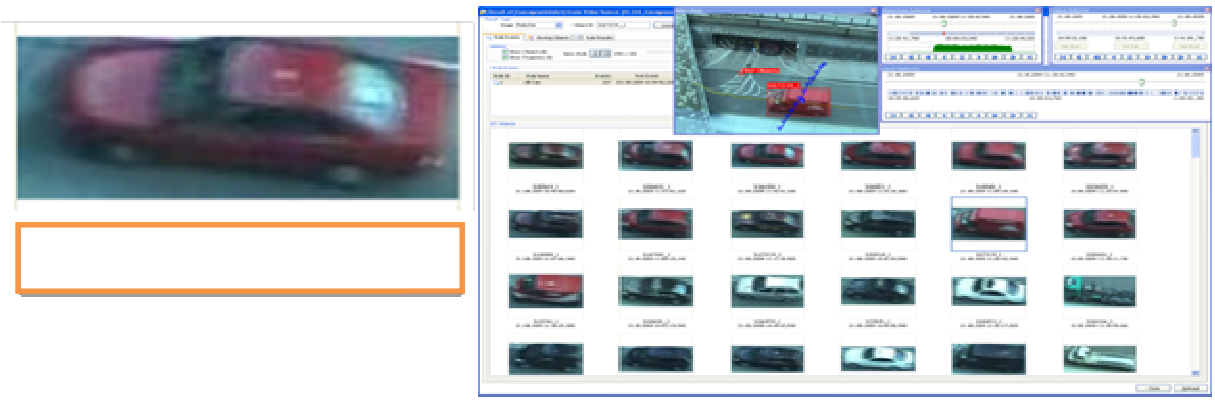


Figure 25. Video archive example: find similar object

Even with such a system and the help of above mentioned actions it is likely that the searched object is not in the first row, but we have a tool, a support-system which helps that the manual search work is reduced to a minimum, we only need to have a look to a few scenes rather that looking into 24h video footage. This would be a tremendous progress compared to the manual approach. We have three main improvements: (i) there is no more need to install algorithmic modules on each camera channel, saving costs and avoiding data preservation. (ii) the system is not a proprietary one, because it works on compressed video in various formats. (iii) There is only one installation and it is only software so upgrading is very easy.

At the moment of time being a variety of image processing algorithms exist: (face detection, face comparison, detection of lost objects, tracking of objects, logo search, license plate recognition, car type detection, person detection, and many more) all in different quality from different vendors with different interfaces and implementations. This prevents a combination of the different methods and algorithms. A future project should enable the following combination: “Show me all the faces which had been seen in combination with face X” this would allow an easy search for confederates of a suspect or guilty person. Future ideas would also show the locations of similar search results in a map of cars or persons, a temporal and spatial correlation of different searches would be of help in criminal investigations. An example is shown in Figure 26.

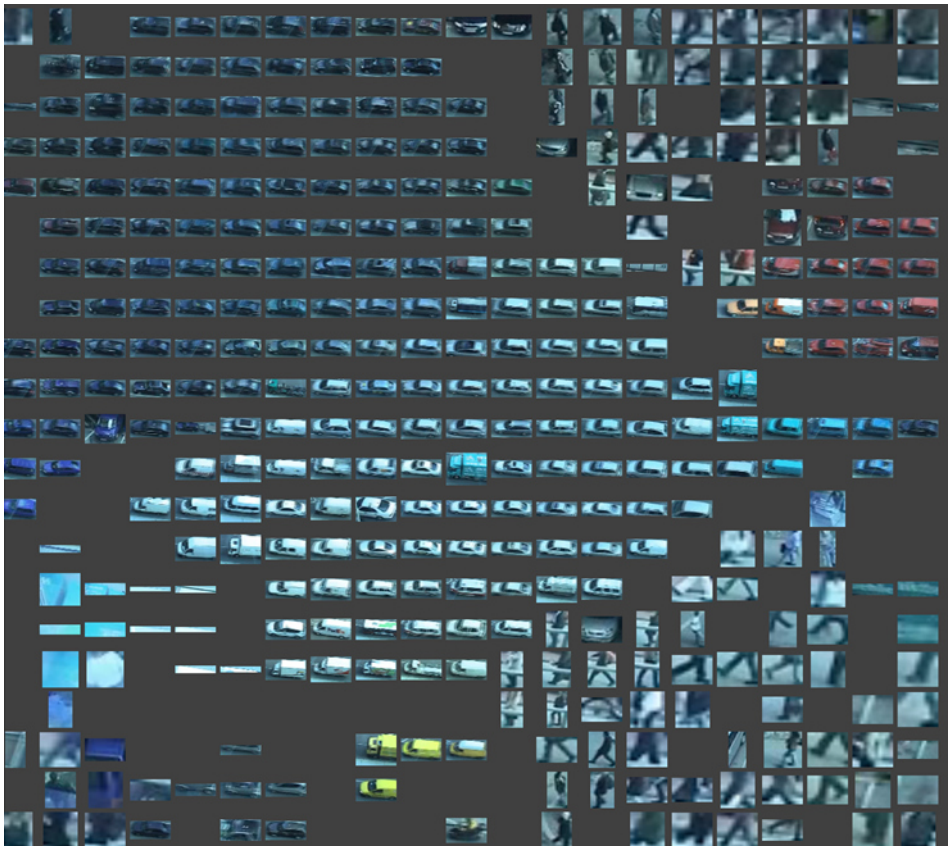


Figure 26. Clustered representation of ~350 objects (cars and incident detection)

If, after pre-filtering and selections, the remaining search space is still too large, a relief can be found in innovative visualization methods. A representation of different attributes in ordered maps would help in sorting data material or visualizing similar signatures which can be clustered automatically. The important point here is that the user selects the criteria in which way the data is clustered (e.g. color, object size, movement speed). In this way our red car would be found very fast.

The enormous speed of modern image processing systems is used for a tooling. The evaluation of the results is still done by the user where it should remain. The paradigm of an interactive archive search represents a novel approach to find critical events in video archives. More than that: new image processing methods can be included in such a system, new or different methods will enrich the pool of available modules and are happily welcomed.

The idea pursuits the development of a framework, interfaces and supporting modules for utilization of different processing modules. This includes the parameterization and embedding of new algorithmic modules, the research of innovative visualization concepts and data access. The big vision of such a system is an interactive search possibility in video data comparable to "google search" for text.

4.1.2 SALT general process to video search technology

In order to integrate SALT framework and process into the lifecycle of a video search technology project, the following steps might be considered (see Figure 27).

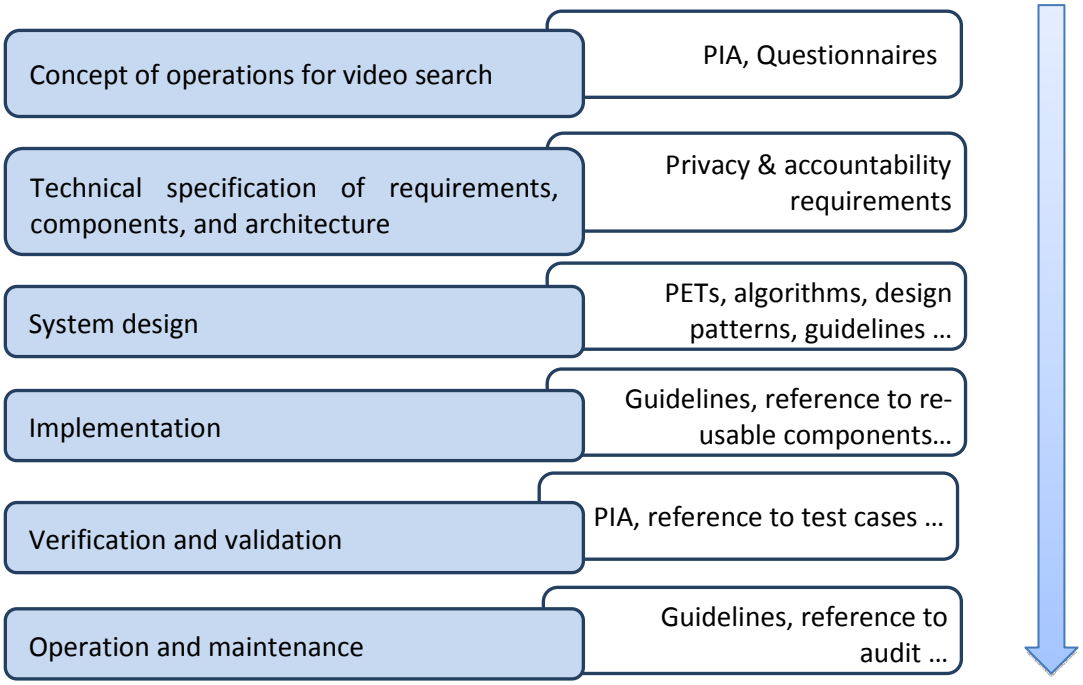


Figure 27. Example process for SALTed video archive search development lifecycle

In different phase of the development lifecycle process, one or more SALT artefacts can be applied. Generally, we can divide the SALT artefacts into (1) guidelines and (2) references to different technical knowledge throughout the lifecycle.

The guidelines can be a set of guidance with appropriate level of technical relativities to instruct the designers in the project lifecycle. The guidelines can be regarded as a "check-list" for a video archive search project.

The references can be any knowledge related to the video archive search, which should be structured and "indexed", such that a designer can find it quickly and apply it without further questions. In this sense, the reference will be a comprehensive design manual. The data controller of a video search system ought to be accountable at least for the following aspects:

- What images are archived in the first place? Who appears in them? Where and how are they stored? What is the level of detail?
- What semantics are extracted? This determines the richness of the data, which is not immediately clear from the initial recording.

- A privacy policy must be declared (specifying, in particular, what is recorded and what can be done with it).
- Legal aspects must be clarified, e.g. data protection regulation specific to a given country. These aspects may need to distinguish different cases depending on location, the level of detail, whether faces of individuals are recorded and so on.
- Data retention, deletion and forwarding.

4.2 Specialisation to biometrics technology

The key difference between the biometric systems and other type of surveillance systems is the nature of the data collected. The biometric data are more intimately connected to individuals, and in some cases they reveal additional personal information, such as health characteristics or the ethnic origin of a person. Thus, biometric systems are potentially more invasive in terms of privacy, and involve more serious ethical considerations, than other systems using different technologies.

As explained in section 4.3.3 of deliverable D2.2, biometric systems can affect all of the seven types of privacy. The degree of seriousness and the particular types of privacy affected by each system mainly depend on the architecture of the system and the specific technology used, as described in Table 1. This table also shows that the *Privacy of the Person* is the type of privacy more affected by biometric systems in general, therefore the SALT Framework should always provide for each biometric system at least some comments or recommendations to protect this kind of privacy.

Tech \ Privacy Risk	P. of Person	P. behaviour and action	P. of communication	P. of data and image	P. of thoughts and feelings.	P. of location and space	P. of association
Fingerprints	X		X	X*		X*	
Iris	X	X		X*	X	X*	
Face	X	X		X*	X	X*	
Hand Geo.	X			X*		X*	
Vein Scan	X			X*		X*	
Ear Geo.	X			X*		X*	
Palm prints	X		X	X*		X*	
Retina Scan	X	X		X*		X*	
Gait	X	X		X*	X	X*	
Voice Reco.	X		X	X*	X	X*	
Signature	X		X	X*	X	X*	
DNA	X	X	X	X*		X*	X
Multimodal systems	X	X	X	X*	X	X*	X

X depends on the system not on the technology itself*

Table 1: Impact of biometric technologies on the seven types of privacy identified by Finn et al.

From the legal point of view, although new laws are being adopted in different countries to regulate the use of biometric technologies, not every country has a specific legislation for the use of biometrics. In these cases the national regulations on personal data protection and the general guidelines on the treatment of personal data shall be applied, such as the LOPD in Spain or the Directive 95/46 at the EU level. Hence, the SALT Framework should point to the specific national laws on the treatment of biometric data if available, and otherwise refer to the general recommendations and regulations in terms of protection and trans-border flow of personal data. In case there is a special law or recommendation for a specific biometric technology, this should also be included in the SALT Framework knowledge repository.

Another concern that is specially critical in the case of biometrics is related to the principle of proportionality of the system, as biometric systems are potentially more privacy invasive. The collection and use of biometric samples must be sufficiently justified in the application context, balancing the performance of the recognition process for security purposes with the right to privacy of individuals.

The transparency is also a relevant principle to take into consideration, as once the biometric samples are collected the individuals may feel they lose control over their personal information, which in the case of the biometric data can be specially sensitive. Therefore, the organizations responsible for biometric systems must provide clear information to individuals about the use of their personal data and the different privacy and security practices implemented.

Another particularity of the biometric process, is that it is composed of two different phases: enrolment and matching. Both should be covered by the SALT Framework, that should point to the different concerns and recommendations that should be taking into consideration at each phase.

Enrolment
Several concerns arise at the enrolment phase:

- First of all the purpose of the registration of a user in the system, that should be clear and justified as explained before according to the principle of proportionality.
- The enrolment phase normally requires the cooperation of the individuals that have to be enrolled in the system. This participation should be voluntary, and in any case, the users must have the option to be unenrolled and to delete from the system all their personal data.
- Another important aspect, is the flow of the biometric data, since it is captured until it is used to generate the template and removed from the system. The data retention period should be defined, and individuals must be also informed about who is responsible for the system and how their data is going to be used and protected.
- During this phase, the most critical component of the system is the one storing the biometric templates, that should be adequately protected. In case of using a centralized storage, it should be clear whether the biometric templates are linked or not with other personal information.
- Finally, the quality of the templates stored is crucial to improve the performance of the system and reduce the false matches and all the problems that this entails.

Matching

These are the main concerns identified at the matching phase:

- As for the enrolment phase, the purpose of the recognition of users should be clear and justified. Biometric samples collected for one purpose should not be used for another without the user's knowledge.
- In some cases the matching process can be performed covertly, without any user interaction, such as in face or gait recognition systems that can capture the data at a certain distance. The individuals targeted by a SALT compliant system must be at least informed about the surveillance activities that are going to be carried out. Where possible, explicit user consent should be required.
- It is important to indicate which people have access to the data, their access rights and the policies of disclosure of information with third parties. The access to the biometric information should be limited to a specific group of authorized people.
- Lastly, the results of the matching process, as well as the biometric templates extracted at this phase, must be protected.

Finally, the components for data acquisition are normally more complex in the case of biometrics, and the features of those components that are relevant for the biometric process are different from the ones required for other surveillance systems. Deliverable D6.1 includes more information about the particular characteristics and types of sensors used by biometric systems.

The data controller of a biometric system is accountable at least for the following aspects:

- The type of collected biometric data and whether it is linked to other categories of personal data.
- The purpose for which it is to be used; special care must be taken for ethnic or health indicators.
- A privacy policy must be drafted.
- Processing of biometric data must be justified with relevant legal texts. SALT can differentiate between fingerprints, iris and so on -- those categories are relevant across countries and contexts.

5 Conclusion

The SALT general process provides useful guidelines that will help surveillance systems developers to include privacy and accountability aspects in the designs of the surveillance systems they develop. That is, it provides privacy-by-design and accountability-by-design approaches, the objectives pursued by the PARIS project.

In this document we have described what are the steps the SALT process follows in order to achieve these goals, and what parts of the process are involved in each stage. Thanks to this we can see what elements and functionalities are involved with each type of user, since they have different needs. E. g., a socio-contextual, ethical, legal or technological expert will commonly add (or update) privacy related information to the SALT repository, whereas systems designers will make use of this information.

Following the SALT process guidelines necessarily requires access to the information stored in the SALT repository, and for that matter we have the SFMT. This tool facilitates the access to the repository, enabling users to interact with the information they need. Moreover, the revised version of the SALT general process provided in this document adds new functionalities, such as the possibility of automatically check the compliance of the privacy concerns provided by the SALT references. These new tasks are carried out with the use of some other tools that complete the whole tool set, such as the UML profile (which aids system designers to create their systems designs) and the PAERIS tool (for the automatic check of the privacy concerns).

This document pays special attention to the two types of surveillance technologies, video search and biometrics, covered by the PARIS project. We have seen a detailed description regarding how the SALT process affects these surveillance systems, i. e. what do we have before and after the application of the SALT process to current surveillance systems. This is reinforced with a description of how the SALT process can be integrated within nowadays surveillance systems processes. The conclusion is a viable result whose application may lead to multiple advantages for future systems developers who want to take into account privacy aspects before deploying their systems.

Finally, we focus in the possible specialization of the SALT general process depending on the type of surveillance system (again, video search systems and biometrics systems). We address the question of possible adaptations of the SALT process to each type of systems. The document focuses in whether there are or not any parts, elements, steps of the SALT general process susceptible of being specifically designed to a particular type of surveillance system and why they are. Uncovering those parts of the process that remain the same for any system is also of interest, particularly regarding implementation purposes.

6 References

[1] A. Cavoukian, "Privacy by Design: The 7 Foundational Principles". http://www.privacybydesign.ca/index.php/about-pbd/7-foundational-principles/
[2] Samsung Techwin, "Networked Surveillance System Design Guide", July 2012.
[3] PARIS FP7 Project Deliverable D2.2 "Structure and Dynamics of SALT Frameworks".
[4] PARIS FP7 Project Deliverable D4.2 "SALT Compliance Processes Definition".
[5] Arun Hampapur, Lisa M. G. Brown, Rogerio Feris, Andrew W. Senior, Chiao-Fe Shu, Yingli Tian, Yun Zhai, and Max Lu, Searching surveillance video, AVSS, page 75-80. IEEE Computer Society, (2007).
[6] Unified Modeling Language (UML). http://www.uml.org