



PrivAcy pReserving Infrastructure for Surveillance

Deliverable D3.3 SALT Framework Management Tool

Project: PARIS
Project Number: SEC-312504
Deliverable: D3.3
Title: SALT Framework Management Tool
Version: v1.0
Date: 31/12/2014
Confidentiality: Public
Authors: Christophe Jouvray (Trialog)
Eztibaliz Arroz Fernandez (Trialog)
Francisco Jaime (UMA)



Funded by the European Union's
Seventh Framework Programme

Table of Contents

DOCUMENT HISTORY	3
EXECUTIVE SUMMARY	3
LIST OF FIGURES.....	4
LIST OF TABLES.....	4
1 INTRODUCTION	6
1.1 Scope.....	6
1.2 Structure of the Document.....	7
1.3 Generic description of SALT contents.....	7
2 SALT REPOSITORY VIEWER	9
3 QUESTIONNAIRE TOOL IMPLEMENTATION	15
3.1 Data base model	15
3.1.1 User tables.....	15
3.1.2 Questionnaire tables.....	17
3.2 Overview of the Questionnaire Tool	19
4 CONCLUSION	24

Document History

Version	Status	Date
v0.1	Document structure	01.12.2014
v0.2	Update of Section 1 & 3	12.12.2014
v0.3	Update of Section 2	19.12.2014
v0.4	Consolidation of all sections	19.12.2014
V0.5	Integration of reviewers comments	24.12.2014
V1.0	Finalization of the document	31.12.2014

Approval		
	Name	Date
Prepared	Christophe Jouvray	31.12.2014
Reviewed	Thales	31.12.2014
Authorised	Antonio Kung	31.12.2014
Circulation		
Recipient	Date of submission	
Project partners	31.12.2014	
European Commission	31.12.2014	

Executive Summary

The solution promoted by the PARIS project is called SALT (Socio ethicAI Legal Technological) framework. It is made of pieces of information spreading from multi-disciplinary domains. The atomic SALT content is called a SALT reference. In order to manage this framework, a software tool featuring user interaction interfaces is developed in the work package 3 of the project. A SALT reference is stored in a specific repository. The project partners have decided to implement a set of tools for managing the references according to the role and the usage. This document presents the implementation of the tools developed in this context.

List of Figures

Figure 1 Basic scheme of the main system.	6
Figure 2 SFMT in the Process	7
Figure 6: Internal contents of the SALT framework per system type	8
Figure 3. Login interface to the SALT repository	9
Figure 4. Searching the SALT repository	10
Figure 5. Creation of a new SALT reference	11
Figure 6. Adding concerns to a SALT reference	11
Figure 7. Adding OCL rules to a given concern	12
Figure 8. Interface to access questionnaires	12
Figure 9. Window showing metadata exclusive for a given repository	13
Figure 10. About tab	13
Figure 11. Logout window	14
Figure 12 Register Window	20
Figure 13 Login Window	20
Figure 14 List of questionnaire	21
Figure 15 Non-answered Questions	21
Figure 16 Question Selected	22
Figure 17 Complex Question	22
Figure 18 Answered Questions	23

List of Tables

None

Abbreviations and Definitions

Abbreviation	Definition
OCL	Object Constraint Language
PAERIS	PrivAcy-by-design EngineeRING aSsistant
SALT	Socio-contextual, ethicAl, Legal, Technological
SFMT	SALT Framework Management Tool
UML	Unified Modelling Language
WP	Work Package

1 Introduction

This version of the deliverable D3.3 entitled SALT Framework Management Tool is a first version. This document corresponds to a textual description of the work done for D3.3, consisting of the development of the needed tools for implementing the SALT framework

1.1 Scope

The SALT framework management system is called SFMT (SALT Framework Management Tool), which assist users (surveillance system designers) to access the SALT Framework.

SALT Framework stores knowledge from different experts in legal, ethical, technological and social scopes. This information concerns about privacy and accountability aspects involved in the design of surveillance system (video surveillance or biometric system) designs.

SALT Framework main functions are:

- 1) Add & Update knowledge
- 2) Display & use knowledge

In the figure 1, it is resumed the main parts of the system:

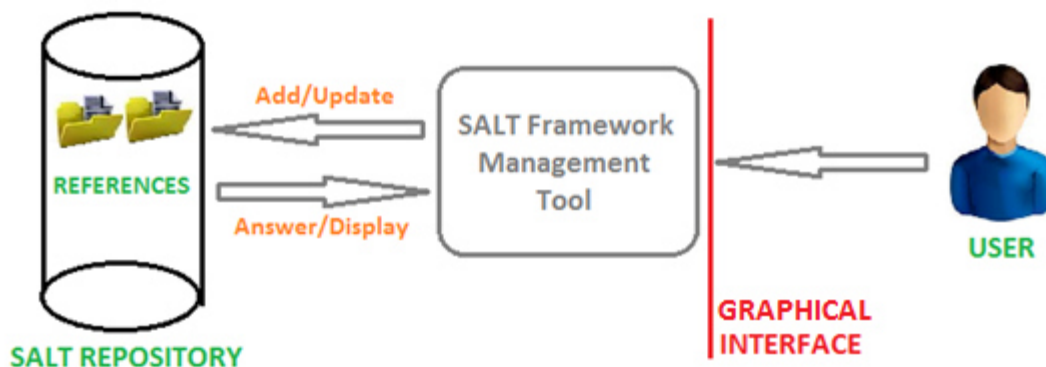


Figure 1 Basic scheme of the main system.

The SFMT contributes to the elaboration for PIAs (Privacy Impact Assessment) and during a privacy-by-design process as explained in Figure 2.

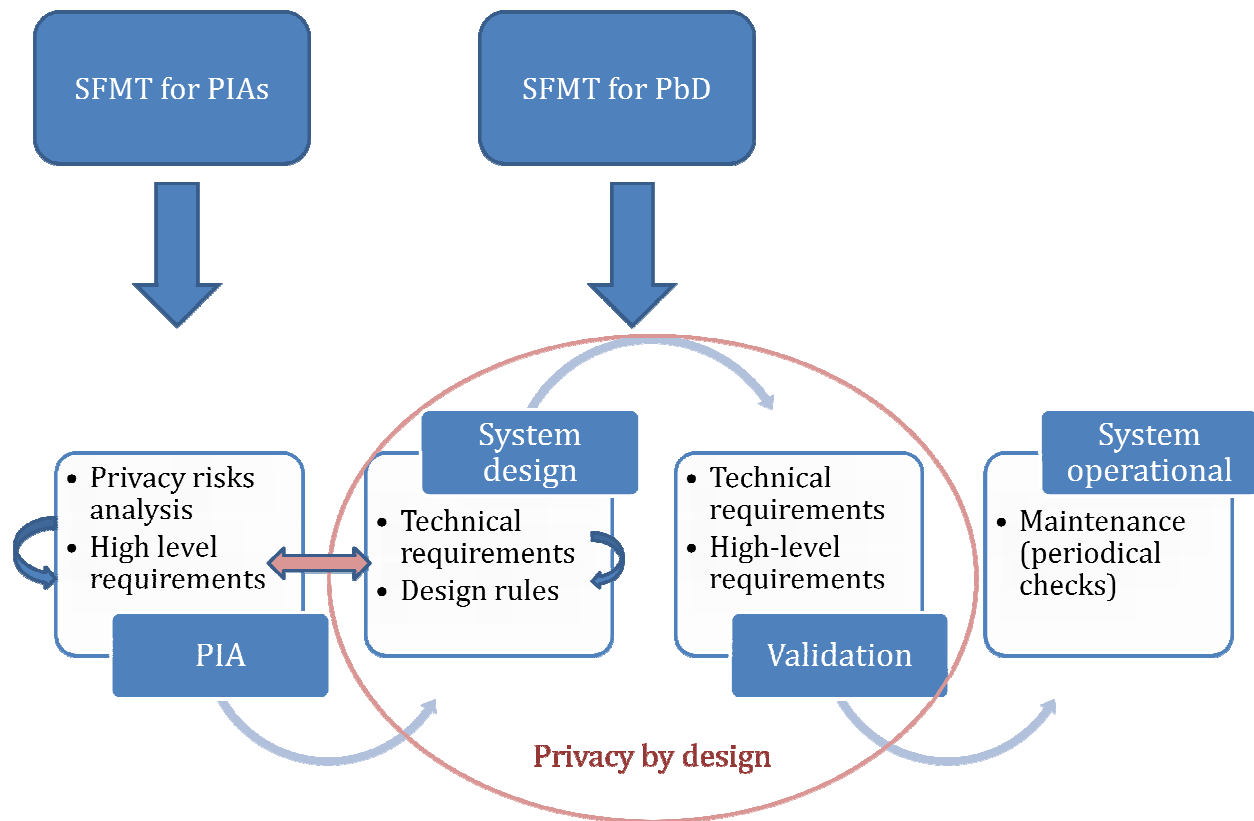


Figure 2 SFMT in the Process

1.2 Structure of the Document

Two tools have been developed in the context of the work-package WP3. The first one, detailed in Section 2, is mainly designed for the privacy-by-design usage. Indeed, Section 2 presents a web interface dedicated to technical experts for:

- Adding or updating new SALT references including OCL rules for design checking.
- Browsing SALT references

Finally, Section 3 describes the questionnaire tool used for developing PIAs.

1.3 Generic description of SALT contents

The atomic content of the SALT framework dedicated to the storage of the main information pieces is the SALT reference. Nevertheless, the SALT framework also contains 2 other types of entities:

- Systems profiles (UML artifacts and systems designs) dedicated to a given system type,
- Risks analysis grid template dedicated to a system type.

The global coherence of the SALT contents is explained below:

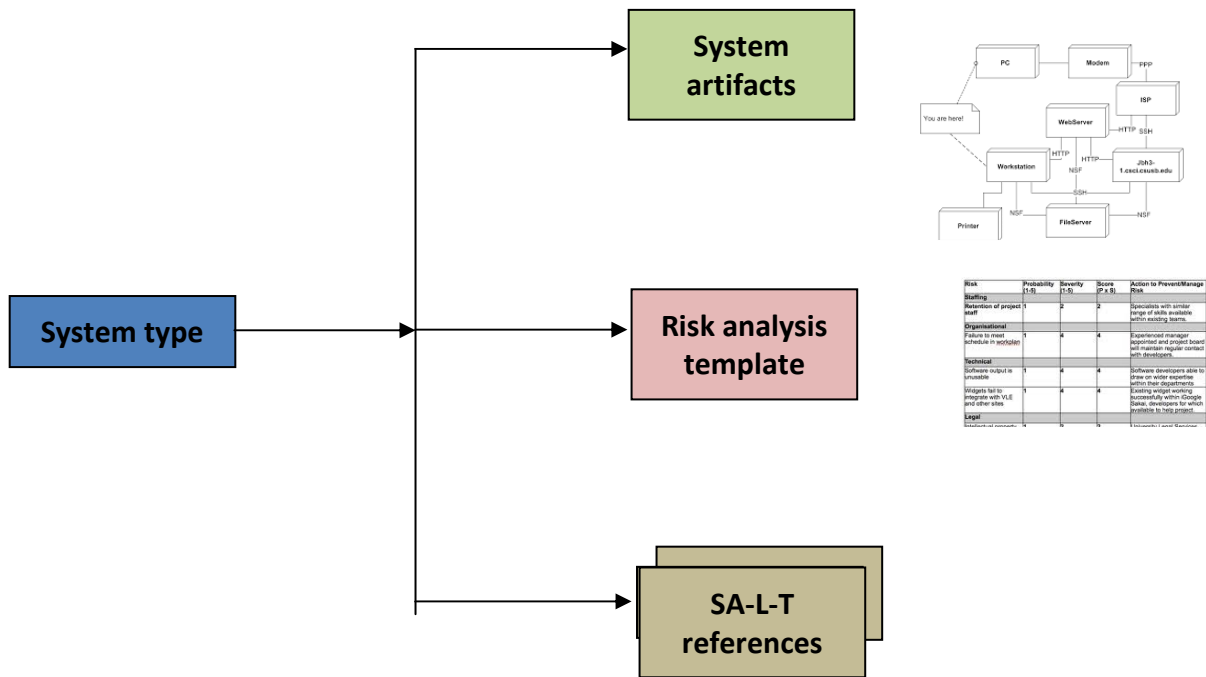


Figure 3: Internal contents of the SALT framework per system type

Examples of systems types are: video-surveillance system, biometric system based on fingerprint recognition.

The OCL rules embedded in some (if any) SALT references are applicable to the system design template (one or several) embedded within the system artifacts related to the system type.

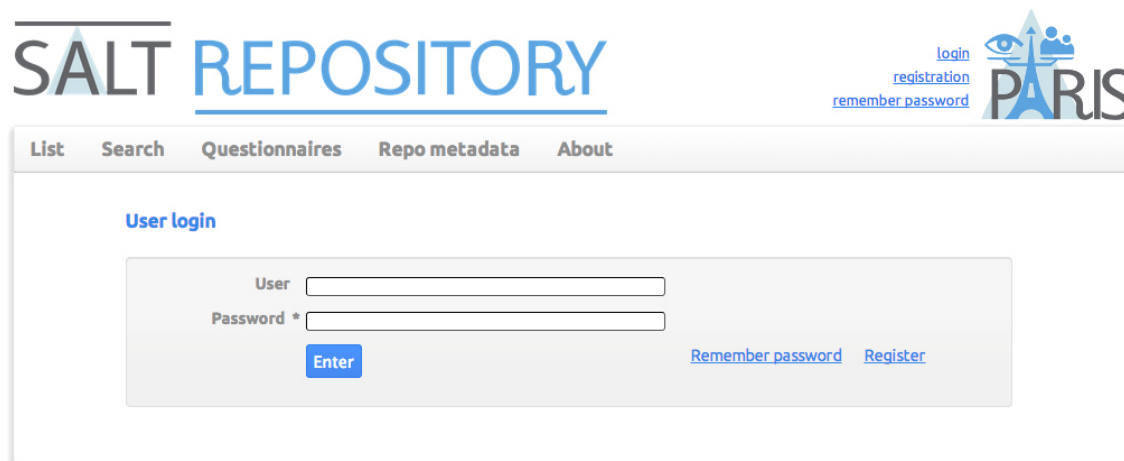
The risk analysis template is an empty matrix, proposing to the SALT users a vertical list of functions that may be embedded in a system entering in the category of the “system type”, and horizontally a list of risks that may weight on the privacy and accountability of the system.

2 SALT Repository Viewer

The SALT repository is the physical storage of the SALT framework, i. e. the place holding the collection of SALT references that contain privacy and accountability concerns related to surveillance systems. Therefore, according to the expected functionality, the SALT repository must provide ways of accessing its information for the creation, update, deletion and consultation.

Due to the interaction of the SALT repository with the different phases of the SALT process, the PARIS project plans to develop different tools to connect to the repository: mainly a web interface for the management (creation/update) of SALT references, and a plugin for the UML profile, which is another tool aimed to help system designers during the development of a surveillance system design. The development of another tool is planned for presenting the information within SALT references in a wiki style, thus making it more accessible to non-technological users.

The first step when going into the SALT repository is to login (or register in case the user still does not have an account). This is very important, since thanks to it any new SALT reference will be linked to a corresponding author. Figure 4 shows the login interface to the SALT repository through the web interface.



The screenshot shows the 'SALT REPOSITORY' web interface. At the top left, the title 'SALT REPOSITORY' is displayed in blue. To the right, there are links for 'login', 'registration', and 'remember password', along with the 'PARIS' logo. Below the title, a navigation menu contains 'List', 'Search', 'Questionnaires', 'Repo metadata', and 'About'. The main content area is titled 'User login' and features a form with two input fields: 'User' and 'Password *'. A blue 'Enter' button is positioned below the password field. To the right of the form, there are links for 'Remember password' and 'Register'.

Figure 4. Login interface to the SALT repository

If the user does not log into the repository he will still be allowed for searching and browsing among the existing SALT references, although he will not be able to create a new one or make any other modification to the repository.

Once the user is logged into the repository he has a complete access to a set of functionalities. Figure 5 shows an example of search. As it can be seen, the interface provides a text field where the user can enter any keyword. The repository will then instantly start to look for it within the stored SALT references, providing a list with all the coincidences. The button named "List" at the top left side of the interface will just provide a long list with the SALT references currently stored into the repository. Browsing through this list may not be very satisfactory for a big enough amount of references, and hence the importance of the *search* function.

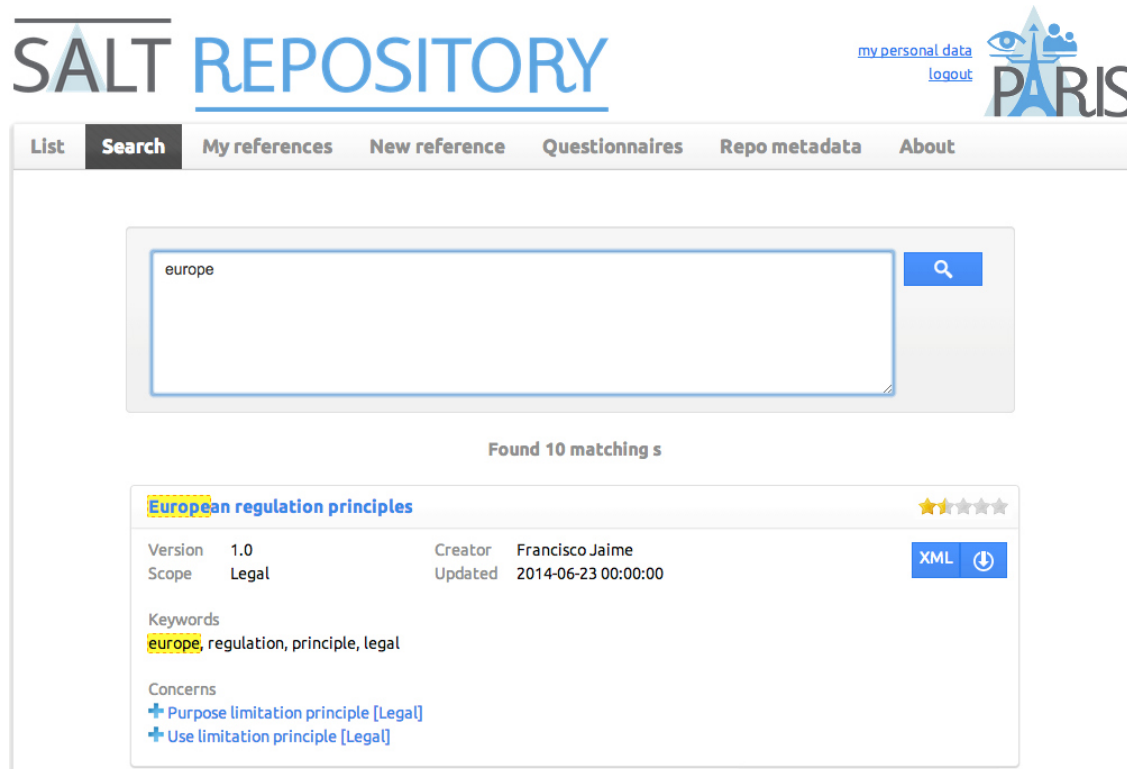


Figure 5. Searching the SALT repository

Next to the “Search” button it is the “My references” button. By clicking on it the user will get another list of SALT references following the same style as the one provided after a search is performed. However, in this case the list will only contain those SALT references that were created by the current user.

The creation of a new SALT reference is achieved through the interface shown when clicking on the “New reference” button, which is depicted in Figure 6. Here the user can introduce the general information for the SALT reference he is about to create, such as the reference’s name, version, scope (social, ethical, legal, technological) and keywords (for search purposes). The name of the reference’s author is set by the tool, taken from the author’s login information.

With the general information already entered, the user can start to add privacy and/or accountability concerns to the SALT reference. There is no restriction in the number of concerns that can be added to a single reference, although a reference with zero concerns would be useless. Figure 7 shows how the web interface manages the addition of concerns: anytime the “New” button is clicked a new concern is added to the reference. The interface facilitates the introduction of information regarding the current concern: the concern title, the category it belongs to, a textual description and the keywords that will later be used to locate it with a search operation.



Figure 6. Creation of a new SALT reference

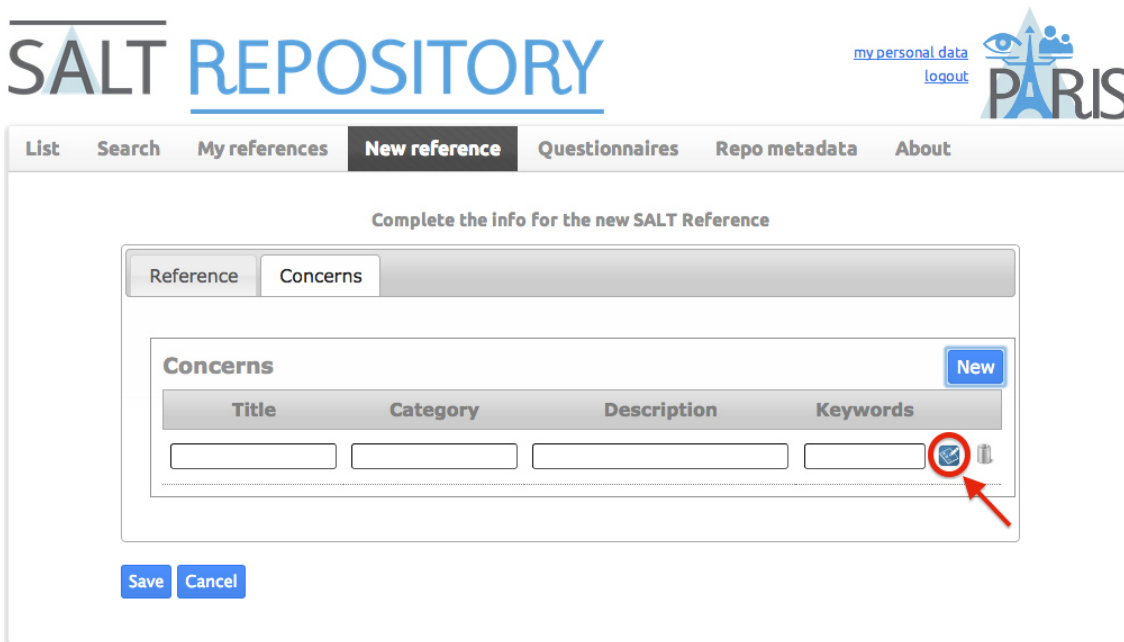


Figure 7. Adding concerns to a SALT reference

Having a look at Figure 7 we notice there is a blue button that allows for the addition of OCL rules for a concern (it is marked in the figure with a red circle and pointed by a red arrow). OCL rules are not mandatory, and hence the user is not forced to introduce them. Nevertheless, OCL rules are the mechanism used for the automated validation of the correct application of privacy/accountability concerns into the surveillance system design. This means that if OCL rules are not provided, the design tool (the UML profile previously mentioned) will not check the corresponding concerns.

Figure 8 shows the interface used for the introduction of OCL rules: a description of the rules (what kind of checking it performs), a message to be displayed in case the OCL rule is not fulfilled (its corresponding concern has not properly been taken into account), the level of

criticality for that rule not to be fulfilled (error, warning, info), and the OCL code itself. By clicking on the “New” button the user can add as many OCL rules as necessary for the corresponding concern to be validated.

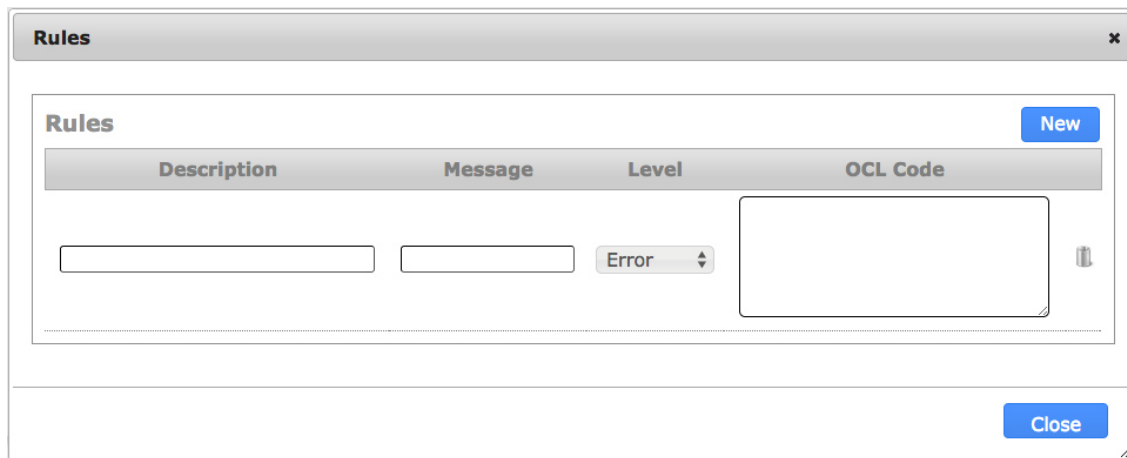


Figure 8. Adding OCL rules to a given concern

The next functionality relates to the questionnaires. By clicking on the “Questionnaires” button the user will access a list with all the questionnaires currently available (we expect to have just a few of them, that is the reason why we do not think a search functionality is necessary here). These questionnaires are intended to be used by systems’ stakeholders and they will help to assess the impact on privacy and accountability of the system to be. Figure 9 shows how the web interface for the SALT repository allows for downloading those questionnaires they user may find useful for the current surveillance system.

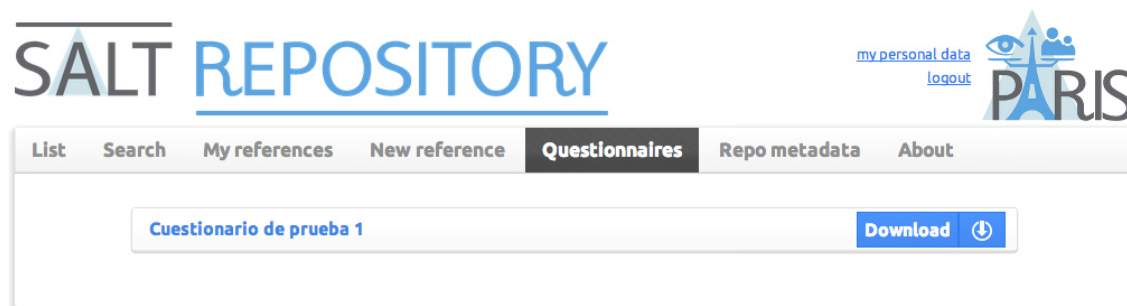


Figure 9. Interface to access questionnaires

At this point of the project, final questionnaires are still under development, hence the repository only stores a provisional one in order to show the interface functionality. An actual questionnaire for biometric systems (although it is not a definitive version) can be found in the appendix at the end of deliverable D6.2 “Biometrics Use Case SALT Compliant Framework”.

Moreover, the interface also counts with a space for the repository metadata. As it has already been stated, the repository is the physical storage of the SALT framework. Therefore, even though the SALT framework can be considered as an unique entity containing all SALT references, the SALT repository can be (and in fact, it will be) a series of distributed databases. Because of this, repositories from different countries, different companies and alike, may have

different policies of use, different restrictions, etc. For this reason, the “Repo metadata” section shown in Figure 10 enables a place for a given repository exclusive information.



Figure 10. Window showing metadata exclusive for a given repository

Finally, the web interface also include an “About” tab where we can find a general description of the repository and a complete list of the members taking part in the PARIS project consortium. See Figure 11.



Figure 11. About tab

Once the user has finished his interaction with the SALT repository, he can logout the system, as depicted in Figure 12.

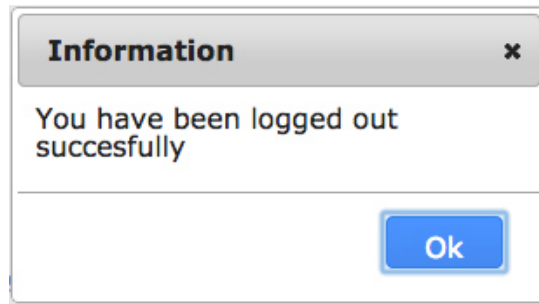


Figure 12. Logout window

3 Questionnaire Tool Implementation

This tool is an editor that will display the different questionnaires, that helps SALT users for designing their surveillance systems. These questionnaires will be prepared by experts in surveillance in several scopes: legal, technical and socio-ethical.

The description of this tool in this document will be divided in different points. The first one will be the structure of the data base model, describing the tables and their fields in depth. Then, screenshots of the tool are shown.

3.1 Data base model

The data base model has been thought in order to allow user:

- 1) View the questionnaire with questions in order.
- 2) Answer its questions in the order they want.
- 3) Save the questionnaire without finishing it and later come back to finish it.
- 4) Delete questions they do not want or need to answer.
- 5) For admin users, they are available to create and update questionnaires.

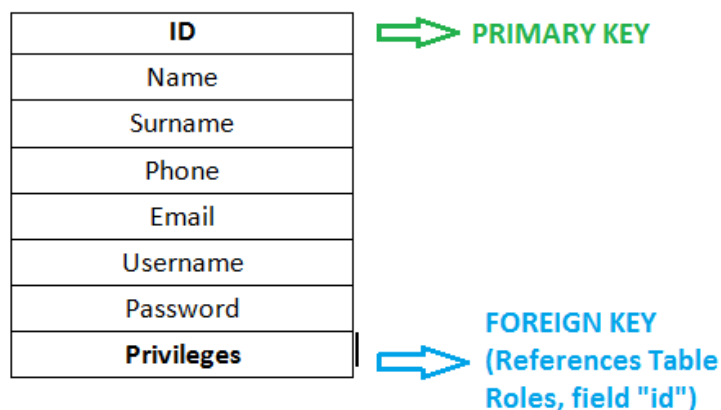
3.1.1 User tables

In this tool will be two types of users, administrators or SALT experts and SALT users. Here it is not necessary to differentiate between legal, technical or socio-ethical experts because each questionnaire has questions of several scopes, due to this all type of experts should have rights to add and change information in them.

The structure will have three different tables:

I. TABLE USER

This table will contain the information of the user, his personal data and his *user* data. It will have relation to the table Roles in order to define the privileges of each user.



Explanation of the fields of the table users:

- **ID:** It is the primary key of the table and identifies univocally each user.
- **Name:** It is the name of the user.

- Surname: It is the surname of the user.
- Phone: Telephone number of the user.
- Email: It is the email of the user.
- Username: It is the name that the user will use to log in the system.
- Password: It is the essential key to log in the system by the user.
- **Privileges**: It is a foreign key that refers to the field id in the table roles, this relation is one to one because a user can only have one role, no more.

II. TABLE ROLES

This table just detail the different roles that exist in the system. There are only two: admin or SALT expert and SALT user.

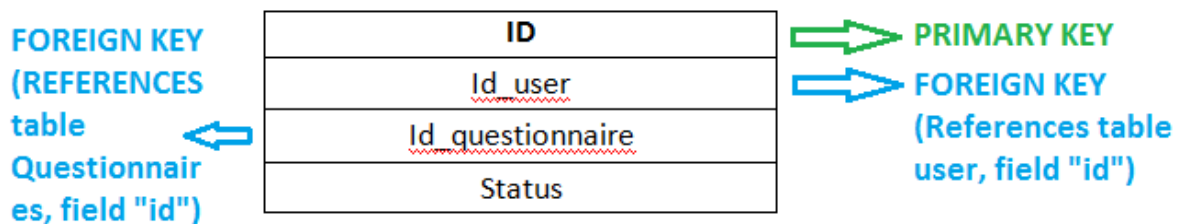


Explanation of the fields of the table roles:

- **ID**: It is the primary key of the table and identify each user.
- **Role**: It contains the privileges or type of user, in this case can only be saltuser or expertuser.

III. TABLE QUEST_USERS

This table details the questionnaires that are being used or filled by each user. The user can start to fill a questionnaire, saved it and log out of the system for finishing later the questionnaire. Maybe the user has finished that questionnaire and this should be registered, too.



Explanation of the fields of the table Quest_users:

- **ID**: It is the primary key that identifies each questionnaire that each user has used.
- **Id_user**: It is a foreign key that references to the table user, to the field id. This field identifies the user in this table.
- **Id_questionnaire**: It is a foreign key that references to the questionnaires table, to the field id. This field identifies the questionnaire that the user has used.
- **Status**: These fields tell if the questionnaire used by this user is:
 - *In progress*: the questionnaire is being filled in by the user.
 - *Saved*: the questionnaire has been saved by the user without finishing it.

- *Finished*: the questionnaire has been finished by the user.


3.1.2 Questionnaire tables

In these tables will be detailed the questionnaires, the questions of each questionnaires and the answers given by users to the questions of each questionnaire. In this part will be three tables: questionnaires, question and answers.

The table structure:

IV. TABLE QUESTIONNAIRES

This table contains the information about each questionnaire of the tool like title, author, status...etc. In this table there will be no information about the questions or answers. In this table it has been thinking in the SALT Experts that add new questionnaires

ID	
Title	
Author	
<u>Createdate</u>	
<u>Lastupdate</u>	
<u>Publishdate</u>	
Status	
Description	

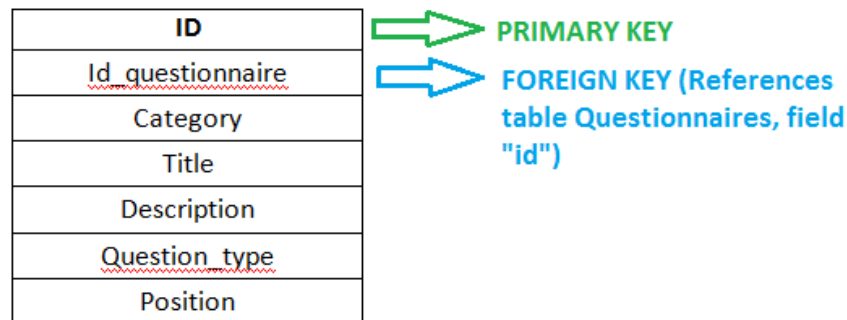
Explanation of the fields of the table questionnaires:

- **ID**: It is the primary key of the table that identifies each questionnaire. This is field is referenced by two tables: question and quest_users.
- **Title**: It describes the title of the questionnaire, the name.
- **Author**: It contains the name and surnames of the person that created the questionnaire that should be an expert user.
- **Createdate**: It contains the creation date of the questionnaire.
- **Lastupdate**: It has the last date of updating of this questionnaire.
- **Publishdate**: This contains the date when the questionnaire was published in order to be seen by the SALT users.
- **Status**: It says if the questionnaire is in progress, so it is being created, or it is published, so it is online and the SALT users can answer the questions.
- **Description**: It describes the content of the questionnaire, it details about the goal and questions of the questionnaire.

V. TABLE QUESTION

This table describes in its fields the information referenced to the different questions of each questionnaire. In a questionnaire the user is allowed to change the position of the questions or answer the questions in the order he wants. Not only this, the user can

delete questions of the questionnaire that is answering to not be shown for him, again. All these functions should be possible to accomplish with these tables.



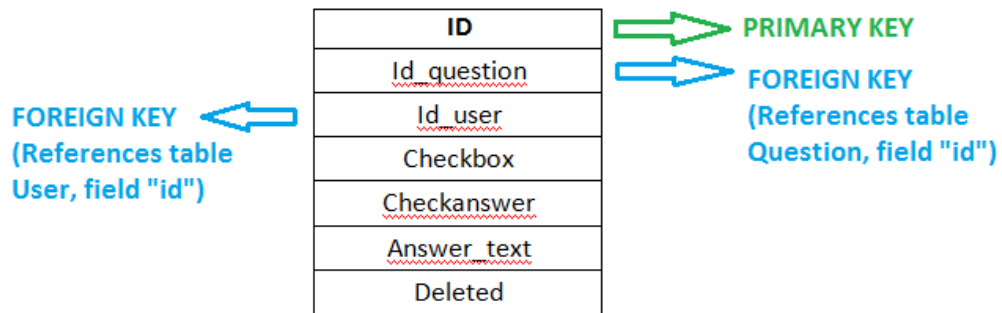
Explanation of the fields of the table questionnaires:

- **ID:** It is the primary key of the table that identifies each question in each questionnaire.
- **Id_questionnaire:** This is the foreign key that says us to which questionnaire belongs the question.
- **Category:** This field identifies to which scope belongs this question because there are several scopes of questions in a questionnaire. Usually the scopes are legal, technical and socio-ethical but they can have another title like Proportionality, interference with privacy rights...etc.
- **Title:** This field is the body of the question, really, the question.
- **Description:** This field will have detailed information about what is the goal of the question in order to help the user in his answer.
- **Question_type:** This field tells if the question is a simple question or it has more questions within it. There are other questions that can be written for answering with checkbox options, so for saving them in the data base it is necessary to know if the question has no more subquestions or has more questions in the format of checkbox (multiple choices).
- **Position:** This field contains the position of the question in the questionnaire for being displayed. But this position can be changed by the user, if he deletes a question the position will change and if he decides to answer before other questions, too.

VI. TABLE ANSWERS

This table saves the answers given by the user to the different questions of the questionnaire. Each answer will be identified by its question and the user who sent it to save it in the database. It is not necessary to identify here the questionnaire because there is a foreign key for this in table question.

There are different types of answers, they can be a text, an explanation written by the user or an answer of a checkbox question. This kind of answers can have text/explanation and a phrase (*checkanswer*) that it is displayed in the questionnaire.



Explanation of the fields of the table questionnaires:

- **ID:** It is the primary key that identifies each answer.
- **Id_question:** This is a foreign key that references the answer to the question, it means that references to the id field of the question table. With this foreign key it is had the relation question-answer.
- **Id_user:** This is another foreign key that references to the table user, so that relates to the user that has answered the question with the answer.
- **Checkbox:** This field says if the answer is in a checkbox with a yes or a no answer. Although the answer is with yes or no to fill in the checkbox, it also has a text box for writing more answer information. If the answer is no the value will be 0 but if it is yes, the value will be 1. There are other options, 2 when it is another text instead of yes o no and the last option will be 3 that means there is no check answer.
- **Checkanswer:** Sometimes the answer is written with different checkbox for choosing them but this text is not a yes or no. So in this field will be saved the answered checked. If this field is not empty the Checkbox field should show a 2. If the answer is not with checkbox, then this field will be null.
- **Answer_text:** It contains the explanation answered by the user to the question.

Deleted: This field shows if the question that this answer belongs to was deleted by the user in the questionnaire. If it was deleted this should show a 1, if not a 0. If it was deleted the question to which it belongs to, then the fields checkanswer and answer_text will be null.

3.2 Overview of the Questionnaire Tool

The questionnaire tool is a web application. The following figures are screenshots of the tool.

SALT QUESTIONNAIRE PARIS

Login Register

Name

Surname

Phone

Email

Username

Role

Password

Confirm

Register

©2014 All Rights Reserved Trialog | Design by Trialog | PARIS Project

Figure 13 Register Window

Figure 13 is the window in order to register. Since the tool is not public yet, no specific controls are applied. For a deployment, each account should be validated and privileges have to be given. When a user has credentials, he is able to login through the window shown in Figure 14.

SALT QUESTIONNAIRE PARIS

Login Register

Username

Password

Remember me

©2014 All Rights Reserved Trialog | Design by Trialog | PARIS Project

Figure 14 Login Window

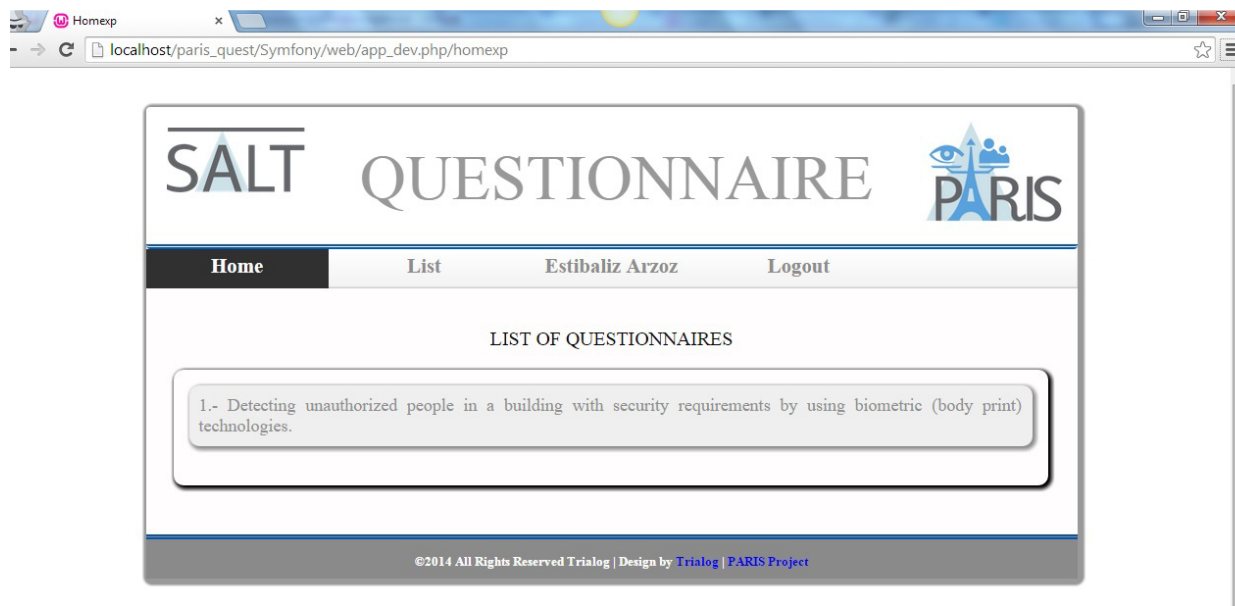


Figure 15 List of questionnaire

When a user is logged in, Figure 15 shows all the questionnaires stored in the repository and available to the user. After a questionnaire is selected, it is possible to answer it. The list of questions are screened and an example is given in Figure 16

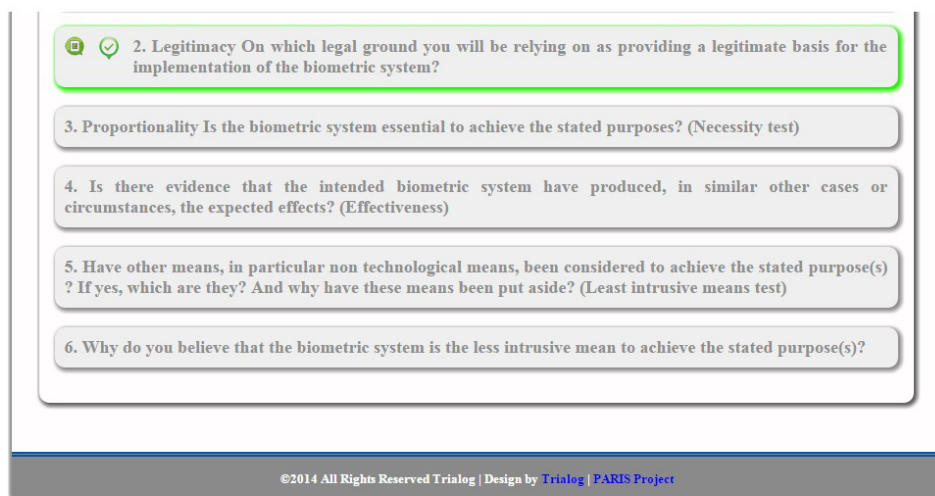


Figure 16 Non-answered Questions

Thanks to colour and icons schemas, it is very easy to identify the progress. The font is black or grey according to the ability to answer to the question. The border is green when the user has already provided an answer. Additionally, it is possible to validate a questionnaire (i.e., to check that all mandatory questions are answered). In the case of invalid questions, the border is red. Finally, icons have been set up in order to enhance the meaning.

The screenshot shows a web interface with a navigation bar at the top containing 'Home', 'List', 'Estibaliz Arzoz', and 'Logout'. The main content area is titled 'QUESTIONNAIRE 1: DETECTING UNAUTHORIZED PEOPLE IN A BUILDING WITH SECURITY REQUIREMENTS BY USING BIOMETRIC (BODY PRINT) TECHNOLOGIES.' Below this, a question is highlighted in a grey box: '1. What is/are the purposes of the biometric system?'. Underneath the question, the category is 'CATEGORY: 1. OPPORTUNITY OF THE SYSTEM'. A detailed description follows, explaining the objective of the question and the need for an internal assessment. At the bottom, there is a text input field labeled 'Respuesta a la primera pregunta.' and a blue 'Save Answer' button.

Figure 17 Question Selected

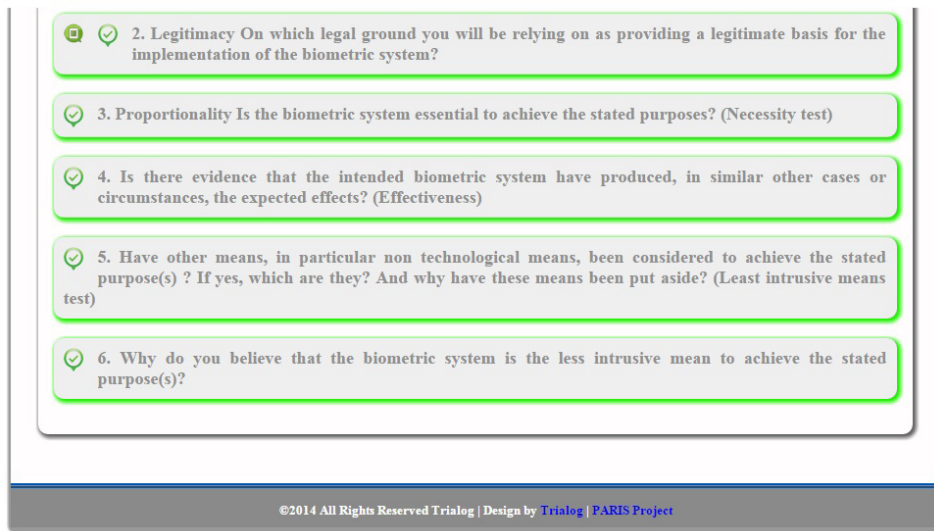
Figure 17 shows how a question is screened:

- Title of the question
- The category
- The description of the question in order to have a better understanding of the question
- A window for providing the answer

The screenshot shows a web interface with a list of questions. The first question, '1. What is/are the purposes of the biometric system?', is highlighted with a green border and a checkmark icon. The second question, '2. Legitimacy On which legal ground you will be relying on as providing a legitimate basis for the implementation of the biometric system?', is also highlighted with a green border. Below this question, the category is 'CATEGORY: 1. OPPORTUNITY OF THE SYSTEM'. A detailed description follows, explaining the objective of the question and the need for an internal assessment. Below the description, three sub-questions are listed, each in a green-bordered box with a checkmark icon: '21. Consent of the individual?', '22. Performance of a task carried out in the public interests or in the exercise of official authority vested in the controller.', and '23. Legitimate interests pursued by the organization/person responsible of the biometric system?'.

Figure 18 Complex Question

In some case, as shown in Figure 19, some questions can be complex by including some sub questions. By clicking on a complex question, the sub-questions are displayed. In order to highlight complex questions, an icon is set up (for example, see question 2.3 in Figure 18). In order to reduce the complexity, we have limited the hierarchy of complex questions at two levels.



The image shows a screenshot of a questionnaire interface. It contains five questions, each in a separate rounded rectangular box. Each box has a green checkmark icon on the left, indicating that the question has been answered. The questions are:

2. Legitimacy On which legal ground you will be relying on as providing a legitimate basis for the implementation of the biometric system?
3. Proportionality Is the biometric system essential to achieve the stated purposes? (Necessity test)
4. Is there evidence that the intended biometric system have produced, in similar other cases or circumstances, the expected effects? (Effectiveness)
5. Have other means, in particular non technological means, been considered to achieve the stated purpose(s) ? If yes, which are they? And why have these means been put aside? (Least intrusive means test)
6. Why do you believe that the biometric system is the less intrusive mean to achieve the stated purpose(s)?

At the bottom of the interface, there is a footer that reads: ©2014 All Rights Reserved Trialog | Design by Trialog | PARIS Project

Figure 19 Answered Questions

Finally, when all questions are answered, all blocks should be green as presented in Figure 19.

4 Conclusion

The project has developed two tools for two important phases in the surveillance system life cycle: PIA and privacy-by-design. These tools will be use in the context of the demonstrators. Feedback will be used in order to enhance the current implementations