PrivAcy pReserving Infrastructure for Surveillance

Deliverable D3.2 SALT Framework Management Tool Implementation Specification

Project:	PARIS
Project Number:	SEC-312504
Deliverable:	D3.2
Title:	SALT Framework Management Tool Implementation Specification
Version:	v1.0
Date:	13/06/2014
Confidentiality:	Public
Authors:	Antonio Maña (UMA) Francisco Jaime (UMA) Fernando Casado (UMA) Marioli Montenegro (UMA) Fernando Gómez (UMA) Christophe Jouvray (Trialog)

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

Table of Contents

Document History

Version	Status	Date
v0.1	Document structure	16/05/2014
v0.2	Added Sections 2 and 2.1	20/05/2014
v0.3	Added Section 3.1	21/05/2014
v0.4	Changed document structure, content on Section 3	22/05/2014
v0.5	Added introduction and Section 2.2	22/05/2014
v0.6	Added Section 3.2.2.1	23/05/2014
v0.7	Added Executive summary	26/05/2014
v0.8	Added conclusion and table of abbrev. Text revised	02/06/2014
v0.9	Added content to Section 3	03/06/2014
v0.10	Added content to Section 3	04/06/2014
v0.11	Revision of Section 3.1	05/06/2014
v0.12	Revision of Sections 2, 2.1 and 3.2.1	09/06/2014
v0.13	Revision of Sections 3.2.2 and 3.3	10/06/2014
v1.0	Final version	13/06/2014

Approval		
	Name	**Date**
Prepared	Francisco Jaime, Christophe Jouvray	10/06/2014
Reviewed	Denis Butin, Maria Cinta Saornil Gomez	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 detailed information about the SALT Framework Management Tool (SFMT), including its specification, its architecture and its implementation plans.

In first place we provide a brief description of the SALT Framework, just to remind the main concepts to the reader, explaining what it is and where its information comes from. Then we move to the SFMT functionalities and the type of users that will handle it, paying attention to the mechanisms the tool uses to provide trust to the information that it manages, and the different forms of representing such information. Next the modelling features are exposed, together with a justification about having used UML. The key modelling concepts, such as

stereotypes, are explained in depth. This part of the document ends with an explanation of the foreseen reasoning features that we think would be useful in the implementation of the final version of the tool.

The second part of the document presents the specification of the SFMT, based on the requirements defined in deliverable D3.1 and the SALT concepts defined in WP2. It begins with an explanation of the different scenarios where the SFMT is used regarding the different types of users, and next it provides a detailed description of the whole toolset architecture, indicating the different elements involved and the links among them. Yet, these elements are explained separately, indicating their requirements (inputs) and functionalities (behaviour). Finally, there is a section regarding the validation criteria.

List of Figures

List of Tables

Abbreviations and Definitions

Abbreviation	Definition
OCL	Object Constraint Language
PAERIS	PrivAcy-by-design EngineeRIng aSsistant
PARIS	PrivAcy pReserving Infrastructure for Surveillance
SALT	Socio-contextual, ethicAl, Legal, Technological
SFMT	SALT Framework Management Tool
SUD	System Under Development
UML	Unified Modelling Language
WP	Work Package

1 Introduction

This document describes the SFMT (SALT Framework Management Tool) focusing on the implementation plans, its architecture and design specification. Therefore, even though the SFMT is tightly related to the SALT framework and the SALT general process, we will only discuss here the parts that are needed for the sake of the on going explanation. To know about these matters, we refer the reader to project PARIS deliverables D2.2[4] "Structure and Dynamics of SALT Frameworks", D4.2[6] "SALT Compliant Processes Definition" and D4.3[7] "SALT Compliant Processes Guidelines for Use Cases".

The SFMT has an important role within the SALT general process, since it serves as an entry point to the privacy and accountability information regarding surveillance systems that is stored in the SALT repository. Due to the diverse types of users that may use the SFMT and the range of different functionalities that it provides, it requires a flexible design.

For the development of a fully operational tool that provides a solution for all our needs, we have to take into account some important criteria:

- What type of user is going to use the tool: socio-contextual/ethical/legal/technological experts, surveillance systems designers, public authorities, etc.
- What privileges are going to be assigned to each user: that is, what restrictions and limitations are going to be applied to each user.
- What functionality do we expect to have for each case: adding information to the repository, searching SALT references, etc.
- What (non-functional) properties do we want our tool to have: it can be scalable, easy to maintain and update (modularized), etc.

Once we answer these questions, it is possible to develop an initial architecture of the tool, which provides a clear view of all the components and how they are related. This is an important step, since it allows knowing what component or connection should be modified if future changes are required.

Then we come to the tool design, where we can describe the inputs and outputs for each tool component, i. e. what behaviour we expect to have and what are the requirements for this to happen. Besides, it is remarkable that there usually are many ways of achieving a given feature or functionality, thus we have to take a decision about what option better fits with our particular needs.

In this document, we clarify this development process regarding the SFMT. We answer all these questions and we describe the decisions we have taken. In the following sections we provide a complete tool description, its main features, what functionalities we expect to have, the tool architecture we have conceived and the design for each component. Therefore, after reading this document, the reader can have a clear vision of what the SFMT is for, what it does, how it does it and how it is internally organised.

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

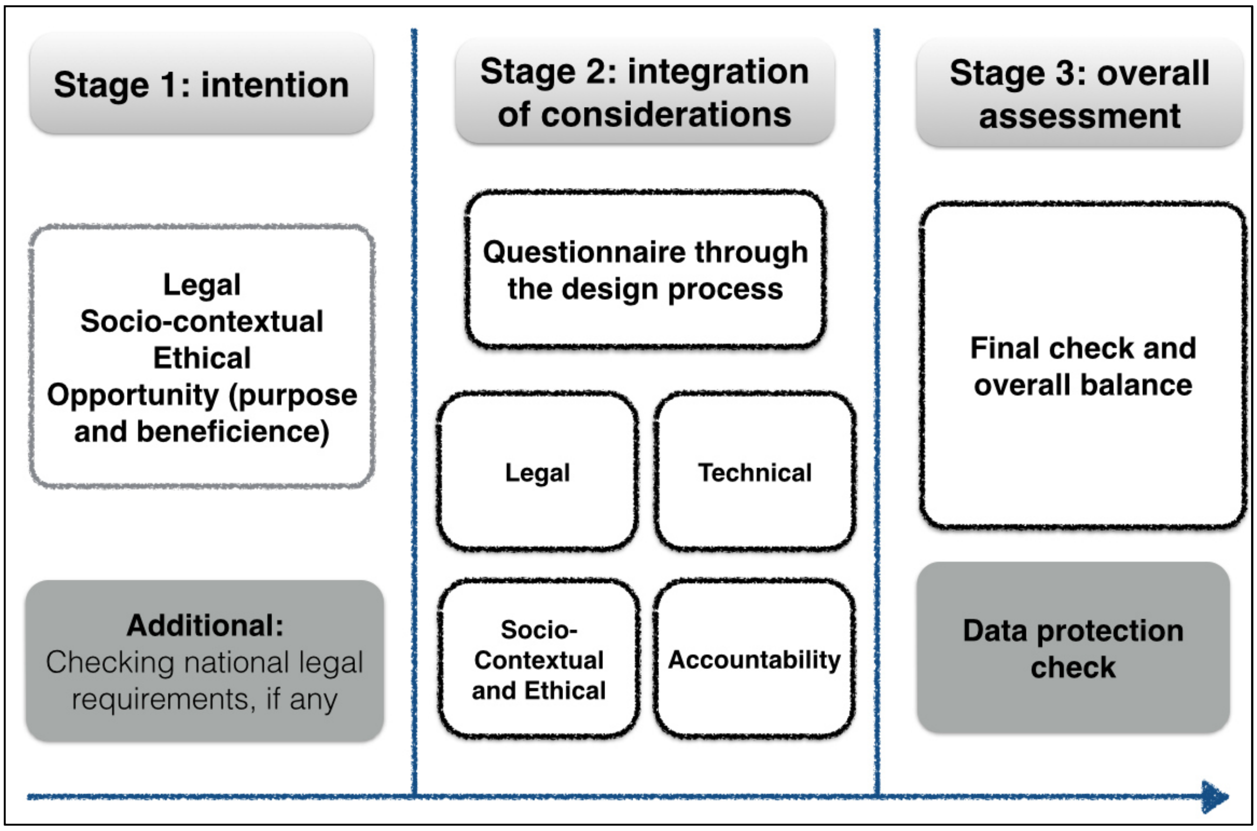

Stage 1: intention
Legal
Socio-contextual
Ethical
Opportunity (purpose and beneficience)
Additional: Checking national legal requirements, if any
Stage 2: integration of considerations
Questionnaire through the design process
Legal
Technical
Socio-Contextual and Ethical
Accountability
Stage 3: overall assessment
Final check and overall balance
Data protection check

Figure 1 Three stage process for SALT Framework

2 Tool Description

The SFMT (SALT Framework Management Tool) is a tool developed within the scope of the PARIS project. Its main task is to assist any user who needs to access the SALT framework in one or another way.

The SALT framework itself is a set of information coming from the knowledge of experts in several areas of expertise: socio-contextual, ethical, legal and technological. This information is focused to the privacy and accountability aspects to be taken into account during the design phase of a given surveillance system (video-surveillance system or biometric system). This said, it is clear that managing the SALT framework information boils down to two main functionalities: the addition/update of such information, and the retrieval of the appropriate information for each particular case. These are the two main tasks covered by the SFMT, although several subtasks are also involved.

Therefore, we could differentiate here two main types of user for the SFMT:

1. Experts from the four different categories (areas of expertise): who are going to provide the information to the SALT framework.
2. Systems designers: who are going to search the SALT framework for the privacy and accountability information that may be applicable to the surveillance system they want to develop.

An important property to be fulfilled by the SFMT is the provision of trusted information to the users, i.e. some kind of mechanism that guarantees the source of the information. This objective is achieved by means of a digital signature. Any expert who wishes to add new information to the SALT framework has to be logged into the SFMT as an accredited author. This is done by means of a digital certificate, which each user must have in order to use the SFMT. Then, whenever an author provides new information to the framework, he will provide a digitally signed copy of the information, permanently linking the information with its corresponding author.

The information provided by an author is gathered within an element called SALT reference, which may contain one or several privacy or accountability concerns. Each of these concerns can have three different representations, two of which are mandatory. The SFMT will provide support for these three possibilities:

1. Textual description of the concern. This is the easiest and more straightforward representation. This description is directly provided by the experts and it is mandatory, since it provides the real explanation of the concern.
2. Another (textual) description that points out how its corresponding privacy/accountability related concern can be addressed by a particular surveillance system design. This description can also be provided by experts, although it may require some technical knowledge to produce it. Besides, this description is also mandatory, since it is what systems designers will look at and follow in order to apply the concerns to their systems under development. There may be several different ways of

implementing a given concern, but at least one possible implementation has to be included with each concern.

3. A formal description of the concern. This description is done with OCL (an Object Constraint Language that provides formal representations) rules, which will later be used by a tool that will check whether the concerns were properly applied to the system concern or not. This representation requires an OCL expert who is also aware of the meaning of the concern. Besides, some concerns are very wide, vague, ambiguous or context dependent, therefore providing a formal representation is not always possible. For these reasons, this representation is not mandatory.

Concerning the addition/update functionality, the SFMT has to provide a mechanism that allows providing the information to the SALT framework following the previous representations.

Also having the expert's point of view in mind, it is important to remark that socio-contextual, ethical and legal experts may not have a deep understanding of technical knowledge and even working with the SFMT may be a hard task for them. However, in an attempt to lighten their tool usage, we have conceived the possibility of having a specific implementation of SFMT in order to enhance the usage to non-technical users. In particular, we studied how to transform the information contained within the SALT framework into a wiki-based representation, which is easier to understand and more accessible.

A questionnaire centric interface will also be provided. This interface will allow experts to introduce questionnaires into the base of knowledge, although these questionnaires are not SALT references themselves, so they will be kept separately. Systems designers will later make use of such questionnaires to check the privacy and accountability compliance of the surveillance systems they intend to design.

Now, let us consider systems designers' point of view. For them, the SFMT provides an interface that allows for searching the SALT references containing the concerns with the privacy and accountability information that may be of interest for the surveillance system they are developing. Once the exact references are located, the SFMT will show in an understandable way the descriptions of the concerns and possible ways of implementation to the users. OCL rules can be hidden to system designers, since they may not exist, and in the case they do, they will be used by an automated tool, but not the designer. In case the OCL rules are not present, an automatic validation of the concerns cannot be performed, however we still can help a human validation by providing relevant information.

Note that the SFMT may find different SALT references regarding the same concerns, but with different implementations. As we said above, there are more than one possibilities of implementing one single concern, thus it should be normal to find these circumstances (this happens because different experts introduced different SALT references regarding the same concerns, but they conceived different implementations). Due to this fact, we have devised the possibility of a reference score, i. e. system designers will be able to evaluate and provide a score about how good (or bad) a SALT reference is. This score will be seen by future users. The ranking methodology used by the SFMT makes use of some algorithms specifically designed for this task.

Finally, we also intend to use other separate functionalities/tools:

1. An UML[9] profile that will help system designers to create the design for the surveillance system under development. This is very handy, since systems designers are not supposed to be UML experts.
2. A tool that will automatically match the OCL[10] rules against the system design, and check whether the privacy and accountability concerns where properly implemented in the system design or not. Initially, we have decided to call it PAERIS (PrivAcy-by-design EngineeRIng aSsistant).

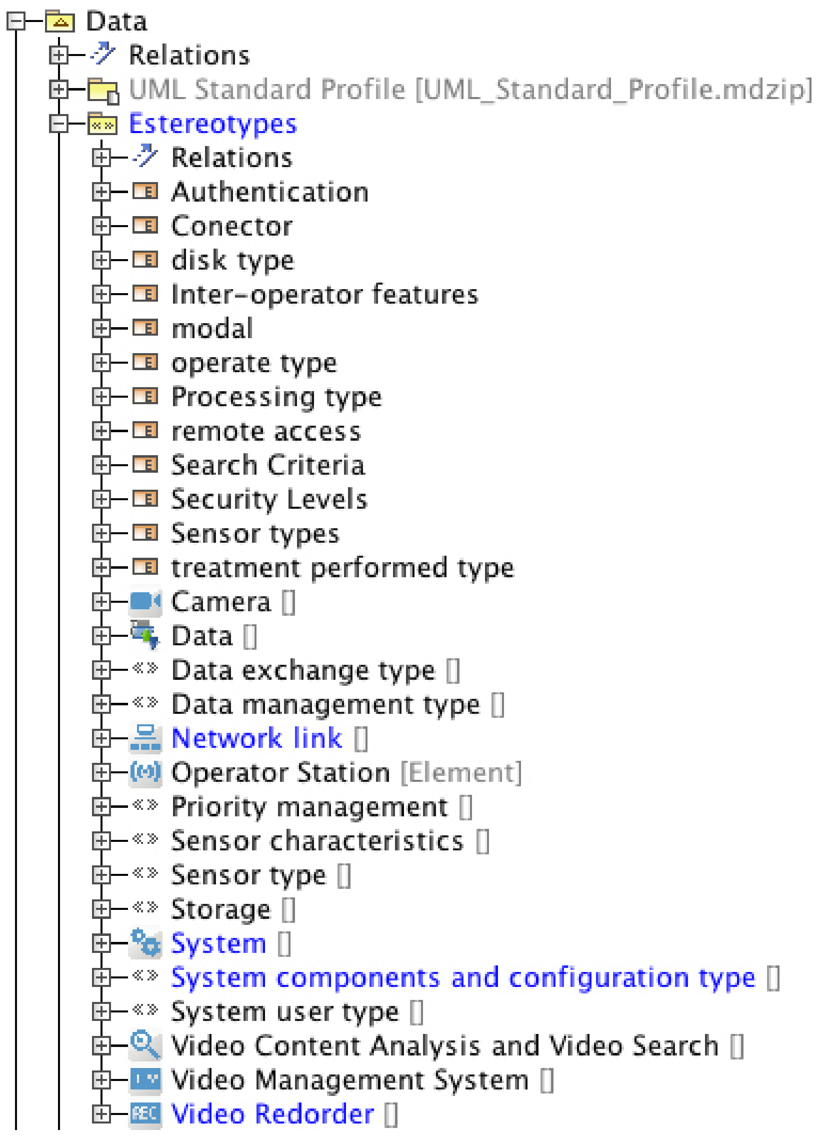
2.1 Modelling Features

All modelling features and artefacts included within the PARIS tools have been performed using UML (Unified Modelling Language), which is a general purpose language used as a standard way to provide systems designs. UML is a very good candidate for this task, since it includes the necessary features, and it also is the most known and used modelling language at the moment. We can see UML as a graphical language for visualizing, specifying, constructing and documenting a system. UML can be used to define a system, its artefacts and methods, to document the system, and subsequently with the elements created, build it.

The modelling artefacts that we have developed regarding the SFMT are included within the UML profile we have specifically created for surveillance systems. The UML profile provides a generic extension mechanism for customizing UML models for particular domains and platforms. Consequently, we have produced an UML profile that contains a set of artefacts intended to model the different elements susceptible of being part of a surveillance system. These artefacts cover the two types of surveillance systems of interest for the PARIS project, i.e. video surveillance systems and biometrics systems.

In the proposed UML profile, all elements of surveillance systems have been modelled using stereotypes. A stereotype is an extensibility mechanism of UML that allows an UML designer to extend the vocabulary of UML in order to create new modelling elements. These new elements are derived from existing ones, but they have specific properties, which are suitable for a particular problem domain or otherwise specialized usage[1]. Therefore, using our UML profile a complete surveillance system can be designed by means of stereotypes.

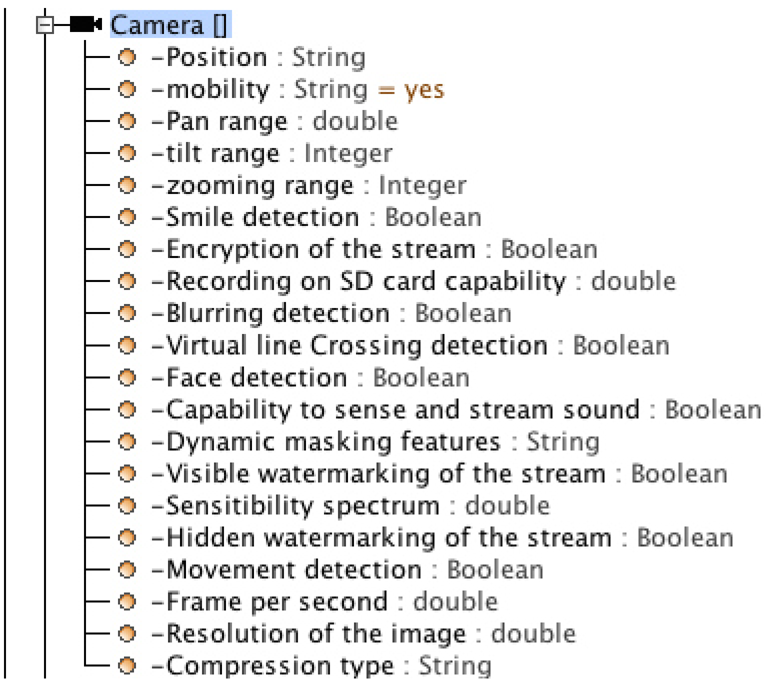
[1] http://en.wikipedia.org/wiki/Stereotype_(UML)



- Data
  - Relations
  - UML Standard Profile [UML_Standard_Profile.mdzip]
  - Estereotypes
    - Relations
    - Authentication
    - Conector
    - disk type
    - Inter-operator features
    - modal
    - operate type
    - Processing type
    - remote access
    - Search Criteria
    - Security Levels
    - Sensor types
    - treatment performed type
    - Camera []
    - Data []
    - Data exchange type []
    - Data management type []
    - Network link []
    - Operator Station [Element]
    - Priority management []
    - Sensor characteristics []
    - Sensor type []
    - Storage []
    - System []
    - System components and configuration type []
    - System user type []
    - Video Content Analysis and Video Search []
    - Video Management System []
    - Video Redorder []

Figure 2. UML profile for biometric systems

Even though the above paragraphs may be seen as "technically sound", the inclusion of an UML profile leads to an ease of usage of the SFMT from the systems designers' point of view. Thanks to this functionality we are able to provide an interface that abstracts the use of UML and allows systems designers to create the design of a surveillance system without having to know about UML. In this way, with the UML profile what the user sees is just a set of icons corresponding to surveillance elements. Behind these icons are the previously mentioned stereotypes, but the user does not need to know about this, he will just use them. Besides, each stereotype has a set of attributes corresponding to the main characteristics of the surveillance element they represent. The system designer will be in charge of assigning a value to each attribute.



```
Camera []
  -Position : String
  -mobility : String = yes
  -Pan range : double
  -tilt range : Integer
  -zooming range : Integer
  -Smile detection : Boolean
  -Encryption of the stream : Boolean
  -Recording on SD card capability : double
  -Blurring detection : Boolean
  -Virtual line Crossing detection : Boolean
  -Face detection : Boolean
  -Capability to sense and stream sound : Boolean
  -Dynamic masking features : String
  -Visible watermarking of the stream : Boolean
  -Sensitibility spectrum : double
  -Hidden watermarking of the stream : Boolean
  -Movement detection : Boolean
  -Frame per second : double
  -Resolution of the image : double
  -Compression type : String
```

Figure 3. A stereotype modelling a camera

At this time of the project development, the UML profile is not completely finished. The set of icons for representing surveillance elements has not yet been created, but a whole set of stereotypes and attributes covering biometrics systems has already been developed. Figure 2 shows a screenshot of the group of stereotypes. System designers will just need to drag and drop these components into their systems designs and then fill in the values for the corresponding attributes. As an example, Figure 3 shows a complete definition of the *camera* stereotype with all its attributes.

2.2 Foreseen Reasoning Features

This section describes the reasoning features we expect the SFMT to have, even though some of them are not yet implemented at this stage of the project. In order to follow a systematic and comprehensive definition of all the foreseen features, we use a template with the following information for each reasoning feature.

- Main actor: the main type of users that are going to make use of the reasoning feature, even though there may be some other users involved.
- Main process: the main functional process where the reasoning feature is used. Again, the reasoning feature may also be used by some other processes as well.
- Functionality: description of the expected functionality of the reasoning feature.

2.2.1 Privacy informer

Main actor	Systems designers.
Main process	Surveillance system design.
Functionality	The PARIS toolset includes advanced browsing and searching capabilities that will help system designers obtaining relevant information regarding the privacy and accountability of their systems.

Table 1. Reasoning feature: Privacy informer

2.2.2 Stakeholder agreement assistant

Main actor	Expert.
Main process	Creation of a SALT reference.
Functionality	Tools to support the process of creating a SALT reference based on the views of multiple stakeholders. This functionality is achieved by means of a questionnaire.

Table 2. Reasoning feature: Stakeholder agreement assistant

2.2.3 Rule-based concerns realization assistant

Main actor	Systems designers.
Main process	Surveillance system design.
Functionality	The tool will provide information to system engineers about the way to address a concern in the design of the SUD (System Under Development) based on the information included in SALT references.

Table 3. Reasoning feature: Rule-based concern realization assistant

2.2.4 Rule-based continuous validation

Main actor	Systems designers.
Main process	Surveillance system design.
Functionality	Once a concern has been addressed in the SUD design, its rules continue to be active to prevent that at a later stage of the design these rules are violated. That is, the tool provides a constant on-the-fly checking.

Table 4. Reasoning feature: Rule-based continuous validation

2.2.5 Documentation validator and generator

Main actor	Systems designers.
Main process	Documentation generation.
Functionality	The tool will guide SUD engineers in providing adequate documentation for the privacy-related aspects of their design in order to ensure consistent future evolution of the SUD and to facilitate the SALT compliance verification by an external actor.

Table 5. Reasoning feature: Documentation validator and generator

2.2.6 Assisted compliance checker

Main actor	External auditor.
Main process	Check the SALT compliance of the system.
Functionality	The documentation generated, together with the privacy-enhanced SUD design, can be fed to the privacy compliance evaluator. The tool can help the process of assessing the SALT compliance of the SUD by automatically checking some aspects that facilitate the task of the compliance evaluator. For instance, the tool can ensure that only some parts of the design must be evaluated for a given concern, thus saving time and effort to the evaluator. A simple concrete example can be that the tool can identify all operations that manipulate privacy-sensitive information, thus avoiding the effort of checking all other privacy-irrelevant operations.

Table 6. Reasoning feature: Assisted compliance checker

3 Specification of the SFMT

This section aims at providing the specifications of the SFMT based on the requirements defined in the deliverable D3.1 and the SALT concepts defined in WP2. The specification will be used in order to develop the SFMT in the next tasks in WP3.

3.1 *Scenarios Using the SFMT*

This section explains the scenarios, which will covered by the SFMT.

3.1.1 Expert Perspective

Experts will use the SFMT for two main reasons: (i) to browse the repository in order to see and/or check the knowledge already stored and (ii) to create or to edit some knowledge. The role of the SFMT is to help him in his tasks the better as possible.

Browsing the repository

The repository can contain various and heterogeneous information (i.e., knowledge). Indeed, as explained in the deliverables D2.1 and D2.2, experts will need to store questionnaires, taxonomies, technical information like system architecture or PET descriptions but also crossed domain data (e.g., Privacy Harms Analysis such as PIAs). It is crucial that the navigation and the search functions have to be very intuitive and efficient.

As defined in the requirements (see the deliverable D3.1), it is mandatory that the SFMT should interact with the user in different ways in order:

- To show information according to the user concerns. The body of knowledge will contain a lot of information. In order to avoid the risk of flooding the user with an excessive amount of information, the browser needs to use an efficient way to navigate in the repository and to filter the information in order to display only the user concern.
- To use the suitable view to the user in order to display a knowledge. A same information can be used by different type of experts. For instance, a legal expert could need to view all the details of a law including all articles. Conversely, a technical expert should need only an abstract.

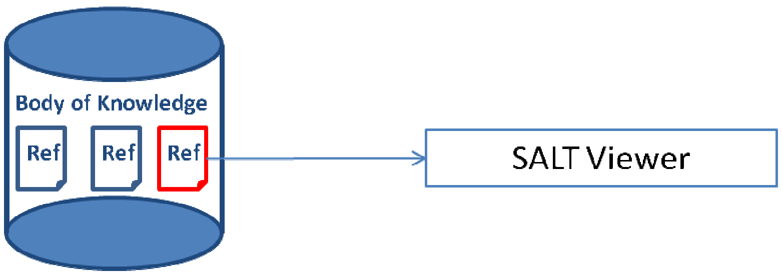


Figure 4. SALT Viewer used by Experts

According to the reference type and the user, specific tools can be used. For instance, in the legal or ethical domains, some questionnaires will be created. These questionnaires will help the designer to check if his system conforms to laws or citizen perceptions. Figure 5 shows a specific instance of the SFMT dedicated to questionnaires. In other words, the SFMT is an aggregation of tools specialized in specific concerns.

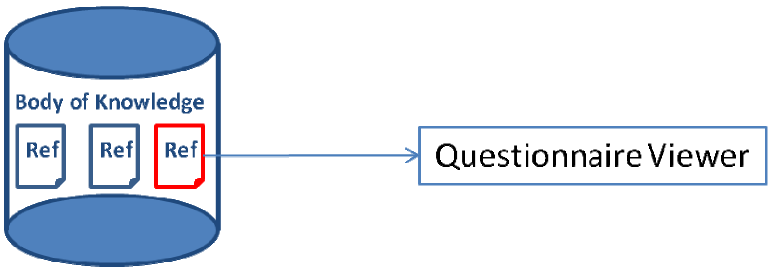


Figure 5. Questionnaire Specific Viewer

In order to navigate and select how information will be represented, the SFMT will provide a "View" mechanism as depicted in Figure 6. The principle is to associate some tags to the knowledge (i.e., a reference). An information can be screened by several views and the results will be adapted to the user concerns. For instance, it will be possible to point out the following:

- Domains like legal, social, ethical, and technical.
- Legal references like Europe, Countries, USA, etc.
- Other useful information such as a methodology or socio-ethical references.

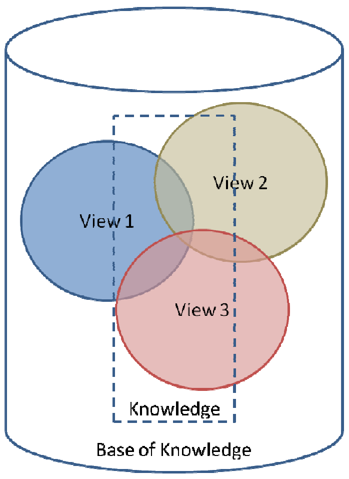


Figure 6. View-based Browsing According to Expert Concerns

At the end, when the user browses the repository, he will be able to filter the content thanks to parameters or keywords.

Concrete examples of the distinction between the views and the knowledge access will be presented in Section 3.2.2.3.

Creating or editing the content of the repository

The repository will contain some knowledge provided by the experts. The SFMT has to allow the creation of new content and to update it in case of practices or regulation evolution. The editor has to provide some flexibility in order to fit with the expert requirements. Both scenarios regarding the creation and the edition are detailed in the following.

Figure 7 shows how the SFMT will create new content in the repository. In this example, the questionnaire use case has been selected. The similar approach is used for other possible content. In order to maximise the trust in the repository content, every reference has to be accountable. For this reason, the SFMT will store also the editor name (i.e., expert who made this reference), the date, etc.

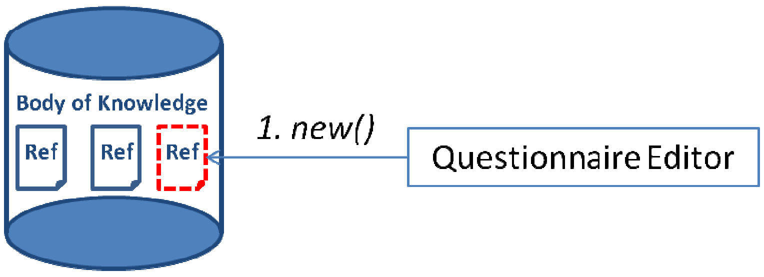


Figure 7. Adding New Content in the Repository

Since the knowledge can be used by different users (i.e., experts or system designer), it is important to understand the terminology used by a reference. For this reason, inside a reference content, the SFMT will automatically highlight the words which have been defined. Thus, the expert who edits the reference can verify if he has the same definition in mind. An expert can also indicate to SFMT new important words and define them.

Moreover, all contents stored in the repository should be editable by the same expert or someone else. According to the purpose of the SALT reference, the suitable editor is selected as for the reference creation. Figure 8 explains the interaction between the repository and the SFMT (i.e., in this example, the questionnaire editor instance is used). The SFMT has to load the reference, and then the expert captures the update. Finally, the SFMT records the new version in the repository. For traceability and governance reasons, the new content has to be signed by the expert and the previous reference is kept. It means that a new version does not replace the previous version and also needs to be signed.

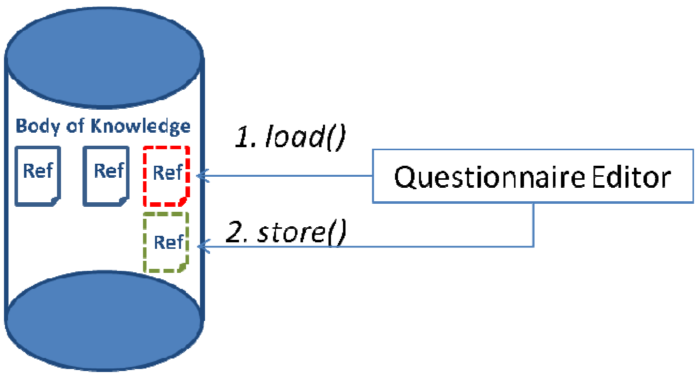


Figure 8. Editing Content in the Repository

A reference will not be an isolate knowledge. Indeed, some relations between several references can exist. For instance, the following example highlights the relations between different taxonomies:

1. A technical expert introduces a taxonomy on surveillance technologies
2. A legal expert develops a taxonomy on privacy harms
3. A technical expert defines a taxonomy in privacy enhancement technologies

There are some relations between the taxonomies as depicted in Figure 9. These relations have to be defined since the system designer will need to take into consideration the privacy harms raised by a surveillance system and to select the appropriate PETs.

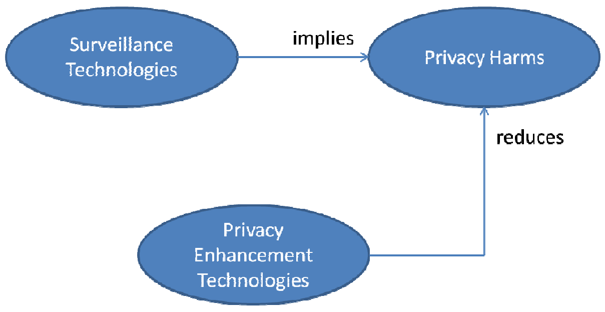


Figure 9. Relationships between Taxonomies

Since the need to capture relationships between the references exists, it is also to ensure the consistency even after an evolution. Indeed, a modification of a reference by an expert can trigger a flood of new updates. For instance, based on Figure 9, if the technical expert adds a new surveillance technology, it is necessary to define the privacy harms of this new technology. The SFMT has to raise an alert in order to get the new information in order to maintain the consistency of all references.

3.1.2 System Designer Perspective

Browsing the repository

To browse the repository can be a tricky task since it can contain a lot of information. It is necessary to filter the content according to the material searched by the system designer. Since some concepts are not technical, the designer can have difficulties to understand the purpose of contents. For this reason, the words defined in the terminology are underlined in the text. When the designer hovers over a word, a tool tip appears with for instance the definition of the concept.

Importing Content to System Developers Tools

Designers can use various tools for developing their systems. When the designer is screening the SALT repository, he should be interested in reusing some knowledge for the design of his system. The SFMT provides a feature which allows the designer to import some content in his development tool.

Running Questionnaire-based SALT Content

In order to take into consideration non technical concerns such as legal, social, or ethical, the PARIS project experts consider that using questionnaires is an efficient way. During the development of the system, the designer will run questionnaires. SFMT has to provide a flexible way to run a questionnaire in terms of:

- Sequence of questions (i.e., non linear questionnaire)
- Interruptible questionnaire (i.e., the expert can start the questionnaire, stop to answer and resume later during the project).

The answers have to be stored in order to be reused later, for instance, during an audit.

Provide Feedback to SALT Experts

The SFMT has to provide a way for getting feedback, bad or good experiences from the experts. This functionality will help:

- The experts to enhance current references
- The system designers in order to select the good references and to use them correctly

3.2 *Tool Design Specification*

According to the scenarios defined in Section 3.1, the SFMT cannot be an isolated and unique tool but will be a set of tools. All of them will be in interaction with the SALT repository in order to get the SALT content.

3.2.1 Overview of the Toolset Architecture

There are several components that have to be taken into account for an overall description of the tool architecture. Figure 10 shows a general architecture of the SFMT and how it is related with the SALT repository and the rest of external tools.

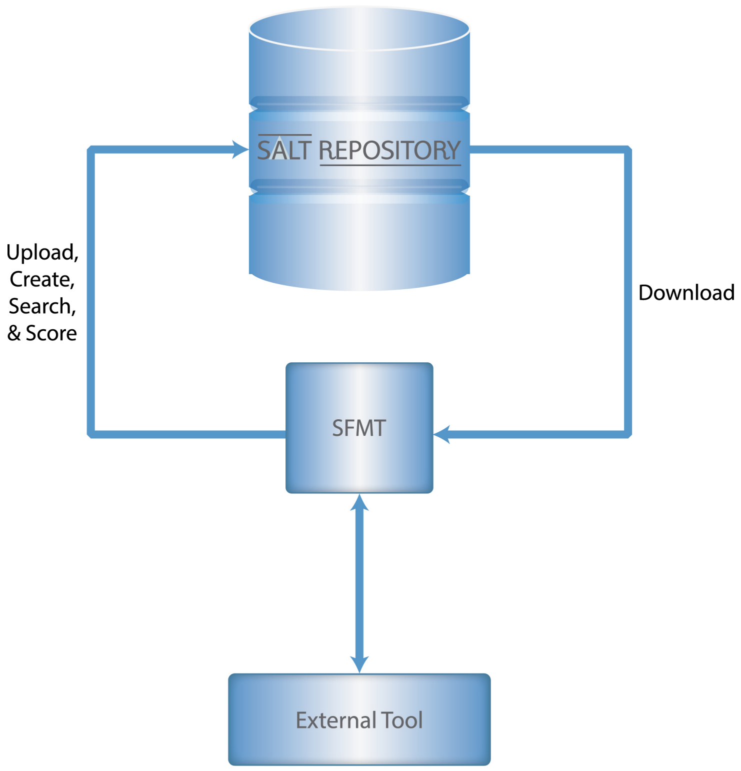


Figure 10. General toolset architecture

From this general architecture we can derive two more specific architectures regarding the two main approaches of the SFMT: the user independent repository interface and the user dependent SALT editor. Figure 11 shows how the web-based SFMT (user independent interface) interacts with the rest of elements of the architecture, with the SALT repository as the main information provider. On the other hand, Figure 12 and Figure 13 show the architectures for the wiki-based SFMT and the questionnaire centric SFMT (user dependent interface).

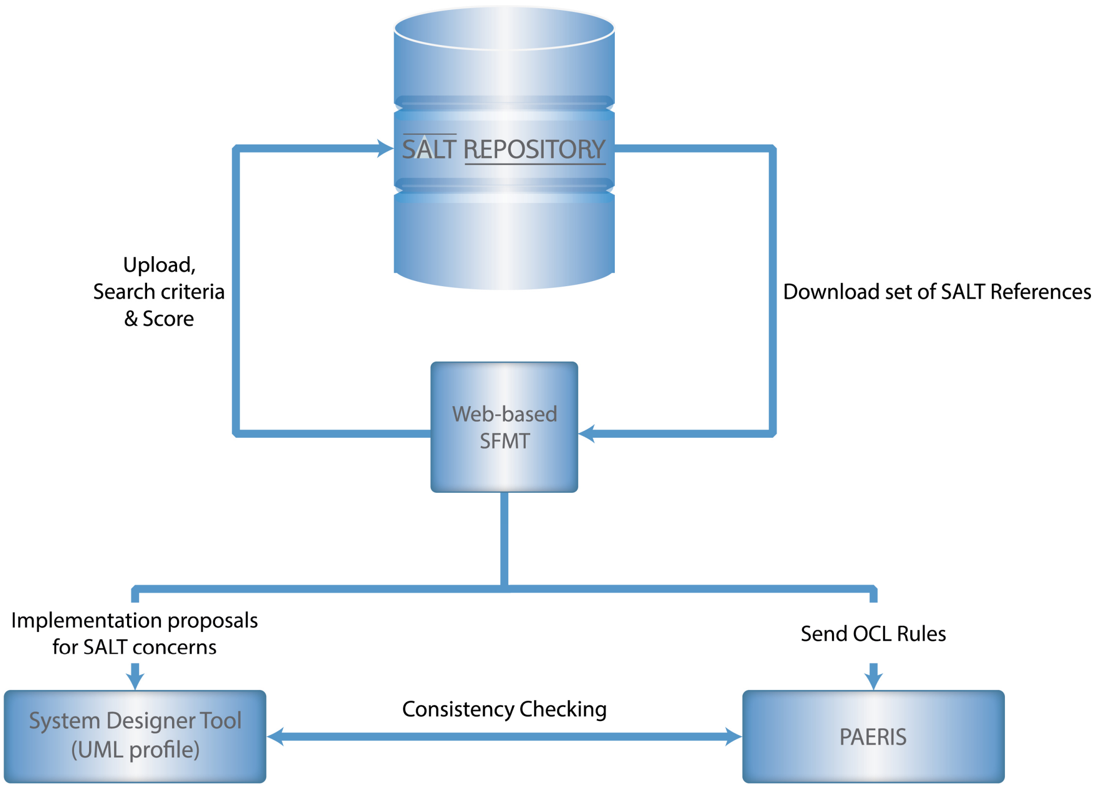


Figure 11. Architecture for the user independent repository interface

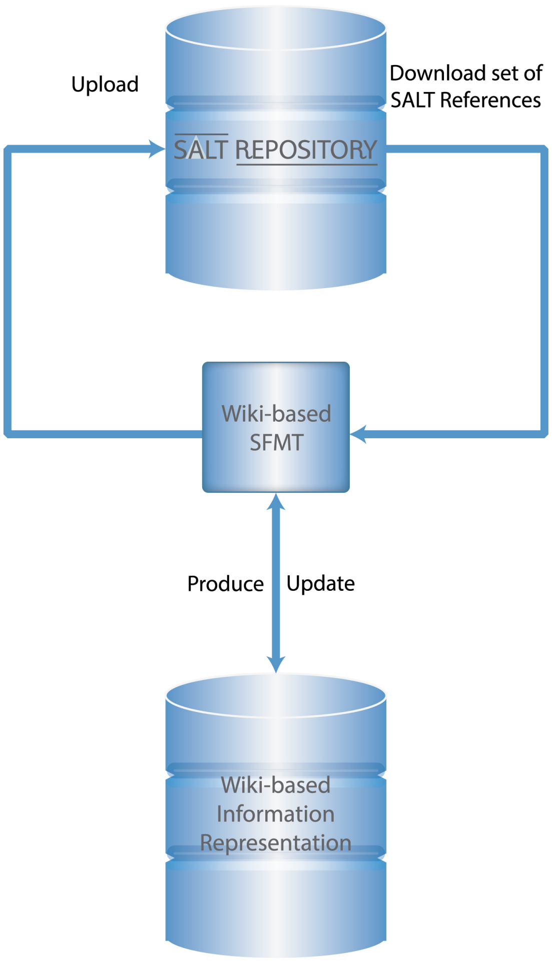


Figure 12. Architecture for the user dependent SALT editor

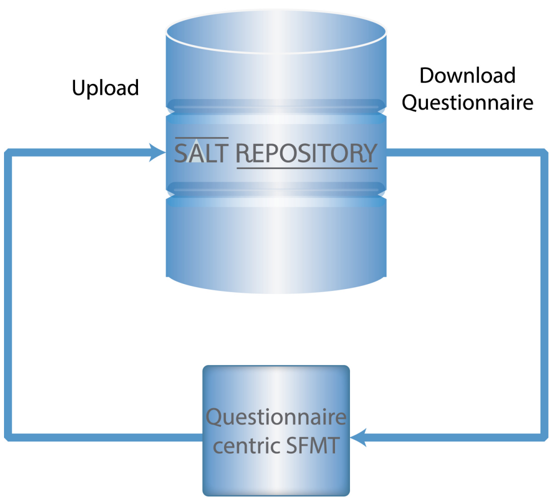


Figure 13. Architecture for the questionnaire centric SFMT

We can see how the two modules of the SFMT, the web-based and the wiki-based, interact with the rest of components. Since they are going to handle SALT references, they both need a connection with the SALT repository, i.e. the place where the SALT framework is stored (typically a database system). Since the information stored in the repository (the experts' knowledge) is a key part of the whole PARIS project, the two approaches are connected to the repository in a way that allows for adding and updating SALT references. This type of connection will mainly be used by experts in the four categories of expertise: socio-contextual, ethical, legal and technological.

The web-based SFMT may also connect with the SALT repository for other different purposes, such as sending to it a search criteria corresponding to a given surveillance system in order to retrieve the proper SALT references that are relevant for that surveillance system; or uploading the scores showing how (positively or negatively) users rated a given SALT reference. This type of connection will be typically used by systems designers.

Two more architectural components are related to the web-based SFMT, the one that provides the UML profile functionality and the PAERIS component. The last one serves as a link between the SFMT and the UML profile, since it requires the OCL rules provided by the SFMT (which are obtained from the SALT repository,) and the UML model of the system design created by the system designer, with the help of the UML profile (see Figure 10).

On the other hand, the wiki-based and questionnaire centric SFMT requires to upload and download the questionnaires that are used to check whether the privacy concerns have been properly addressed or not. This functionality also requires a connection to the SALT repository, since questionnaires need to be stored too. However, we have to keep in mind that questionnaires are not ordinary SALT references, thus even though they can be stored alongside SALT references, they have to be kept separately.

The last major component is the place where wiki-based representation of SALT references reside. This component is directly connected to the wiki-based SFMT. Thanks to this, the SFMT can transform SALT references, obtained from the repository, into a wiki-based representation that will be displayed to users. This is a more friendly representation mainly targeting socio-contextual, ethical and legal experts.

Table 7 summarizes the list of major components of the whole toolset.

Component	Short description
SALT repository	It stores privacy/accountability concerns related to surveillance systems.
Web-base SFMT	User independent repository interface.
Wiki-base SFMT	User dependent SALT editor.
Questionnaire centric SFMT	Tool in charge of handling questionnaires.
UML profile	It aids system designers to develop surveillance systems designs.
PAERIS	Provides automatic validations for OCL rules (formal representation of privacy concerns within SALT references).
Wiki-based repository	It stores the wiki-based representation of SALT references.

Table 7. Toolset main elements

3.2.2 Tool Design and Implementation Proposals

In this section, the design of the different instances of the SFMT is explained.

3.2.2.1 User Independent Repository Interface

This section explains the design of the web-based SFMT, thus it describes what tasks the tool carries out and what are the requirements (inputs) for a given behaviour to happen. For a better understanding, we can divide the operation of the tool into three parts, according to the main functionalities of the SFMT: managing the SALT repository, assisting the surveillance system design, and validating that privacy concerns were properly applied to the system design.

Regarding the repository, this is the place where SALT references are stored, and the actions related to it must allow for the creation, update, browse and search of these references. The design assistance is related to how the tool can help systems designers in their way to achieve a SALT compliant system design. The validation phase is focused to checking whether the system design fulfils or not the privacy concerns provided by the SALT references. Moreover, let us remark that the validation phase can be performed on-the-fly while the system design is being developed.

3.2.2.1.1 Managing the SALT repository

Creation of a SALT reference:

- Requirements: an expert (from the socio-contextual, ethical, legal or technological area) must have a knowledge related to privacy and/or accountability concerns regarding surveillance systems within a determined context.
- Behavior: the information entered by the expert will be gathered into a (one or several) SALT reference, which is digitally signed by its author (the expert) and stored

into the SALT repository. Once the information is stored in the repository, it is available for other users to access it. Figure 14 describes this behavior.

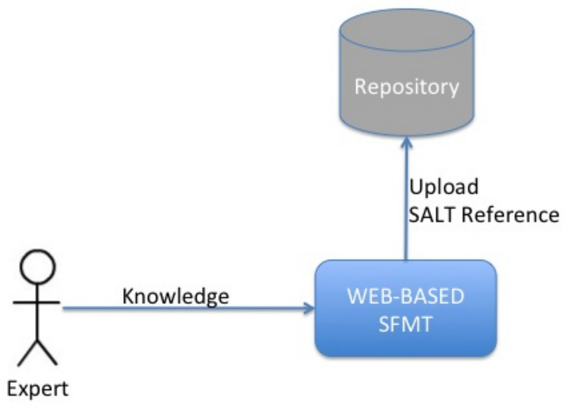


Figure 14. SALT Reference creation

Update of a SALT reference:

- Requirements: the SALT reference to be updated must be stored in the SALT repository. An expert user (in socio-contextual, ethical, legal or technological area) with the proper permissions (must be a valid author) provides the knowledge required to update the SALT reference.
- Behavior: the SFMT downloads the original SALT reference, which is updated with the new information coming from an expert (including a new digital signature). Then, the SFMT uploads the new version of the reference to the repository, but the old one also remains there, it is not erased. The deletion of the old reference will take place when its expiration date has been reached. Figure 15 describes this behavior.

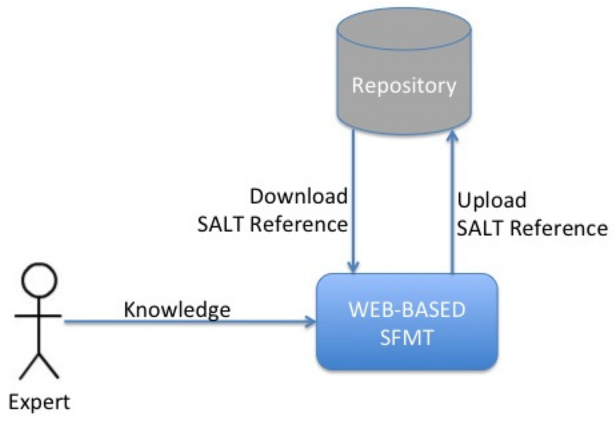


Figure 15. SALT Reference update

Search SALT references:

- Requirements: the user who performs the search must have the appropriate permission to do it, and of course he has to provide the search criteria.

- Behavior: the SFMT returns a list of SALT References (which have been retrieved from the SALT repository according to the search criteria) sorted by matching. Additionally, the tool may also provide the possibility of sorting the list by categories. Figure 16 describes this behavior.

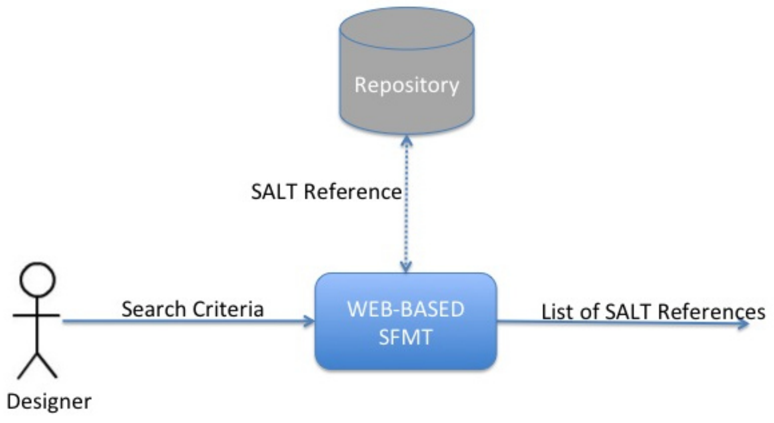


Figure 16. Search SALT References

3.2.2.1.2 Assisting the surveillance system design

Creation of the system design:

- Requirements: there must be at least one SALT Reference (in the repository) susceptible of being applied to the SUD. The system designer must have the necessary knowledge (system requirements, specifications, environment conditions, etc.) regarding the surveillance system.
- Behavior: the tool assists the designer in the task of creating a surveillance system design that matches the privacy and accountability concerns from the SALT References. However, designers are responsible of the design decisions, since they can dismiss the recommendations provided by the SALT References. Figure 17 describes this behavior.

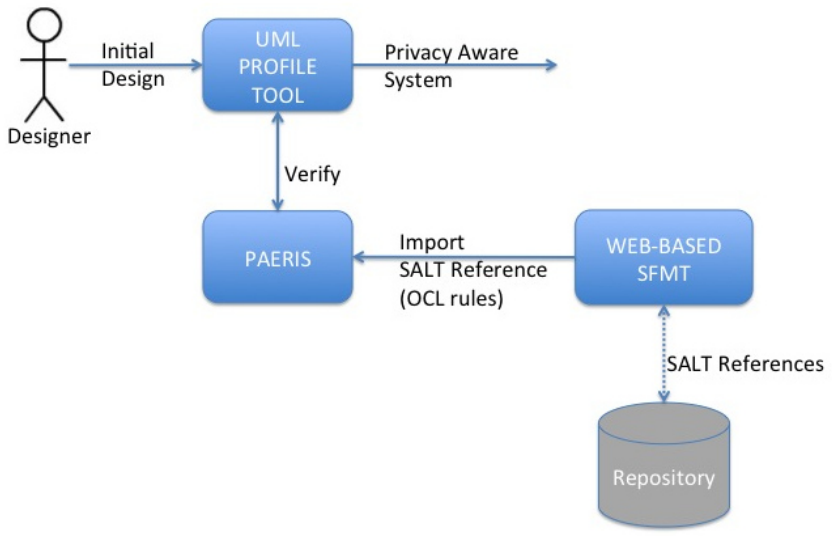


Figure 17. Creation and validation of a system design

3.2.2.1.3 Validating privacy concerns

Checking that privacy concerns have been properly applied:

- Requirements: there must be at least one SALT Reference (in the repository) whose concerns have been applied to the SUD. There must also be a surveillance system design to be checked (finished or still in progress), provided in the form of an UML model.
- Behavior: the PAERIS tool checks whether the privacy concerns from the SALT References have been properly implemented into the system design or not (this is an automated process thanks to the OCL rules within the SALT References). In case the validation does not succeed, the tool reports the errors corresponding to those concerns that have not been fulfilled. Figure 17 describes this behavior.

3.2.2.2 Structured Textual Repository Interface

The SALT Repository contains various contents which can be accessible through the web-based interface. This kind of interface can be used by technical stakeholder but it is too abstract for other experts. For this reason, the project has decided to investigate other ways to browse the repository.

Already presented in the deliverable D3.1, the first tentative has consisted in converting the SALT repository into a wiki similar to *wikipedia* for example. For this reason, the abstract architecture has to be instantiated as presented in Figure 18.

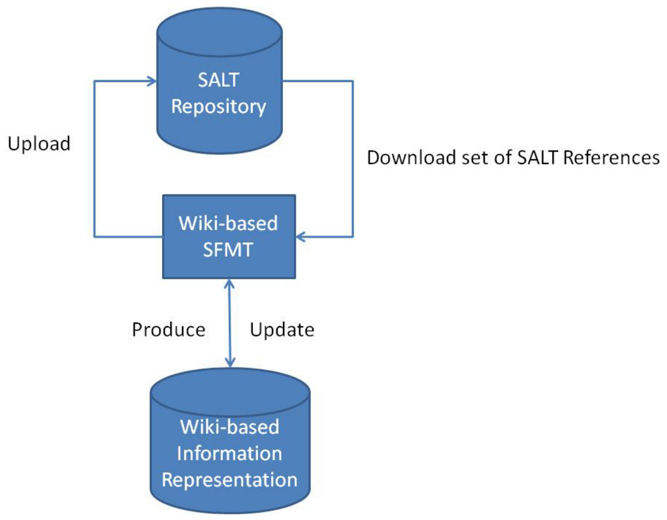

SALT Repository
Upload
Download set of SALT References
Wiki-based SFMT
Produce
Update
Wiki-based Information Representation

Figure 18. Instantiation of the Abstract Architecture into a Wiki-based Representation

In this version of the SFMT, the tool is screening the repository and then generates wiki pages. For this functionality, it is necessary to transform the models stored in the repository into text with the wiki tags. Model to text transformation is a technology which is well-known in model driven engineering. In the context of D3.1, some tests have been made with Acceleo (i.e., a plug-in for the Eclipse software development platform). Then, for producing the wiki, a bot is used in order to create the pages. For enhancing the performances, it is possible to generate only the pages with new or updated content. If an expert made a modification in the wiki, it should be interesting to update the model stored in the repository. This functionality is tricky to implement since the literature lacks of suggestion for carrying out this function. Indeed, "Text to Model" generation is widely used for reverse engineering in the context of source code. Applying this kind of technique to free text pages is not possible. However, if the structure of a wiki page is fixed by PARIS, a text analysis with model generation should be possible. Some investigations have to be performed.

3.2.2.3 User Dependent SALT Editor

A standalone dedicated SFMT is also planed in order to facilitate the usage by the experts. In order to present the design of this instance, some examples will be provided along the section.

Example of viewpoint-dependent content browsing

As explained in Section 3.1.1, it is necessary to filter the content of the repository. Indeed, each expert wants to address specific concerns. They do not need to visualize the full content. Figure 19 and Figure 20 provide an example regarding how the viewpoint can be implemented. In order to browse the repository, it seems that tree-based visualization is a good approach. Indeed, it is very easy to navigate for every user and to understand the purpose of a content. Of course, in order to have more information, it is possible to display a description when a node is selected.

In the left of Figure 19, the content of the repository is shown. On the right of Figure 19, we have defined some keywords in order to select only a part of the content. Figure 20 gives an example of the content shown when the filters are activated.

In Figure 19, a possible content is presented. The "SALT Reference" represents some knowledge coming from the literature such as laws for legal references or surveillance technologies used in current systems. This figure has been presented in the deliverable D2.3. All of this type of content is used as reference in terms of definition allowing people to have a common understanding. The "SALT Process" corresponds to all the knowledge directly used for helping designers to specify their system. For instance, some templates are given in order to make privacy harm analysis or questionnaires in order to help designer to take the good decisions. This common knowledge can be refined or customized for specific systems. The specific knowledge is stored in "My SALT Process".

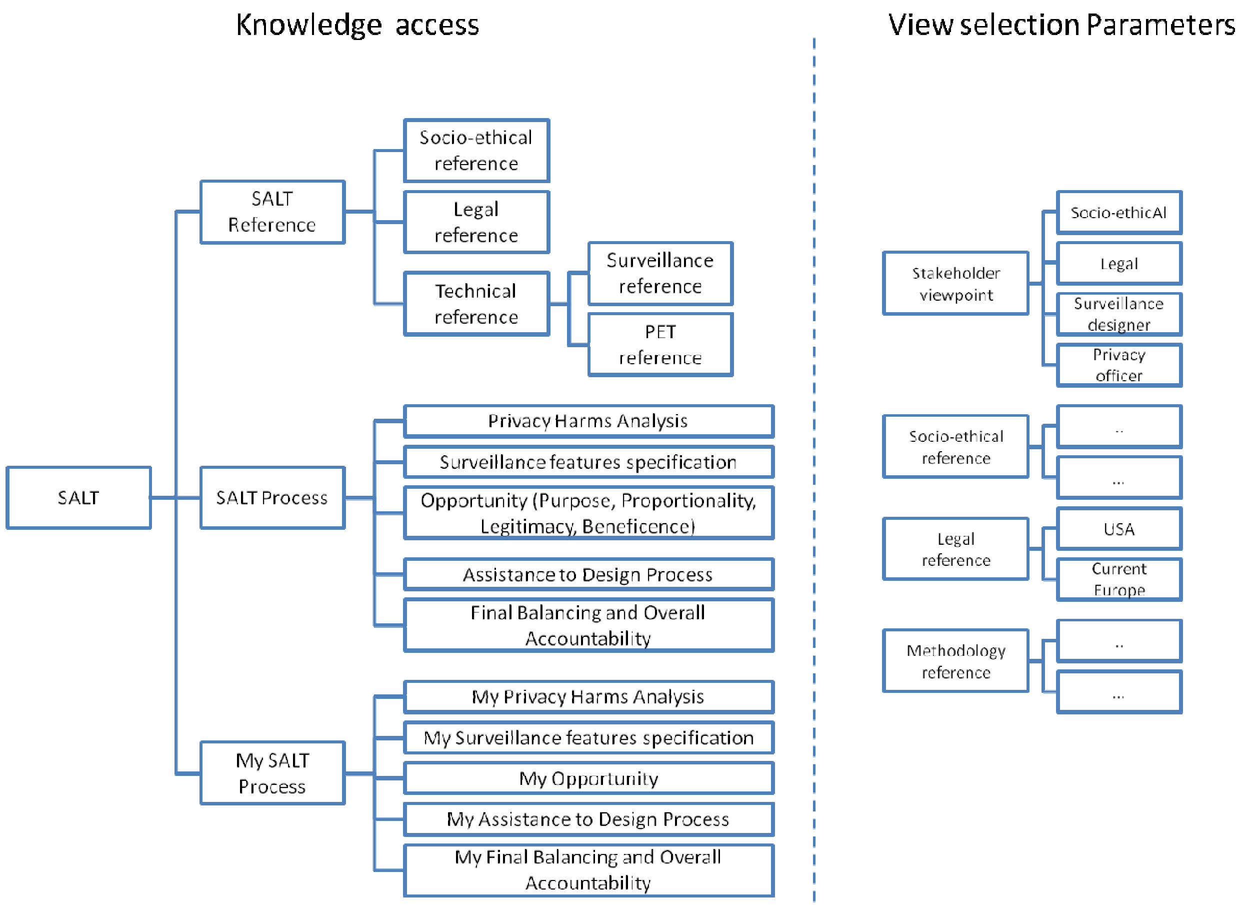


Figure 19 Example of a SALT Repository Viewer

In order to implement this functionality, all models stored in the repository have to be tagged by some values in order to customize the visualisation. It is a constraint on the schema of the model.

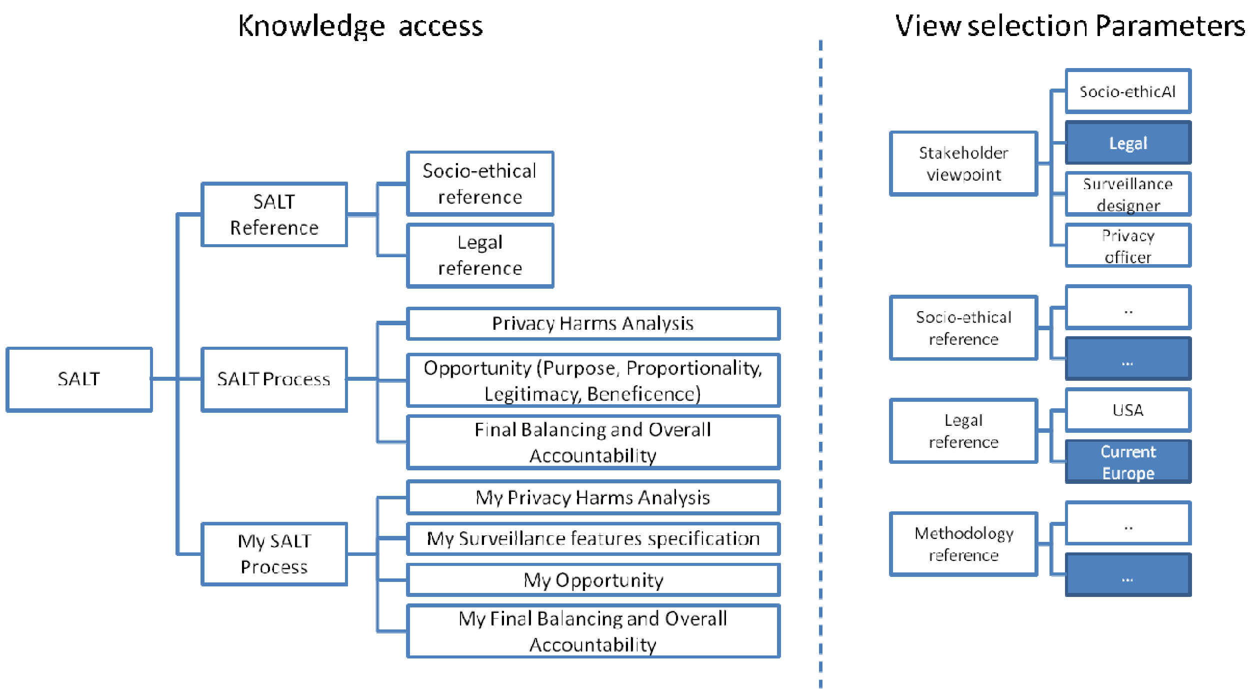


Figure 20. Working Example for Setting up the Tool

Example of the taxonomy centric SFMT

One possible content stored in the repository will be some taxonomies like surveillance technologies, privacy harms, PETS. In the SALT repository, it will be stored in the "SALT Reference" part. The instance of the SFMT will be able to create or to edit a taxonomy as depicted in Figure 21.

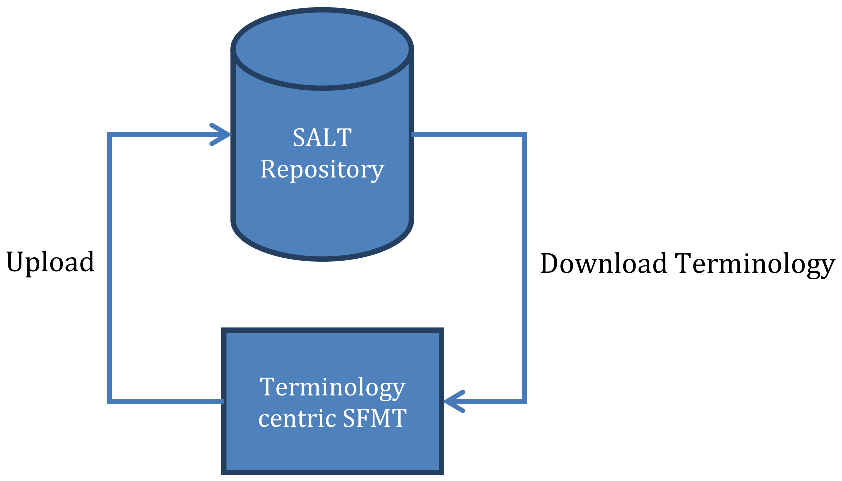
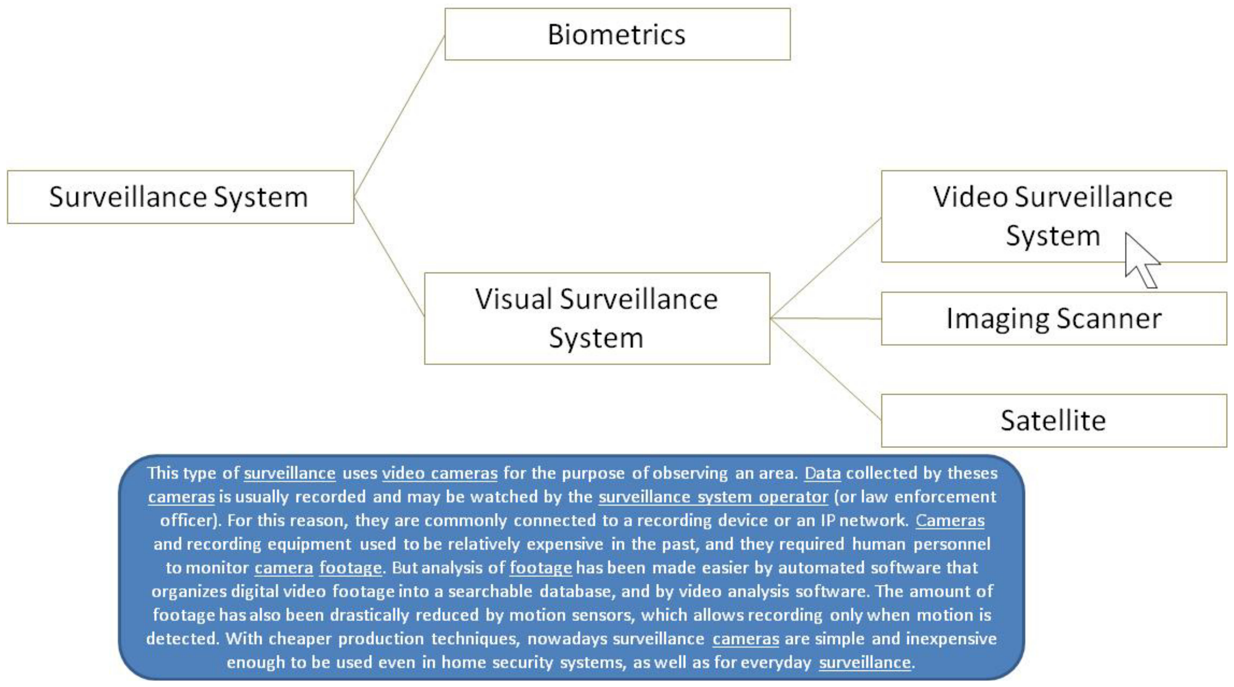


Figure 21. Instantiation of the Abstract Architecture for Managing Taxonomies

The editor shows the taxonomy through a tree in order to highlight the relations between the concepts (see Figure 22). When the expert selects a concept, the definition appears in a tooltip. In the description, the editor highlights other concepts already defined by underlining them.



This type of surveillance uses video cameras for the purpose of observing an area. Data collected by theses cameras is usually recorded and may be watched by the surveillance system operator (or law enforcement officer). For this reason, they are commonly connected to a recording device or an IP network. Cameras and recording equipment used to be relatively expensive in the past, and they required human personnel to monitor camera footage. But analysis of footage has been made easier by automated software that organizes digital video footage into a searchable database, and by video analysis software. The amount of footage has also been drastically reduced by motion sensors, which allows recording only when motion is detected. With cheaper production techniques, nowadays surveillance cameras are simple and inexpensive enough to be used even in home security systems, as well as for everyday surveillance.

Figure 22 Example of a Possible Editor

As you can see in Figure 22, a taxonomy is composed of the terms with their definition. The links corresponds to references in the definitions. Thanks to these definitions, it is possible to check the consistency of the different visions of the experts.

The editor allows usual edition features like renaming, updating a description, adding new concepts, removing concepts, and so on.

Example of the questionnaire centric SFMT

Similar to the taxonomy editor, it is necessary to create and edit some questionnaires. As depicted in Figure 23, the SFMT will interact with the repository in order to search, load and upload some questionnaires.

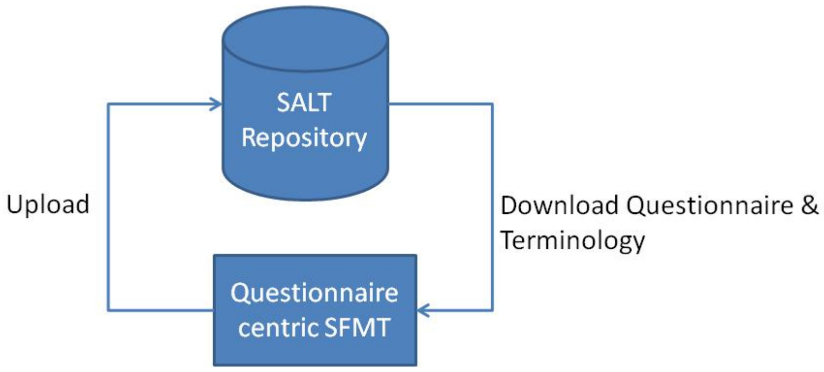
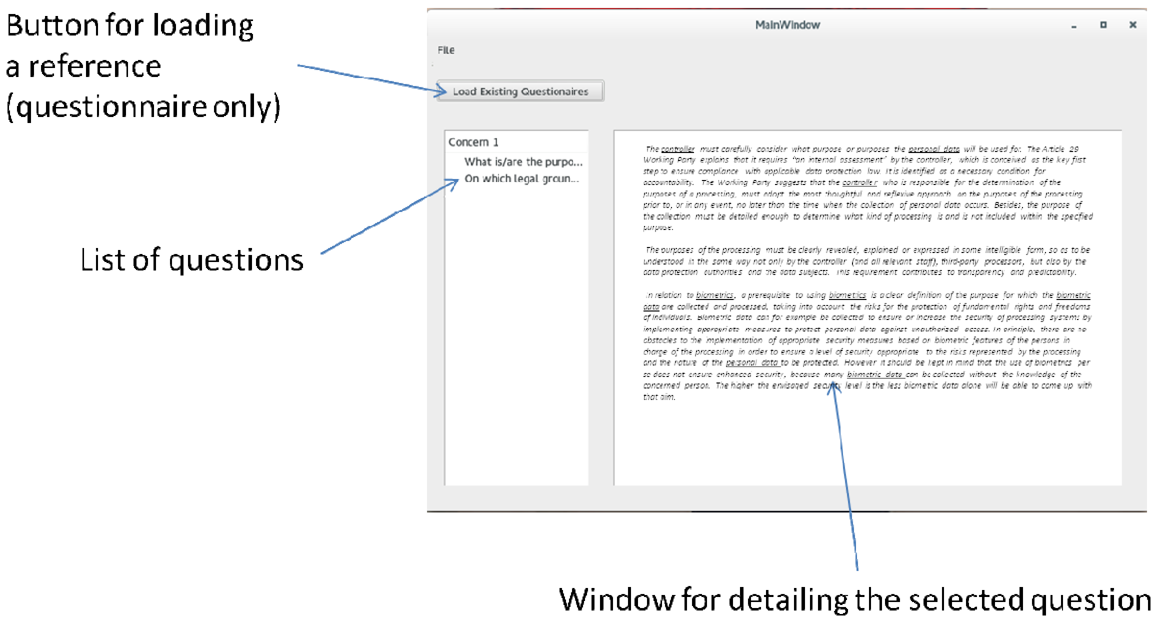

SALT Repository
Upload
Download Questionnaire & Terminology
Questionnaire centric SFMT

Figure 23. Instantiation of the Abstract Architecture for Managing Questionnaires

Some sketches have been already developed in order to discuss about the functionalities. Two examples are given in Figure 24 and Figure 25.


Button for loading a reference (questionnaire only)
List of questions
Window for detailing the selected question
MainWindow
File
Load Existing Questionaires
Concern 1
What is/are the purpo...
On which legal groun...

The controller must carefully consider what purpose or purposes the personal data will be used for. The Article 29 Working Party explains that it requires "an internal assessment" by the controller, which is conceived as the key fist step to ensure compliance with applicable data protection law. It is identified as a necessary condition for accountability. The Working Party suggests that the controller who is responsible for the determination of the purposes of a processing, must adopt the most thoughtful and reflexive approach on the purposes of the processing prior to, or in any event, no later than the time when the collection of personal data occurs. Besides, the purpose of the collection must be detailed enough to determine what kind of processing is and is not included within the specfied purpose.

The purposes of the processing must be clearly revealed, explained or expressed in some intelligible form, so as to be understood in the same way not only by the controller (and all relevant staff), third-party processors, but also by the data protection authorities and the data subjects. This requirement contributes to transparency and predictability.

In relation to biometrics, a prerequisite to using biometrics is a clear definition of the purpose for which the biometric data are collected and processed, taking into account the risks for the protection of fundamental rights and freedoms of individuals. Biometric data can for example be collected to ensure or increase the security of processing systems by implementing appropriate measures to protect personal data against unauthorized access. In principle, there are no obstacles to the implementation of appropriate security measures based on biometric features of the persons in charge of the processing in order to ensure a level of security appropriate to the risks represented by the processing and the nature of the personal data to be protected. However it should be kept in mind that the use of biometrics per se does not ensure enhanced security, because many biometric data can be collected without the knowledge of the concerned person. The higher the envisaged security level is the less biometric data alone will be able to come up with that aim.

Figure 24. Screenshot of a Possible Questionnaire Browser

On the left of Figure 24, the user can see the list of questions. When one question is selected, the browser provides some details about the question. Indeed, when an expert defines a questionnaire, a textual description is aggregated to the question in order to help the final user. Automatically, the SFMT is also searching if some keywords have been defined in the taxonomies. In this case, the word is highlighted in the text.

Figure 25 is an example of a questionnaire and uses a similar approach view as the browser.

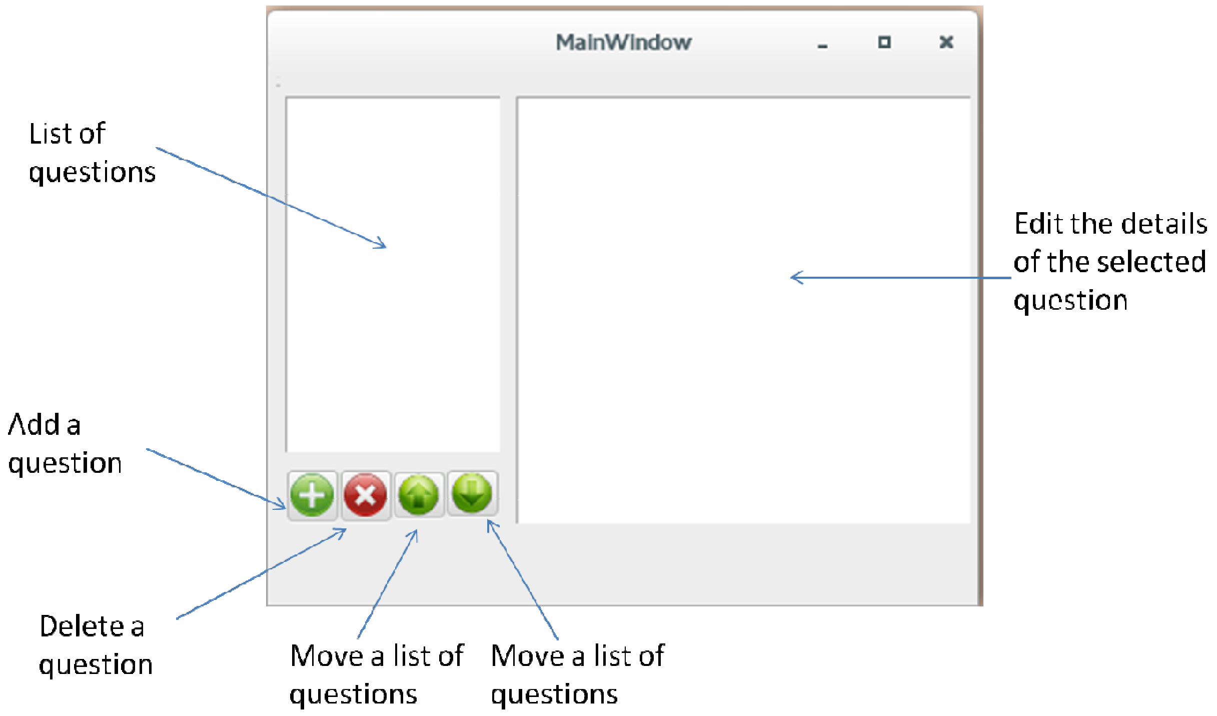


Figure 25.Example of a Possible Questionnaire Editor

The list of the questions is on the left of the editor. However, it seems important to add some user flexibility for managing the questions. Indeed, some questions are related. In order to provide this information, it is necessary to provide a way for defining the structure of a questionnaire. An example is presented in Figure 26. When there is a link between some questions, it means that the questions are sequential. Otherwise, the questions are completely independent and can be later answered separately.

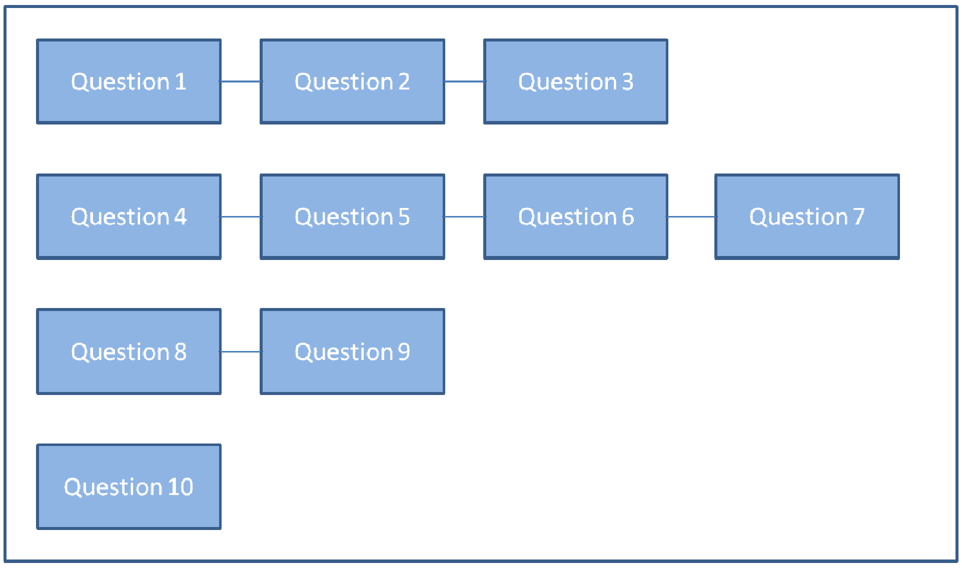


Figure 26. Definition of a Questionnaire Structure

Example of the relations between two SALT contents

In order to illustrate this functionality, we will reuse the content presented in the deliverable D2.2 [4] based on the types of privacy.

R. L. Finn, D. Wright and M. Friedewald in [11] have defined 7 types of privacy which can be threatened by some surveillance technologies. When the system designer specifies his system, he will need to check the privacy harms according to the selected technologies. Table 8 explains how the surveillance technologies interfere on the privacy.

		Technology				
		Whole body imaging scanners	RFID-enabled travel documents	Unmanned aircraft systems	Second-generation DNA sequencing	Human enhancement technologies
Type of privacy	Privacy of the person	X			X	X
	Privacy of behaviour and action	X	X	X	X	X
	Privacy of communication					X
	Privacy of data and image	X	X	X	X	X
	Privacy of thought and feelings					X
	Privacy of location and space		X	X	X	
	Privacy of association			X	X	

Table 8. Impact of Surveillance Technologies on Privacy

In order to store such information in the SALT repository, it is necessary to define two taxonomies (i.e., the surveillance technologies and the type of privacy) and the relationship between them.

In the operational lifecycle of the SALT repository, the SFMT needs to check the consistency between all inputs. Table 9 shows an update of the surveillance taxonomy with the addition of the second-generation biometrics. The SFMT will need to raise a warning in order to request an update from an expert.

		Technology					
		Whole body imaging scanners	RFID-enabled travel documents	Unmanned aircraft systems	Second-generation DNA sequencing	Human enhancement technologies	Second-generation biometrics
Type of privacy	Privacy of the person	X			X	X	?
	Privacy of behaviour and action	X	X	X	X	X	?
	Privacy of communication					X	?
	Privacy of data and image	X	X	X	X	X	?
	Privacy of thought and feelings					X	?
	Privacy of location and space		X	X	X		?
	Privacy of association			X	X		?

Table 9. Update of the Technology and Privacy Map

In summary, the process is the following:

1. Expert A creates a taxonomy of surveillance technologies.
2. Expert B creates a taxonomy of privacy types.
3. Expert C creates some relations between the surveillance technologies and privacy types.

   a. This information allows system designer what are the privacy risks triggered by the system and what should be the solutions in terms of PETs
4. Expert A identifies a new surveillance technology and updates the taxonomy in consequence.
5. The SALT Framework detects that the taxonomy is updated and checks the consistency with other inputs.
6. The framework raises a warning since an expert needs to fill out the relation between the new surveillance technology and the privacy types

For information, this kind of table is part of privacy risk analysis like the CNIL method [12]. Indeed, in such methods, it is necessary to identify the privacy risks of a system. Due to the usage of specific technologies, some privacy risks are more or less higher. Table 8 and Table 9 are just some examples which are not fully conformed to the reality. Indeed, in usual risk analysis method, the risk values are not “yes” or “not” but are weighted by an enumeration. Moreover, it should be interesting to customize the table. In the repository, the customization will be stored in “My SALT Process”. For example, some devices which do not normally affect the privacy person are stored as privacy person preserving. But, for instance, if the devices used by the company embed a chipset which is beneath someone’s skin, the table has to be updated in consequence.

Figure 27 explains the architecture in order to achieve this functionality. When a modification is done by an expert, a consistency checking is performed. The tool will trigger some alerts if new contributions are needed.

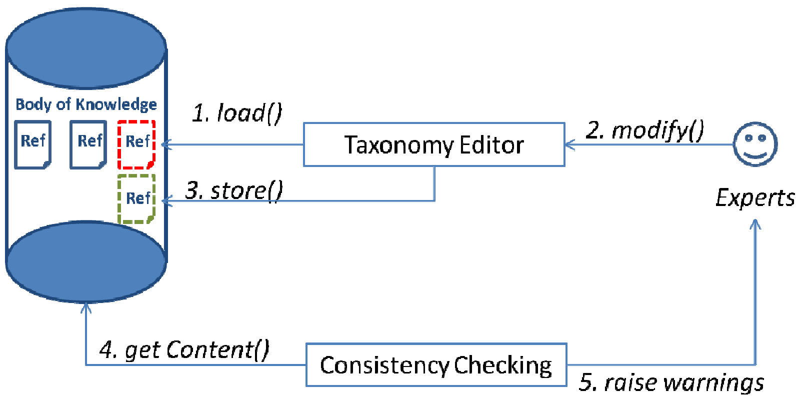


Figure 27. Architecture of the SFMT for Checking the Consistency

3.2.2.4 Running a Questionnaire

In this subsection, we present how the SFMT can be used by a system designer. Since a system designer is not always familiar with legal, social or ethical domains, some questionnaires for helping him will be stored in the repository. This use case presents how the designer can run a questionnaire in order to check the consistency with the laws or ethical perceptions. Figure 28 shows the interaction between the repository and the SFMT.

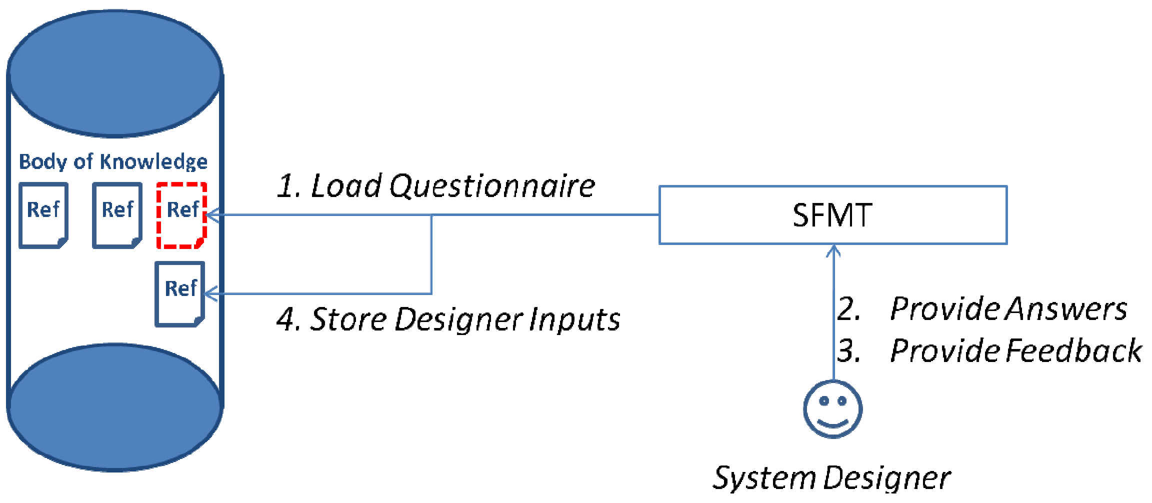


Figure 28. Instantiation of the SFMT for Running a Questionnaire

The system designer will have to answer the questionnaire. He can also add some comments or feedback in order to enhance the questionnaires or to help other designers. The SMFT will deal with the load of the questionnaire and the storage of answers and feedback. Figure 29 presents a possible screenshot of the tool. When the designer has finished answering to a question, a tick is added in the question icon. As the screenshot shows, it is not necessary to fill out the questionnaire in the numerical order. It is possible to start in the middle of the questionnaire since the previous questions are independent. Moreover, the system designer can stop answering and resume later. All the questions already done are stored in the repository.

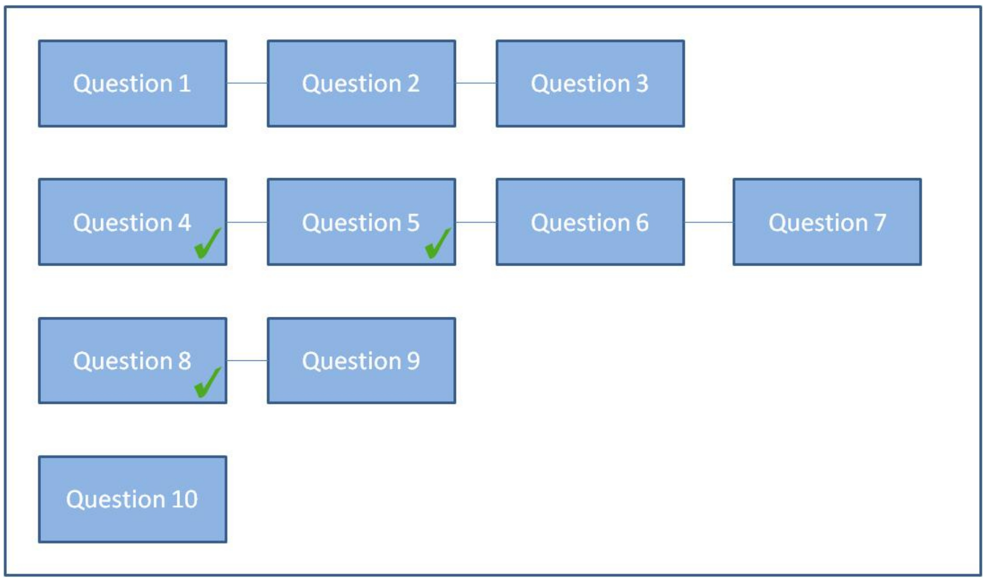


Figure 29 Questionnaire Map

3.2.3 Validation Criteria w.r.t. the Scenarios

The toolset developed in the Work Package 3 will be implemented in the task T3.4. In order to check the different functionalities of the tools, content for the SALT framework is needed. For this reason, collaboration between all work packages is required in order to (i) have a realistic content and (ii) validate through the demonstrators that the tools meet the requirements.

As usual practices, test scenarios will be run in order to check all functions. Scenarios defined in the deliverable D2.2 [4] will be used in order to check the requirement compliance.

4 Conclusion

The SALT Framework Management Tool (SFMT) is linked to some other tools/applications, which all together form the whole toolset involved within the SALT general process. Therefore, it is important to clarify the inputs and functionalities of these tools in order to understand how they work and how they interact with each other. With this goal in mind, we have described the technology, concepts and requirements used for developing the SFMT, as well as the role of the SFMT in the SALT general process. Thanks to this we get a clear view of the type of users that will handle the tool, and what privileges are granted to each one.

The document also provides details regarding the modelling features of the toolset, which have been totally covered with the use of UML. Moreover, even though the SFMT is not fully implemented at this step of the project, we provide a list of the foreseen reasoning features that we would like the final version to have: privacy informer, documentation validator and generator and assisted compliance checker among others.

Regarding the specification of the SFMT, the first thing to consider are the different scenarios where the tool is going to be used. We describe these scenarios according to the point of view of the different types of users, such as knowledge experts and surveillance systems designers. With all this information clear we can move to a description of the toolset architecture, where we show all the elements taking part in it: the SFMT, the SALT repository, the UML profile, the PAERIS tool (automatic OCL rules validator) and the wiki-based repository. We discuss the functionality of each element and provide the details.

We have decided to separate the SFMT into two applications, which complement each other. Both approaches manage the information stored within the SALT repository, although this decision has been made following the needs of the different users: the wiki-based representation is mainly focused to socio-contextual, ethical and legal users, whereas the web-based approach should be more practical for technological users. There is also a questionnaire based functionality for surveillance systems developers within the wiki-based tool. Each questionnaire is composed of a set of questions and systems developers will run them in order to check if the SUD (System Under development) properly addresses the privacy concerns. A given questionnaire can be run at the beginning of a surveillance project and resumed later.

The current implementation state of the SFMT includes a first version of the web-based SFMT, which initially is integrated within the implementation of the SALT repository. Both have been developed with PHP, including functionalities not only for storing SALT references, but also for searching and retrieving them. Some other functionalities, such as rating SALT references or digital signature capabilities for trusting capabilities are still under development. Moreover, the interface currently provided is not definitive. It is functional, thus it allows for working with the tool while helping with the development itself, but it still needs more refinement for the sake of clarity and ease of use, specially regarding non technological users (socio-contextual, ethical and legal categories).

The wiki-based SFMT is also under development. A first wiki version has currently been developed, where SALT references can be stored and displayed to users in more friendly, text-based way. The "point of views" approach, depending on the type of user who handles the tool, will be developed for the next version of the tool, together with the questionnaire-based functionalities.

Regarding the rest of the toolset, the UML profile is also under development. Modelling artefacts covering biometric and video-surveillance systems have been developed, although the user interface still needs more refinement. The first attempt has been develop using the MagicDraw modelling tool. The automatic validator (the PAERIS tool) in charge of checking OCL rules against a system design (an UML model) is planned to be developed as a plugin also for MagicDraw.

5 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] A. Cockburn, "Writing effective use cases", Addison-Wesley, 2001.
[4] PARIS FP7 Project Deliverable D2.2 "Structure and Dynamics of SALT Frameworks".
[5] PARIS FP7 Project Deliverable D2.1 "Contexts and concepts for SALT Frameworks".
[6] PARIS FP7 Project Deliverable D4.2 " SALT Compliant Processes Definition".
[7] PARIS FP7 Project Deliverable D4.3 " SALT Compliant Processes Guidelines for Use Cases".
[8] PARIS FP7 Project Deliverable D3.1 " SALT Framework Management Tool Requirements".
[9] UML http://www.uml.org/
[10] OCL http://www.omg.org/spec/OCL/
[11] R. L. Finn, D. Wright and M. Friedewald, "Seven Types of Privacy," in *European Data Protection: Coming of Age*, S. Gutwirth (ed), Dordrecht: Springer, 2013.
[12] CNIL, “Methodology for Privacy Risk Management. 2012. http://www.cnil.fr/fileadmin/documents/en/CNIL-ManagingPrivacyRisks-Methodology.pdf