COMPETITIVENESS AND INNOVATION FRAMEWORK PROGRAMME
ICT Policy Support Programme (ICT PSP)

Towards pan-European recognition of electronic IDs (eIDs)

ICT PSP call identifier: ICT-PSP/2007/1
ICT PSP Theme/objective identifier: 1.2

Project acronym: STORK

Project full title: Secure Identity Across Borders Linked
Grant agreement no.: 224993

D5.8.1d Security Principles and Best Practices

Deliverable Id :	**D5.8**
Deliverable Name :	**D5.8 Technical Design**
Status :	**Final**
Dissemination Level :	**Public**
Due date of deliverable :	**31/7/2009**
Actual submission date :	**9/9/2009**
Work Package :	**5.1**
Organisation name of lead contractor for this deliverable :	**BE FEDICT**
Author(s):	**Marc Stern**
Partner(s) contributing :	**AT TUG, BE FEDICT, DE BSSI, IT POLITO, PT MULTICERT**

Abstract: This document aims at the description of security requirements that have to be fulfilled by the interoperability layer developed in the STORK project. As the STORK project is concerned with interoperability issues between governmental institution within the EU, personal data of EU citizens are processed, transmitted and temporarily stored by the interoperability layer. Hence, the assets to protect in STORK are personal information of citizens, issued by governmental or other institutions. Security in this context is concerned with the protection of these assets. A security-specific impairment of the assets typically includes the loss of asset confidentiality, loss of asset integrity or loss of asset availability. The STORK interoperability layer must provide sufficient security functions that counter the identified threats. This document gives a detailed description of identified threats, derived security objectives and necessary security functions that shall be implemented by the STORK system. The threats, objectives and functions define a sound set of security requirements to be fulfilled by the STORK system.

Project co-funded by the European Community under the ICT Policy Support Programme
© Copyright by the STORK-eID Consortium

History

Version	Date	Modification reason	Modified by
0.1	15-01-09	Initial draft	M. Stern, BE Fedict
0.2	23-01-09	Incorporated remarks from Matthias Schwan	M. Stern, BE Fedict
0.3	02-02-09	Incorporated remarks from the WG session of 26-01-09 Grouped several principles into one	M. Stern, BE Fedict
0.4		Added threats from WP2	M. Schwan, DE BSSI
0.5	18-02-09	Added additional threats New structure Added security functions and mapping	M. Stern, BE Fedict
0.6	24-02-09	PROPOSAL: new structure (threat → attacks)	T. Rössler, AT-TUG
0.7	03-03-09	Integration of new structure and all remarks (AT TUG, DE BSSI, IT POLITO, PT MULTICERT)	M. Stern, BE Fedict
0.8	30-03-09	Integration of remarks from Enisa and Carlos Ribeiro	M. Stern, BE Fedict
0.8.1	28-04-09	Re-organisation, renaming of objectives, etc.	M. Stern, BE Fedict
0.9	06-05-09	Added remarks from Thomas Rössler, and filled all cross tables	M. Stern, BE Fedict
1.0	01-06-09	Final functional version	M. Stern, BE Fedict
1.0.1	21-06-09	Corrections	M. Schwan
1.0.2	15-07-09	Added SP authentication	M. Stern, BE Fedict
1.0.3	12-08-09	Clarifications after comments from the consortium	M. Stern, BE Fedict
1.0.4	04-09-09	Quality Check	Capgemini
1.0.5	08-09-09	Resolved formatting issues	M. Stern, BE Fedict
final	09-09-09	Quality review & finalization	Capgemini Quality Team

Table of contents

HISTORY 2
TABLE OF CONTENTS 3
LIST OF TABLES 5
LIST OF FIGURES 6
1 INTRODUCTION 7
1.1 SCOPE AND OBJECTIVES 7
1.2 METHODOLOGY 7
2 SYSTEM OVERVIEW 11
3 ATTACKS 14
4 THREATS 17
4.1 IDENTITY THEFT 17
4.2 PRIVACY 18
4.3 ACCOUNTABILITY AND USER CONTROL 19
4.4 IMPLEMENTATION AND OPERATION 20
5 SECURITY PRINCIPLES 22
6 SECURITY OBJECTIVES 24
6.1 IDENTITY PROTECTION (AUTHENTICATION) 24
6.2 PRIVACY PROTECTION 25
6.3 ACCOUNTABILITY AND USER CONTROL 25
6.4 SYSTEM IMPLEMENTATION AND OPERATION 26
7 BEST PRACTICES 28
8 SECURITY FUNCTIONS 29
8.1 ACCOUNTABILITY AND USER CONTROL 29
8.2 PRIVACY PROTECTION 29
8.3 DESIGN AND IMPLEMENTATION 31
8.4 IMPLEMENTATION AND OPERATION 31
9 SECURITY TECHNICAL RECOMMENDATIONS 35
9.1 JUST-IN-TIME VALIDITY 35
9.2 STRONG AUTHENTICATION 35
9.3 AUDITING 36
9.4 PRIVACY PROTECTION 36
9.5 IMPLEMENTATION AND OPERATION 38
9.6 STRONG KEYS AND CRYPTOGRAPHIC MECHANISMS 38
10 APPENDIX: SECURITY TABLES 41
10.1 TABLE OF ATTACKS 41

10.2 TABLE OF SECURITY THREATS 42
10.3 TABLE OF SECURITY PRINCIPLES 43
10.4 TABLE OF SECURITY OBJECTIVES 44
10.5 TABLE OF SECURITY BEST PRACTICES 45
10.6 TABLE OF SECURITY FUNCTIONS 46
10.7 TABLE OF SECURITY RECOMMENDATIONS 47
11 REFERENCES 48

List of tables

Table 1: Attacks vs. Threats 21
Table 2: Threats & Principles vs. Objectives 27
Table 3: Objectives vs. Functions 33
Table 4: Attacks vs. Functions 34
Table 5: Recommendations vs. Functions & Best practices 40

List of figures

Figure 1: Security approach.....8
Figure 2: System Overview.....11

1 Introduction

1.1 Scope and objectives

This document aims at the description of security requirements that have to be fulfilled by the interoperability layer developed in the STORK project. As the STORK project is concerned with interoperability issues between governmental institution within the EU, personal data of EU citizens are processed, transmitted and temporarily stored by the interoperability layer. Hence, the assets to protect in STORK are personal information of citizens, issued by governmental or other institutions. Security in this context is concerned with the protection of these assets. A security-specific impairment of the assets typically includes the loss of asset confidentiality, loss of asset integrity or loss of asset availability. The STORK interoperability layer must provide sufficient security functions that counter the identified threats. This document gives a detailed description of identified threats, derived security objectives and necessary security functions that shall be implemented by the STORK system. The threats, objectives and functions define a sound set of security requirements to be fulfilled by the STORK system.

1.2 Methodology

To find and describe security requirements, a methodology and approach depicted in the figure below is applied. First, the threats the system could face are given, which are partly motivated by known attacks. Then security objectives are derived from the identified threats and from requirements coming from the over-all project. Thirdly, security functions are defined that implement the security objectives and counter the threats. Lastly, the document gives some practical recommendations how to put the abstract security functions in practice. These security technical recommendations receive input from commonly accepted security mechanisms and best practices. A summary of the used terminology is given in the table below.

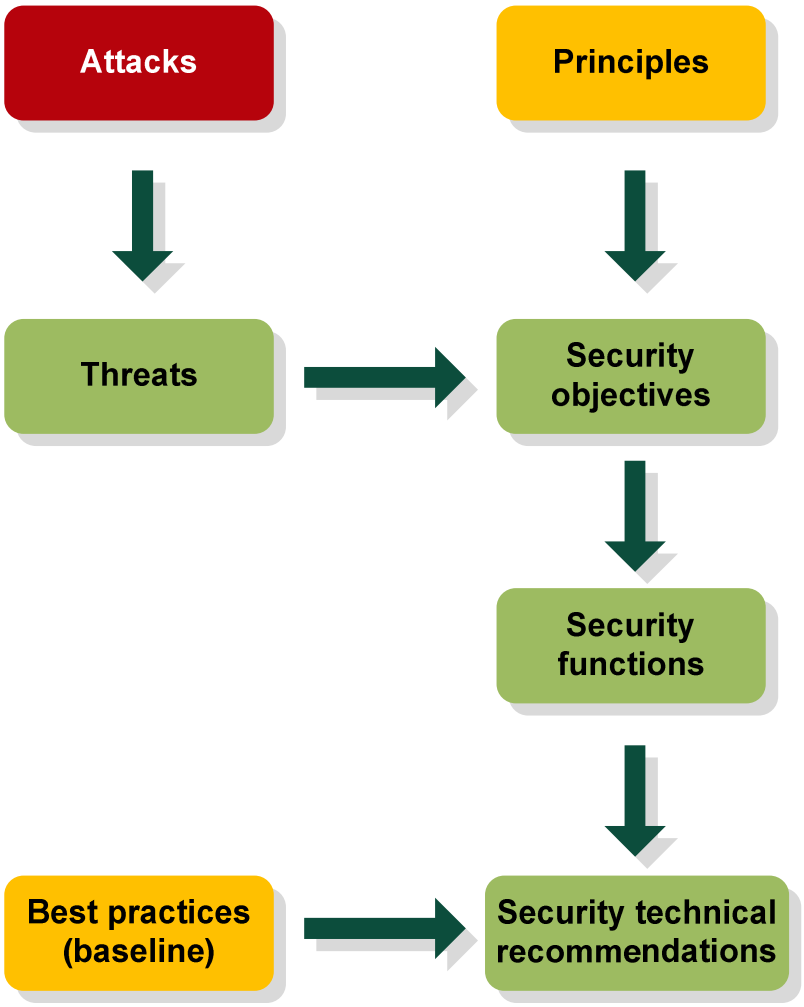


Figure 1: Security approach

In order to describe security requirements this document first gives an overview of the system under consideration in Chapter 2, i.e. the STORK interoperability layer. The chapter defines all components and communication relations of the system as they are given by the architectural design.

Based on the assets to be protected typical known attacks on IT systems in general and the STORK system in particular are discussed in Chapter 3. The collection of attacks leads to various threats the STORK system could face. Chapter 4 lists the threats taking into account aspects of identity theft, privacy, user control and system design and implementation. The chapter also gives a mapping of attacks to the identified threats.

In addition to threats, security principles are a valuable input for the definition of security objectives. These security principles either derive from the Document of Work of the STORK project, or were agreed on STORK work package meetings. The common security principles are given in Chapter 5.

Security objectives are the counterpart of one or more threats and hence, are derived from threats and security principles. Chapter 6 lists the security objectives following the same structure as in Chapter 4 Threats. The objectives cover issues of identity and privacy protection, user control and system implementation. A mapping of threats to objectives shows that there is at least one objective countering a threat.

Security functions define a realisation of a particular security objective leaving out specific implementation details. For instances, the objective of confidentiality, which may counter the threat of disclosing citizen data, could be realised by either a cryptographic protocol and encryption mechanism or by physical protection, e.g. communication takes place in a trusted environment. The former alternative neither states a particular protocol nor a specific key length. Such practical recommendations of implementing security functions are discussed in the final Chapter 9. The definition of security functions in Chapter 8 is additionally influenced by Best Practices, which give valuable input from security organisations or infrastructure providers. Best practices are given in Chapter 7.

The used terminology is summarised in the following table.

Security Requirements:	Security requirements are the sound set of threats, objectives and functions. Security functions are to be implemented by a particular system, i.e. the STORK interoperability layer, in order to protect the defined assets from the identified threats.
Assets:	Citizen's personal data issued by a governmental institution to be processed, transmitted and temporarily stored by the STORK systems
Attacks:	attacks are the realisation or implementation of a threat and hence motivate the identification of threats
Threats:	Assets are protected from threats by countermeasures, i.e. security functions.
Security Principles:	Security Principles are statements or requirements coming either from the Statement of Work or from very strong decisions made in the early stages of the project at a global level.
Security Objectives:	Security Objectives derive from threats. They counter the identified threats and satisfy the security principles.
Security Functions:	Security Functions implement security objectives and hence, counter the threats.
Best Practices:	Best Practices are particular security functions recognised by the security sector and professionals. These practices are recommended by security organisations (SANS, NIST, OWASP, etc.), development framework providers (Microsoft, Sun, IBM, etc.), and infrastructure providers (firewall vendors, etc.). These recommendations are to be considered within the STORK system as a baseline needed for any solution, independently of any business or technical context. Best practices motivate security technical recommendations
Security technical recommendations:	Security technical recommendations are considered, by default, as mandatory, and should be followed by all STORK players (architects, designers, developers), even for components under Member States responsibility. Most requirements relate to the STORK system, but some are also recommendations for the part left to each Member State responsibility, either for the development of the national part of the system or for its operation. These recommendations will be identified with the following symbol: 

Although the terminology used in this document has been partially borrowed from Common Criteria, this document does not claim being a full Protection Profile for STORK components and systems. Therefore, the security considerations provided in this document focus on the most relevant elements only (see system overview provided in section 2). It should serve a basis for further security considerations which have to be undertaken in the course of implementation and deployment of components.

This document is part of the D5.8.1 Technical design, where a more complete summary and introduction are included.

2 System Overview

Figure 2 depicts all components required in a STORK-enabled scenario (the depicted scenario shows all components required for PEPS-PEPS and PEPS-MW processes) and thus outlines the floor plan of typical STORK pilot applications. The components that are Member State specific are coloured dark-grey; the common STORK components are coloured black.

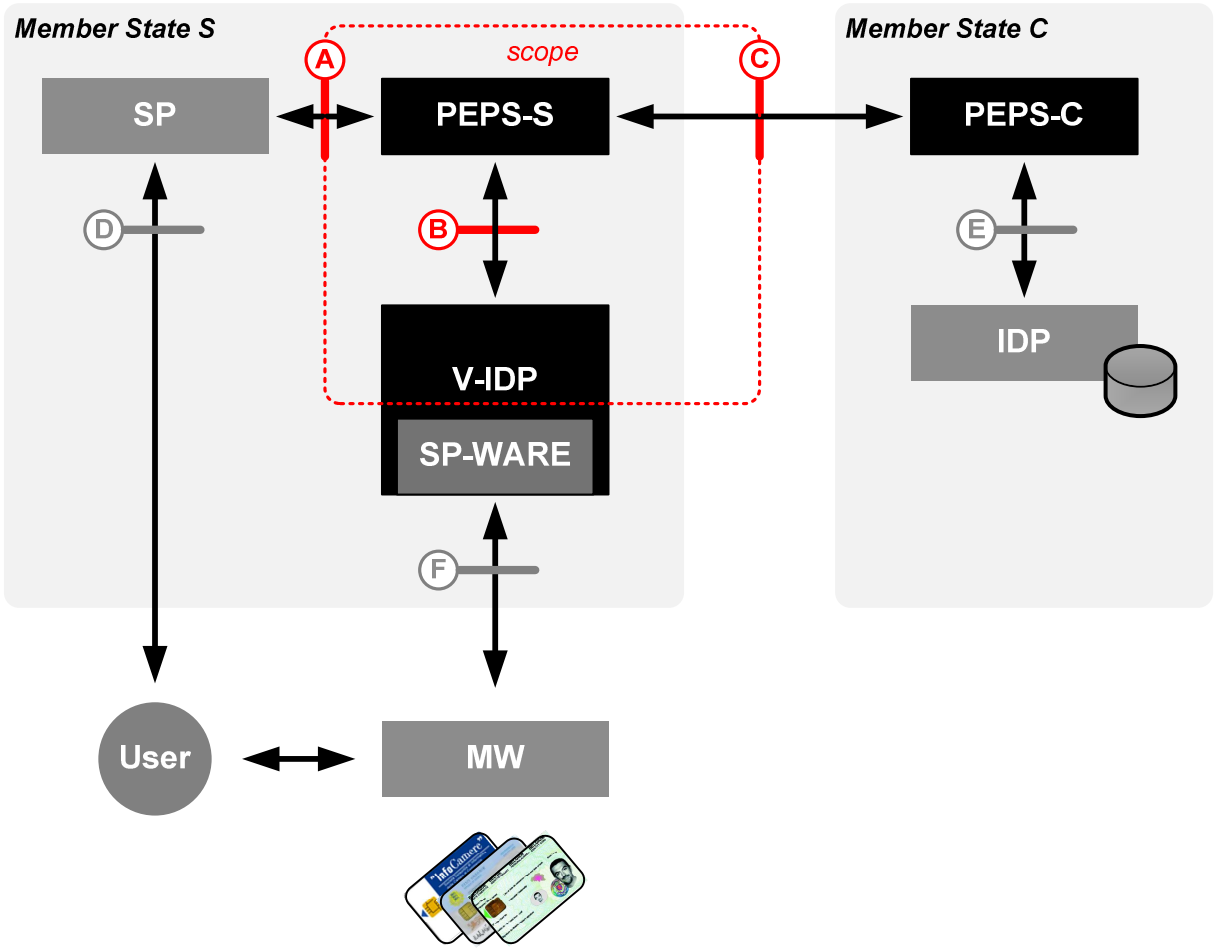


Figure 2: System Overview

(dark grey: Member State specific, black: common STORK components)

It is important to mention that this schematic is very abstract and focuses only on those aspects that are relevant for the security considerations presented in this document. Several details and aspects, e.g. user interactions, technical redirects through users' internet client (browser), etc., have been omitted. Furthermore, the paths given in Figure 2 identify the communication flows and the required interfaces only; this figure makes no further assumptions with respect to the underlying technical communication protocols.

This illustrative scenario requires the following components (according to the STORK terminology):

- **Service Provider (SP)**

  The Service Provider is located at Member State S. It provides some services the user aims to access/use.

- **PEPS (PEPS-S and PEPS-C)**

  It is assumed that Member State S makes use of PEPS-S (i.e. the PEPS that is located at the Member State S which is hosting the Service Provider) in order to identify and authenticate users. Furthermore, a second PEPS is assumed being deployed in Member State C, i.e. PEPS-C (the citizen's domestic PEPS) providing authentication for users of Member State C.

- **Virtual Identity Provider (V-IDP)**

  The PEPS of Member State S is able to access the Middleware (MW) of other Member States through a Virtual Identity Provider (V-IDP). The V-IDP is connected directly to Member State S's PEPS.

- **SP-WARE**

  The V-IDP running aside the PEPS of Member State S makes use of SP-WARE in order to access Member State specific Middleware (MW) implementations. Those Member States that use MW to access e-ID tokens provide their specific SP-WARE implementations.

- **Middleware (MW)**

  The Middleware (MW) is Member State specific. It is used to access the user's e-ID token. The MW is a component at the user's side and it is Member State specific.

- **Identity Provider (IDP)**

  Member State C uses the Identity Provider to provide entity authentication. It is Member State specific.

- **User**

  The user is an entity that aims to access/use a service provided by the SP of Member State S. The user may use either a PEPS or MW.

This scenario identifies the following major interfaces (major interfaces from the STORK perspective):

A) SP <> PEPS
B) PEPS <> V-IDP
C) PEPS <> PEPS
D) SP <> User
E) PEPS <> IDP
F) SP-WARE <> MW

Interfaces A, D, E and F are Member State specific.

The scope of the security considerations presented in this document focus on the components covered by the dotted square in Figure 2, i.e.:

- PEPS,
- PEPS-specific part of the V-IDP
- and the interfaces A, B and C.

As this document aims to give general security recommendations, it also provides recommendations and best practices applicable for all affiliated components as well. Furthermore, this document makes assumptions and defines policies especially for those components that directly connect to STORK components (i.e. components beyond interface A, B and C).

3 Attacks

This section gives a collection of attacks and risks that may threat our system. These attacks are considered being "helping friends" in order to achieve an almost complete list of threats. Thus, the following list of concrete attacks should help to find concrete threats and to deepen the understanding. The following enumeration of attacks is sorted arbitrarily and does not claim to be complete.

A1 Spoofing

Spoofing is a means to hide one's real identity. This attack is heavily used at the network level (IP address spoofing), or in e-mail exchanges (origin address spoofing), but can be used in any system reliable on any kind of identity or identifier.

A2 Guessing

Guessing is a simple attack where a malicious entity tries to guess a secret used in a communication (e.g., an encryption key, a PIN) for instance for entity authentication. This attack works in cases where the secret is weak. For instance, a simple password can be easily guessed using dictionaries.

The security level provided by users' passwords, PINs, etc., are not covered here, as they are under each Member State's responsibility.

A3 Communication eavesdropping

Eavesdropping is an attack that consists in observing the messages passing through a communication channel, which could be, for example, the credentials of an authentication protocol. The messages are stored usually for performing some off-line analysis of the information, used for launch successive attacks; for example eavesdroppers generally attempt to obtain tokens to pretend to be the claimants.

A4 Communicated data tampering

Tampering is an attack that consists in changing some data passing through a communication channel. This could be used, for example, to change the recipient account number in a Web Banking application.

A5 Session hijacking

Hijacking is an attack consisting in taking over an already authenticated session to learn sensitive information information, or to perform actions in the name of the authenticated entity.

A6 Replay Attack

Replay is a form of attack where a malicious entity repeats or delays previously intercepted messages in order to gain access to sensitive information.

A7 Echo Attack

Echo is a form of attack where a malicious entity sends a message back to its originator, usually leading to secret information disclosure.

A8 Man-in-the-middle Attack

Man-in-the-middle attacks are a form of active eavesdropping – and possibly tampering – in which the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection when in fact the entire conversation is controlled by the attacker. The attacker must be able to intercept all messages going between the two victims and inject new ones.

A9 Skimming

Skimming is an attack particularly applicable to smart card based eID tokens. Skimming stands for the class of attacks where an attacker gains illegitimate access to the data of the smart card.

Application Note: This analysis does not deal with this attack as it is under each Member State's responsibility.

A10 User profiling

Profiling is the recording and classification of user behaviours. This occurs through aggregating information from, e.g. online and offline purchase data, supermarket savings cards, white pages, surveys, sweepstakes and contest entries, financial records, property records, credit card transactions, phone records, credit records, product warranty cards, the sale of magazine and catalogue subscriptions or public records. Profiling has sparked an entire industry euphemistically labelled "Customer Relations Management" (CRM) or "Personalisation".

In the particular case of online services, users typically do not get online access to all their personal data including those being stored in log files or being processed by profiling, scoring or data mining systems.

A11 Action repudiation

An entity could potentially deny an action it performed.

A12 Attack without trace

An attack (successful or not) could be unnoticed because no trace about it is available on the system(s).

A13 Attacker covers trace

A successful attack could be hidden by the attacker by, for instance, deleting the tracks he left (log files, etc.)

A14 Incorrect design and implementation

Implementation (coding) of a component of the system could contain vulnerability in the design or the code. This could lead to a security hole.

A15 Incorrect Usage (Parameterisation)

An incorrect use of the system by an entity (PEPS, SP, IDP, AP) could lead to a security hole. This is typically the case with a too permissive policy or use of a request, or new behaviour of an updated module.

A16 Unauthorised access

This attack consists in gaining an access to a system that should have been disallowed. This could happen at different levels: network, Operating System, or application (whether it is the front-end application, or a back-end one, like the database for instance).

A17 Fuzzing

Fuzzing is an attack consisting in injection unexpected data to a system or application to confuse it. This could lead to unexpected behaviours like crash, unauthorised access, arbitrary code execution, internal information disclosure, etc.

A18 Race condition

A race condition occurs when two processes do not perform in the sequential manner that was intended per business rules. This type of attack aims at interacting with the system between two steps that should normally be sequential, in order to bypass some control in the second process based on data collected during the first one.

A19 Denial of Service Attack

Denial of Service is an attack consisting of using a service in such a way that the system becomes unavailable for genuine users, either by saturating the service, or by crashing it, or by locking an exclusive resource blocking the system for other users.

A20 Social engineering

Social engineering is an attack intended to fool a user of the system (a citizen, or an operator/administrator) to get him into an action that would benefit to the attacker. This could be an e-mail asking a citizen to send his password for verification, or a phone call to an administrator to change a setting, etc. Those attacks may also aim at confusing the user by displaying text or interactive fields that could be wrongly interpreted and lead the user to wrong interactions.

4 Threats

This section analyses possible threats to the STORK framework (i.e. systems, components, processes, etc.). In order to have a structured analysis this section deals with the following threat classes:

1. Threats regarding Identity Theft
2. Threats regarding Privacy
3. Threats regarding Accountability and User Control
4. Threats regarding Implementation and Operation

Depending on the chosen abstraction layer further threats and threat classes might appear. However, the following threat classes are explicitly out of scope of this security analysis:

- Threats regarding eDocuments
- Threats regarding Service Providers

The aforementioned threat classes, which are not further discussed in this analysis, have to be considered by the providers of services, applications and STORK components.

In addition to the threats and threat classes defined in this chapter, chapter 3 gives a list of risks and attacks relating to these threats.

4.1 Identity Theft

The STORK Glossary [2] defines the Identity of an entity (where this notion is inclusive of human users and system components alike) as the dynamic collection of all of the entity's attributes. Thus, an Identity Theft aims at obtaining credentials of an entity in order to impersonate him/her/it. WP2 [3] identified and summarised a number of attacks (in together with other ones coming from trusted security repositories like SANS, NIST, OWASP, etc.).

This section is not limited to the impersonation of persons (i.e. natural person and legal persons) but deals with the impersonation of systems as well. Therefore, this analysis makes use of the term 'entity' instead of person or party.

The following list defines concrete threats of this class:

T1 Impersonation of a citizen

One or several STORK components are erroneously led to believe that they are communicating with a citizen while communicating with some other entity, allowing this entity to perform actions in the name of the citizen. This way, the fraudulent entity circumvents the authorisation mechanism, or to performs illegal or abusive actions in name of the citizen.

T2 Impersonation of system

Some entity impersonates a STORK system/component/server/application so that other entities (i.e. persons, or systems) believe that they are communicating with a real STORK system/component/server/application. Well known examples of this attack are network address spoofing (IP address), or e-mail address spoofing (origin address), but these attack can be used towards any system relying on any kind of identity or identifier.

T3 Identity data forge

Generation, modification, insertion, or deletion of identity data during exchange between STORK parties. Either this could be used to gain access to services that should not be allowed, or, if another user's identity data is modified, to block the victim to access a service.

4.2 Privacy

Since the term "*privacy*" is defined differently in various countries, we focus on the already harmonised definition given in [1] and [2]:

> *Privacy is the right of an entity – in this context usually a natural person – to decide himself when and on what terms its attributes should be revealed. Privacy can alternatively be described as the freedom of a natural person to sustain a "personal space", free from interference by other entities. In an ID Management context, privacy is mostly used as a synonym of "informational privacy", i.e. the interest of a natural person to control, or at least significantly influence the handling of data about themselves, also taking into account the nature of the applicable attributes and the entity in charge of data management.*

Note that the goal here is to provide as much privacy protection as possible, but some governments may impose additional privacy protections as well (like forbidding transmitting a national identifier, even with the citizen consent, etc.).

The following list defines concrete threats of this class:

T4 Privacy – user data

Disclosure of user information, e.g. personal data, documents, messages, decisions, etc.

T5 Privacy - trail

Gathering and disclosure of data trails used by an application for security, debugging, or performance measuring purpose. These data trails may be used for data mining or profiling, or to get some internal data helping to launch an attack.

T6 User profiling

Information available in some parts of the solution, like in a PEPS, could potentially be correlated with other ones in order to profile or trace a user's connections.

T7 Unawareness of privacy issues

Many people are completely unaware of the important privacy issues that arise from the use of new data collection technologies, social networks, pervasive technologies, etc. Others feel uneasy about some kinds of data processing, but still cannot fully grasp the potential consequences of their actions on their own privacy. Still others may be aware of the privacy issues but do not know what to do to protect their privacy. Those that want protection often decline to participate in the digital world and therefore cannot reap the benefits of the information society. Those that want to reap the benefits often give up on their privacy protection, viewing they have no choice in the matter." – see [4].

T8 Usability of privacy protecting tools

For users that know how to maintain their private sphere, taking the necessary steps may be too costly or too cumbersome for them. The same is true for situations when their privacy rights have been violated. Actions for redress are usually time-consuming, and in several cases the effects of the privacy infringement are not revocable anyway.

Since privacy effects are often based on user actions, the challenges are even more urgent if the goal is to protect users privacy in information technology, e.g., for the elder generation or for handicapped people. Usability is an important, yet not satisfactorily solved issue when designing tools for protecting one's privacy – see [4].

4.3 Accountability and User Control

One of the most important design principles of STORK processes is that the user should be in control of it. This implies that all actions should be transparent to the user. Furthermore, all security relevant actions should require an explicit consent from the user. Transparency requires having a user interface that is understandable and easy to use.

On the other hand, all actions – especially all security related actions – should be logged. Logs should provide not only evidence of user actions, but also traces of attack detection.

The following list defines concrete threats of this class:

T9 Denial

A party could potentially deny an action it performed.

T10 User – Accidental misuse

Any form of accidental misuse, for example due to bad user interface design or lack of clear instructions, leads to unintended results, like a user accepting some behaviour without understanding the consequences.

T11 User – Forced misuse

A user is led to misuse the system.

One particular group of users very sensitive to this threat are the users with disabilities; as an example, an attack may target blind users using screen reader software.

Another case is when the user consent is forged for some actions, leading the system to conduct actions that were not explicitly allowed by the user.

T12 Log missing

Some illegitimate actions are unnoticed because no trace (log) about the access is available on the system(s).

T13 Log forged

The audit log gets forged by insertion, modification or deletion of data in a way that illegitimate actions are not detected.

4.4 Implementation and Operation

Various threats relate to the design/implementation of the system and its operation conditions. This analysis does not deal exhaustively with operation conditions, as they are under the responsibility of system/service providers (i.e. Member States and their sub-contractors). Therefore, this section approaches these issues only with one very generic threat (T18).

The following list defines concrete threats of this class:

T14 System compromise

The system may be compromised, using its own interfaces, by exploiting its interfaces, or parts of it to exploit vulnerabilities from the design or the code.

T15 System malfunction

The system does not work properly according to the specifications. This could be used by an attacker to misuse the system.

T16 System Denial of Service

The system is prevented from servicing legitimate entities requests by exploiting some design or code vulnerability or by consuming all resources in illegitimate or abusive actions.

T17 System availability

The availability of the service is not ensured due to bad design, bad operation conditions, etc.

T18 Operation security

The operation environment is abusively accessed by privileged operation interfaces, either by legitimate operators or by external entities that were able to circumvent the physical or remote access barriers.

The following table shows the relations between attacks and threats:

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18
A1	💣	💣																
A2	💣	💣														💣		
A3	💣	💣		💣												💣	💣	
A4			💣															
A5	💣	💣																
A6	💣	💣														💣		💣
A7				💣														
A8	💣	💣	💣	💣												💣	💣	
~~A9~~																		
A10						💣												
A11									💣									
A12													💣					
A13														💣				
A14															💣	💣	💣	💣
A15															💣	💣	💣	💣
A16															💣	💣	💣	
A17															💣	💣	💣	💣
A18																💣		
A19																	💣	💣
A20							💣	💣		💣	💣							

Table 1: Attacks vs. Threats

5 Security Principles

These principles have been either deduced from the DoW, or agreed during the various WP meetings.

P1 User centric – info

The user must see the exact information that will be transmitted to an entity, the exact origin and destination of this information. The context where it will used may also have to be displayed, like the sector (government, public, health, …), a liability disclaimer, the privacy or data usage policy, or any other context depending on each country legislation.

The user must explicitly give his consent before revealing any personal information to an entity.

Remarks:

- All information must be displayed in a user-understandable format, not in coded information – ex: “profession=doctor”, not “attribute n° 45677=59946”)
- If the law allows this in some countries, the user may be presented with the data meta-information only, without the actual content – ex: “Your birth date will be sent to …”
- Origin and destination have to be understood in the context of Stork only. Origin is a trusted identity or attribute provider, which may be based other services or repositories. Destination is the Service Provider, not any other entity linked to it who may potentially receive some information.
- Destination display is important for security, to ensure the user that the data will go to the right entity, but also to ensure non repudiation.
- Consent may consist, in the simplest case, of a click on a button, or to involve mechanisms that are more complex if the legislation requires it (ex: digital signature).

P2 Just-in-time validity

Any information transmitted or presented must be valid at the time it is transmitted or presented.

By valid, we assume it is the latest data available to the identity/attribute provider, although this information may always be outdated (ex: very recent address change). The main goal of this principle is to always use fresh data coming from the identity/attribute provider for each request/transaction, and not using some cached data or long-lived data retrieved a few days, or even a few minutes ago.

P3 Privacy – Minimal Disclosure

Personal information revealed to an entity should be the minimal needed for the purpose of the service provided.

Ex: To check that someone is older than 18, a SP should not ask his birth date, but a question "*Is the citizen older than 18?*"

As a particular case, personal identifiers should be kept to the minimum needed. This should be treated as a special case because of very strict legal limitations related to national identifiers in some countries. A country-level policy must thus allow the following possibilities related to personal identifiers:

- If identifier is not needed, it should not be transmitted (ex: SP limited to adults > 18)
- Restricted to the country, sector, usage, institution, or application using it
- Not linkable to the real identity unless needed
- Maybe linkable to the real identity only by originating country official instances (government, justice, …)
- Anyway, at least one identifier received by a SP is supposed to be persistent; that is, whenever a user logs on to the SP, the same identifier will be sent for the whole citizen's life. The case where no persistent identifier is provided will be treated as a special case.

P4 Auditing

For the situations where it is intended, it must be possible to audit the system to trace fraudulent transactions. Note that not all types of requests require that, and some policies may be implemented to allow this only to some specific instances (Justice …).

P5 Client compromising mitigation

Client PC compromising is user's sole responsibility. STORK solution does not endorse the responsibility for securing the user's PC, and requires a healthy, correctly installed, and secure client to be used in a secure way. STORK solution cannot be secure if the user's PC is compromised.

However, when designing the solution, some measures may be applied in order to mitigate the impact of a user's PC, but this does not intend to solve any problem due to a client security hole.

6 Security objectives

These objectives are deducted from chapter 4 "**Threats**" and 5 "**Security Principles**". They describe the goals to protect against all threats and to comply with all principles.

6.1 Identity protection (Authentication)

O1 Entity identification

Any entity (system, machine, etc.) must be clearly identified in all levels of each layer (network, application, etc.).

O2 Protection against guessing

Any secret information (encryption key, session identifier, etc.) must be impossible to guess, either by understanding its syntax (sequential numbers, etc.), or by brute force attack.

O3 Transmitted data confidentiality

Any sensitive information (whether it is personal or technical/internal data) must be protected against eavesdropping by anybody but the intended communicating parties.

O4 Transmitted data integrity

No one must be able to modify any data transmitted between parties.

O5 Session protection

Any session must be protected against hijacking.

O6 Protection against replay

All communication must be secured against replay attacks using appropriate mechanisms, e.g. session tokens or nonces/MAC.

O7 Entity Authentication

Any entity (system, machine, etc.) must be authenticated. As a corollary, man-in-the-middle attacks must be impossible between the parties: user, SP, PEPS, IDP, AP.

Although STORK does not address the Member States specific protocols (user to SP/IDP/AP), STORK protocol may not introduce a possibility of MITM attack between the different parts of the workflow. Thus, it must be possible to integrate STORK protocol in a complete workflow where no MITM attack is possible between any parties.

O8 Just-in-time validity

Any information transmitted or presented must be valid at the time it is transmitted or presented.

6.2 Privacy protection

O9 User centric approach

See P1

O10 Minimal Disclosure

Personal information revealed to an entity should be the minimal needed for the purpose of the service provided.

O11 Awareness of privacy issues

Users must be presented with all needed information to clearly understand and decide which personal information will be transmitted, to whom, and the finality of the processing of these data.

O12 Usability of privacy protecting tools

Users must have an easy way to decide if they agree to send personal data to an entity, and, if relevant, which part of the requested data.

O13 User profile privacy

It should be impossible, for any entity of the STORK infrastructure, to profile users; that is, no entity should be able to re-trace a user's requests.

6.3 Accountability and User Control

O14 Identification

It must be possible to identify uniquely each user (citizen):

- One identifier must be linked to only one user
- In most cases, one user should also be represented by one life-long identifier.

O15 Non-repudiation

Except in cases where non-repudiation is explicitly removed as a requirement, e.g. because it is in conflict with privacy requirements, no entity should be able to deny an action it performed. The system should provide a trace of all actions taken by an entity (user) along the path (end to end).

O16 Auditing

It must be possible to find traces of all actions operated on the system, in order to trace any attack (successful or not).

O17 Audit protection

It must be impossible, for an attacker, to cover his tracks. Therefore, the access to the audit logs must be strictly controlled.

6.4 System implementation and operation

O18 Safe design and implementation

Design and implementation should be as robust as possible.

A list of categories of problems to avoid is given by NIST "*Common Weakness Enumeration*" (http://nvd.nist.gov/cwe.cfm).

Moreover, an evaluation of system components under international accepted criteria, e.g. Common Criteria, may help to increase system reliability and user confidence. Those evaluations also cover aspects of secure system initialisation, access control, and secure system operation (see also O19, O20, O25).

O19 Secure default parameterisation

The system should minimise risks of security holes in case of installation or configuration mistake or misunderstanding. That is, in case of misunderstanding or forgetting, the safest choices must be applied.

O20 Strict access control

Access to all resources must be strictly controlled.

O21 Robustness against invalid input

All input data must be normalised (decode) and strictly validated.

O22 Immunity against race condition

Potential race conditions must be identified, and the logic must be adapted accordingly.

O23 Robustness

The system should be protected against Denial of Service attacks, and all resource abuse attacks.

O24 Operational staff awareness and competences

All parties involved in the project should know what actions they may and may not perform, and how to react in unusual situations.

O25 Secure operation

The system must be operated in safe conditions. This encompasses administrators and operators access, deployment procedures, backups, key handling, etc.

O26 Client compromising mitigation

See P5

The following table shows the relations between threats and principles, and objectives.

The following symbols are used:

✓ This function protects against the threat (or implements a principle)

± This function may partially help to protect against the threat (or implement a principle), but not totally

	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10	O11	O12	O13	O14	O15	O16	O17	O18	O19	O20	O21	O22	O23	O24	O25	O26
T1	±	±	±	±	±	±	±											±	±	±	±	±		±	±	
T2	±	±	±	±	±	±	±											±	±	±	±	±		±	±	
T3				±		±	±											±	±	±	±	±		±	±	
T4			±		±		±		±	±	±	±						±	±	±	±	±		±	±	
T5													✓				±									
T6													✓													
T7											✓															
T8												✓														
T9															✓											
T10											±	±								±					±	
T11											±	±													±	
T12																✓										
T13																	✓		±		±	±	±			
T14	±	±			±														±	±	±	±			±	
T15																			±			±		±		
T16																							✓			
T17																			±	±	±	±	±	±	±	
T18																					±			±	±	
P1									✓																	
P2								✓																		
P3										✓																
P4														±	±	±										
P5																										✓

Table 2: Threats & Principles vs. Objectives

7 Best practices

These recommendations come from the security communities, and software producers and integrators; they are general practices any solution must follow in order to avoid security holes.

B1 Established technologies usage

Established and proven standards, protocols, algorithms, methods must be used.

"Security by obscurity" must be avoided.

B2 Positive security model

A "positive" security model (also known as "white list") defines what is allowed, and rejects everything else. This should be contrasted with a "negative" (or "black list") security model, which defines what is disallowed, while implicitly allowing everything else (typically, unknown things).

B3 Fail-safe

Security mechanisms must be designed so that a failure follows the same execution path as disallowing an operation.

B4 Defence in depth

Layered security mechanisms increase security of the system as a whole. If an attack causes one security mechanism to fail, other mechanisms may still provide the necessary security to protect the system.

B5 Simplicity

The solution must be as simple as possible. The more complex, the more difficult to assess the security, the more risk of a security hole.

B6 End-to-end security

End- to- end security must be applied within the entire system, which covers information flowing from the origin, i.e. a trusted Identity and Attribute Provider, to the destination, i.e. the Service Provider (see also P1 "User centric – info"). The end-to-end security includes all intermediate entities, which means that:

- The emitter of a request must authenticate to the receiver
- The receiver must authenticate to the emitter
- The receiver must check that the request is destined to him

B7 Encryption should be used by default

Even if not considered as needed for confidentiality reasons, authenticated encryption should always be used, unless there is a specific reason to not do that (like if the data has to be analysed by intermediary routers, etc.).

This ensures that no private data transits in plain text because someone uses an unencrypted channel, but also may disable some not yet envisioned attacks on the data, even signed ones.

8 Security functions

F1 Just-in-time validity


All information must be retrieved from the originating source. No information may be cached.

8.1 Accountability and User Control

F2 Strong authentication


Except for the user authentication, which is left to each Member State policy, all parties (servers) should be strongly authenticated. This authentication must be checked by the protocol at all needed places, and the authenticated information must displayed to the user when needed.

For users, it must be possible to identify uniquely each entity:

- One identifier must be linked to only one entity
- In most cases, one entity should also be represented by one life-long identifier.

Dependencies: F10, F12

F3 Auditing


All actions must be audited (logged). Personal data may not be logged in any way.

F4 Audit protection


Audit logs must be protected against tampering, deletion, and access by unauthorised people.

Dependencies: F3

8.2 Privacy protection

F5 Minimal Disclosure


Personal information asked by a SP must be the minimal one needed for its purpose. The user should also have the possibility to restrict the personal data requested by the SP, e.g. by disabling particular attributes to be transferred to the SP (if some are optional).

F6 Personal identifiers

In order to accommodate with some countries restrictions, and to avoid correlation of databases coming from totally different domains, it must be possible to derive an identifier unique to:

- a country
- a sector
- an organisation
- a sub-organisation (application, department, etc.)

F7 User interface – information

Users must be presented information needed to clearly understand and decide which personal information will be transmitted, to whom, and the finality of the processing of these data.

In the practice, at least the following must be presented:

- Each attribute or data (either the data meta-information or the data content)
- To whom information will be transmitted (organisation name, country, etc.)
- Where information was retrieved (optional?)
- Why information is requested

All presented data is supposed to be valid; thus, information about the parties (especially the SP) must be validated.

Furthermore, the user must always visualise, in an easy and effective way, which service and country he is interacting with, especially when being redirected from SP to PEPS, to IDP, to AP, …

In order to ease the user's understanding, all presented information should have a similar look and feel, adapted to the information content. For instance, the following should apply:

- Wording should be the same
- Order of presented data should be the same (ex: country, organisation name, logo, URL, etc.)
- Confirmation should be presented the same way (positive/negative questions, etc.)
- Standardised logos and pictograms should be promoted.

To avoid attacks aimed at confusing the user by displaying text that could be interpreted as one of the presented fields[1], all data should be presented as "structured", for instance in tables.

F8 User profiling

Central servers (PEPS) should not store which SP a user connects. This clearly conflicts with auditing requirements (F3).

Ideally, the solution could support a technique so that no central server (PEPS) even know which SP a user connects (unless the user or the SP discloses it). This will be studied in a later phase.

[1] See http://aviv.raffon.net/2008/01/02/YetAnotherDialogSpoofingFirefoxBasicAuthentication.aspx for an example of such an attack on Firefox

8.3 Design and implementation

F9 Safe design and implementation

Design and implementation should follow all design and development recommendations, like mechanisms protecting against SQL injection, XSS, etc.

F10 Strong keys and cryptographic mechanisms

The solution should only use strong cryptographic mechanisms, protocols, etc., together with strong keys.

F11 Secure defaults

Default values for all parameters must be the least privileged ones or the stricter ones. Every "opening" of a security feature must be explicitly set, as well for system configuration, as for requests parameters.

F12 Secure communication channels

All communications between parties must be encrypted and mutually authenticated.

Dependencies: F10

F13 Secure session handling

Session handling must be immune to session hijacking and replay.

F14 Input validation

All input data must be normalised (decode) and strictly validated.

F15 Protection against replay

All communication must be secured against replay attacks using appropriate mechanisms, e.g. session tokens or nonces/MAC.

F16 Race condition

Potential race conditions must be identified, and the logic must be adapted accordingly.

8.4 Implementation and operation

F17 Access protection

Access to all resources must be strictly controlled.

F18 Denial of Service

The system should be protected against Denial of Service attacks.

F19 Operational staff awareness and competences

All parties involved in the project should know what actions they may and may not perform, and how to react in unusual situations.

F20 System update

 All systems must be maintained to a secure level by applying the provider's patches.

F21 Life cycle

Software life cycle must follow strict procedures to ensure the needed stability

The following table maps the security functions to the identified principles.

The following symbols are used:

- ✓ This function entirely fills an objective
- ± This function partially fills an objective

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21
O1		✓																			
O2		✓										✓									
O3												✓									
O4												✓									
O5													✓								
O6															✓						
O7		✓																			
O8	✓																				
O9							±														
O10					±	±															
O11							✓														
O12							✓														
O13								✓													
O14		✓																			
O15		±	±																		
O16			✓																		
O17				±													±			±	
O18									✓	±				±							
O19											✓										
O20																	✓				
O21														✓							
O22																✓					
O23																		✓			
O24																			✓		
O25																				±	±
O26																					

Table 3: Objectives vs. Functions

The following table shows the relations between attacks and functions. This table allows checking that no attacks were left without mitigation.

The following symbols are used:

✓ This function protects against the threat (or implement a principle)

± This function may partially help to protect against the threat (or implement a principle), but not totally

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21
A1		✓																			
A2		✓								✓		✓									
A3												✓									
A4												✓									
A5													✓								
A6															✓						
A7										✓											
A8		✓																			
~~A9~~																					
A10								✓													
A11		±	±																		
A12			✓																		
A13				✓																	
A14									✓												
A15											✓										
A16																	✓				
A17									±					±						±	
A18																✓					
A19																		✓			
A20																			✓		

Table 4: Attacks vs. Functions

9 Security technical recommendations

For each security function, we detail a technical recommendation about one or several technologies, their implementation, or their operation. Although some other similar choices could be performed, we do not list here all other alternatives, although we may sometimes argument on the choice on one solution compared to another one.

9.1 Just-in-time validity

R1 SAML token

SAML token must have a validity period short enough (a few minutes) to forbid attacks (re-use, etc.).

9.2 Strong authentication

R2 Browser-server

All connections between a browser and any server must use HTTPS[2].

Although the user may not always use TLS client authentication (with a X.509 certificate), servers must always use the classical TLS authentication with a X.509 certificate recognised by the citizen's browser.

R3 PEPS-PEPS

Although no direct connections are currently envisioned between PEPS – as all requests transit through the browser – this could come at some time. In this case, HTTPS with 2-way authentication should be used[3].

R4 PEPS-IdP/AP

When direct connections are performed between the PEPS and an IdP/AP, a strong 2 way authenticated protocol must be used. HTTPS with 2-way authentication is a good candidate. In case both servers are in a secure environment (same data centre, VPN, etc.), this recommendation may sometimes be ignored.

R5 SP-PEPS

When direct connections are performed between the SP and the PEPS, a strong 2 way authenticated protocol must be used. HTTPS with 2-way authentication is a good candidate.

[2] See R16 for recommended TLS parameters

[3] IPSec could also be possible, although the set up is heavier

When requests transit through the browser, it is very important to authenticate the SP prior to send any personal information to it. Although the user will acknowledge sending the information, the PEPS must ensure that the actual URL where to send the information to actually corresponds to the intended institution. The most obvious solution is to extract the institution information from the certificate, to check if it matches the response URL, and to display it (at least the Common Name) to the user. In case a local "anonymising" client is used (in the future), it could do this instead of the PEPS. To get the institution certificate, there are, at least, two common ways:

- verification of the certificate from the request signature
- calling directly the response URL – which must be a HTTPS one

9.3 Auditing

R6 Auditing


All actions must be audited (logged). Personal data may not be logged in any way.

R7 Audit protection


Audit logs must be protected against tampering, deletion, and access by unauthorised people.

9.4 Privacy protection

R8 Minimal Disclosure


Personal information asked by a SP must be the minimal one needed for its purpose. The user should also have the possibility to restrict the personal data requested by the SP, e.g. by disabling particular attributes to be transferred to the SP (if some are optional).

R9 Personal identifiers

In order to accommodate with some countries restrictions, and to avoid correlation of databases coming from totally different domains, it must be possible to derive an identifier unique to:

- a country
- a sector
- an organisation
- a sub-organisation (application, department, etc.)

See [6] for an in-depth analysis and practical recommendations about algorithms, etc.

This point is highly recommended, even if no legislation obliges to do so, in order to enhance the general privacy of the STORK solution. Note that this solution is impossible to implement in a later phase without breaking the compatibility with the identifiers already registered by the SP.

R10 User interface – information

Users must be presented information needed to clearly understand and decide which personal information will be transmitted, to whom, and the finality of the processing of these data.

In the practice, at least the following must be presented:

- Each attribute or data
- To whom information will be transmitted (organisation name, country, etc.)
- Where information was retrieved (optional?)
- Why information is requested (if needed – may come from the SP itself, like a button with a pop-up …)

All presented data is supposed to be valid; thus, information about the parties (especially the SP) must be validated.

Furthermore, the user must always visualise, in an easy and effective way, which service and country he is interacting with, especially when being redirected from SP to PEPS, to IDP, to AP, …

In order to ease the user's understanding, all presented information should have a similar look and feel, adapted to the information content. For instance, the following should apply:

- Wording should be the same
- Order of presented data should be the same (ex: country, organisation name, logo, URL, etc.)
- Confirmation should be presented the same way (positive/negative questions, etc.)
- Standardised logos and pictograms should be promoted.

To avoid attacks aimed at confusing the user by displaying text that could be interpreted as one of the presented fields[4], all data should be presented as "structured", for instance in tables.

R11 User profiling

Central servers (PEPS) should not store which SP a user connects. This clearly conflicts with auditing requirements (R6).

Ideally, the solution should support a technique so that no central server (PEPS) even know which SP a user connects (unless the user or the SP discloses it). In order to support this, a local client is needed (or a trusted third-party service). This will be analysed in a later phase, when potential clients will be envisioned for other purposes (like SAML key binding).

[4] See http://aviv.raffon.net/2008/01/02/YetAnotherDialogSpoofingFirefoxBasicAuthentication.aspx for an example of such an attack on Firefox

9.5 Implementation and operation

R12 Safe design and implementation

Design and implementation should follow all design and development recommendations, like mechanisms protecting against SQL injection, XSS, etc.

Some practical recommendations (non-exhaustive) are:

- All input data must be normalised (decode) and strictly validated before entering the business logic of any component
- All data must be adequately encoded before being transmitted to any external party. A typical example is to encode in HTML all data being displayed in a Web application
- Only parameterised SQL/XPath queries may be used, no dynamic queries

R13 Secure defaults

Default values for all parameters must be the least privileged ones or the stricter ones. Every "opening" of a security feature must be explicitly set, as well for system configuration, as for requests parameters.

R14 Race condition

Potential race conditions must be identified, and the logic must be adapted accordingly.

R15 Private key operations

All private keys should be protected in a Hardware Security Module with state of the art access control. An accreditation (EAL 4+) for this environment is highly recommended.

9.6 Strong keys and cryptographic mechanisms

R16 TLS parameters

The solution should only use strong cryptographic mechanisms, protocols, etc., together with strong keys.

Typical strong parameters supported by browsers are

- Protocol: TLS, SSLv3
- Authentication: RSA, DSS
- Key exchange: (Enhanced) Diffie-Hellmann, RSA
- AES, 3DES, IDEA with min. 128 bits
- MAC: SHA-1

R17 Secure communication channels

All communications between parties must be encrypted and mutually authenticated.

R18 Secure session handling

Session handling must be immune to session hijacking and replay.

R19 Access protection

Access to all resources must be strictly controlled.

The system should be protected against Denial of Service attacks.

R20 Social engineering

All parties involved in the project should know what actions they may and may not perform, and how to react in unusual situations.

R21 System update

 All systems must be maintained to a secure level by applying the provider's patches.

R22 Life cycle

Software life cycle must follow strict procedures to ensure the needed stability

The following table maps the security functions to the identified principles and objectives. Threats that are out of scope are not represented here. The following symbols are used:

- ✓ This function entirely fills an objective
- ± This function partially fills an objective

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22
B1																						
B2												✓										
B3												✓										
B4		±	±	±	±																	
B5																						
B6		±	±	±	±																	
B7		✓	✓	✓	✓																	
F1	✓																					
F2		✓	✓	✓	✓																	
F3						✓																
F4							✓															
F5								✓														
F6									✓													
F7										✓												
F8											✓											
F9		±	±	±	±																	
F10		±	±	±	±										±	±	±	±	±	±		
F11													✓									
F12		±	±	±	±							✓										
F13																		✓				
F14												✓										
F15	±	±	±	±	±																	
F16														✓								
F17												±			±				✓			
F18																			✓			
F19																				✓		
F20																					✓	
F21																						✓

Table 5: Recommendations vs. Functions & Best practices

10 Appendix: Security tables

10.1 Table of attacks

A1 Spoofing 14
A2 Guessing 14
A3 Communication eavesdropping 14
A4 Communicated data tampering 14
A5 Session hijacking 14
A6 Replay Attack 14
A7 Echo Attack 14
A8 Man-in-the-middle Attack 15
A9 Skimming 15
A10 User profiling 15
A11 Action repudiation 15
A12 Attack without trace 15
A13 Attacker covers trace 15
A14 Incorrect design and implementation 15
A15 Incorrect Usage (Parameterisation) 15
A16 Unauthorised access 16
A17 Fuzzing 16
A18 Race condition 16
A19 Denial of Service Attack 16
A20 Social engineering 16

10.2 Table of security threats

T1 Impersonation of a citizen 17
T2 Impersonation of system 17
T3 Identity data forge 18
T4 Privacy – user data 18
T5 Privacy - trail 18
T6 User profiling 18
T7 Unawareness of privacy issues 18
T8 Usability of privacy protecting tools 19
T9 Denial 19
T10 User – Accidental misuse 19
T11 User – Forced misuse 19
T12 Log missing 19
T13 Log forged 19
T14 System compromise 20
T15 System malfunction 20
T16 System Denial of Service 20
T17 System availability 20
T18 Operation security 20

10.3 Table of security principles

P1 User centric – info 22
P2 Just-in-time validity 22
P3 Privacy – Minimal Disclosure 23
P4 Auditing 23
P5 Client compromising mitigation 23

10.4 Table of security objectives

O1 Entity identification 24
O2 Protection against guessing 24
O3 Transmitted data confidentiality 24
O4 Transmitted data integrity 24
O5 Session protection 24
O6 Protection against replay 24
O7 Entity Authentication 24
O8 Just-in-time validity 24
O9 User centric approach 25
O10 Minimal Disclosure 25
O11 Awareness of privacy issues 25
O12 Usability of privacy protecting tools 25
O13 User profile privacy 25
O14 Identification 25
O15 Non-repudiation 25
O16 Auditing 25
O17 Audit protection 25
O18 Safe design and implementation 26
O19 Secure default parameterisation 26
O20 Strict access control 26
O21 Robustness against invalid input 26
O22 Immunity against race condition 26
O23 Robustness 26
O24 Operational staff awareness and competences 26
O25 Secure operation 26
O26 Client compromising mitigation 26

10.5 Table of security best practices

B1 Established technologies usage 28
B2 Positive security model 28
B3 Fail-safe 28
B4 Defence in depth 28
B5 Simplicity 28
B6 End-to-end security 28
B7 Encryption should be used by default 28

10.6 Table of security functions

F.1 Just-in-time validity....29
F.2 Strong authentication....29
F.3 Auditing....29
F.4 Audit protection....29
F.5 Minimal Disclosure....29
F.6 Personal identifiers....30
F.7 User interface – information....30
F.8 User profiling....30
F.9 Safe design and implementation....31
F.10 Strong keys and cryptographic mechanisms....31
F.11 Secure defaults....31
F.12 Secure communication channels....31
F.13 Secure session handling....31
F.14 Access protection....31
F.15 Input validation....31
F.16 Race condition....31
F.17 Denial of Service....31
F.18 Operational staff awareness and competences....32
F.19 System update....32
F.20 Life cycle....32

10.7 Table of security recommendations

R1 SAML TOKEN....35
R2 BROWSER-SERVER....35
R3 PEPS-PEPS....35
R4 PEPS-IDP/AP....35
R5 SP-PEPS....35
R6 AUDITING....36
R7 AUDIT PROTECTION....36
R8 MINIMAL DISCLOSURE....36
R9 PERSONAL IDENTIFIERS....36
R10 USER INTERFACE – INFORMATION....37
R11 USER PROFILING....37
R12 SAFE DESIGN AND IMPLEMENTATION....38
R13 SECURE DEFAULTS....38
R14 RACE CONDITION....38
R15 PRIVATE KEY OPERATIONS....38
R16 TLS PARAMETERS....38
R17 SECURE COMMUNICATION CHANNELS....39
R18 SECURE SESSION HANDLING....39
R19 ACCESS PROTECTION....39
R20 SOCIAL ENGINEERING....39
R21 SYSTEM UPDATE....39
R22 LIFE CYCLE....39

11 References

[1] Modinis Study on Identity Management in eGovernment: "*Common Terminological Framework for Interoperable Electronic Identity Management*", November 23, 2008.

[2] STORK "*Glossary v2.0*", 2008.

[3] "*Quality Authenticator scheme*", STORK "*Deliverable 2.3 v.1.0*", 2009.

[4] "*Technology – induced challenges in privacy & data protection in Europe*", ENISA AdHoc Working Group on Privacy & Technology, October 2008.

[5] Common Criteria - http://www.commoncriteriaportal.org/

[6] "*National Identifier privacy*" , STORK WP5, 2009.