COMPETITIVENESS AND INNOVATION FRAMEWORK PROGRAMME
ICT PSP Fifth Call for proposals 2011 - Pilot Type A

Towards a single European electronic identification and authentication area

ICT PSP call identifier: CIP-ICT-PSP-2011-5
ICT PSP Theme/objective identifier: 4.2

Project acronym: STORK 2.0
Project full title: Secure idenTity acrOss boRders linKed 2.0
Grant agreement no.: 297263

D4.12 Final version of security recommendations

Deliverable Id :	**D4.12**
Deliverable Name :	**Final version of security recommendations**
Status :	**Final**
Dissemination Level :	**PU**
Due date of deliverable :	**July 31st, 2015**
Actual submission date :	**August 11th, 2015**
Work Package :	**WP4**
Organization name of lead contractor for this deliverable :	**Approach**
Author(s):	**Marc Stern, John Heppe**
Partner(s) contributing :	**Approach, ARGE, ES-UJI, Cassidian, NL-V&M, UAegean, LuxTrust, ATOS, MINHAP**

Abstract: This document aims at the description of security requirements that have to be fulfilled by the interoperability layer developed in the STORK 2.0 project. This deliverable is an update of D4.5 and contains the final version of the security recommendations.

Project co-funded by the European Community under the ICT Policy Support Programme

History

Version	*Date*	*Modification reason*	*Modified by*
0.0	18/12/2014	D4.5 used as a template	
0.1	15/07/2015	Added implemented solution(s) to the first version of the security recommendations (D4.5)	Marc Stern
0.2	05/08/2015	Quality check	ATOS
0.3	11/08/2015	Corrected document information and style. Added section 6 “Code security”. Extended management summary.	Marc Stern
0.4	11/08/2015	Final quality check	ATOS
FINAL	11/08/2015	Final deliverable	

Table of contents

List of figures

List of tables

List of abbreviations

AAS	Attribute Aggregation Service
AP	Attribute Provider
A-PEPS	The PEPS role for attribute collection in foreign countries (not the citizen’s country)
AQAA	Attribute Quality Authentication Assurance
AUB	Authentication on Behalf
C-PEPS	The PEPS role to attend national Citizens
CSRF	Cross-site request forgery
DB	Database
eID	Electronic Identity
IdP	Identity Provider
MS	STORK 2.0 Member State
MW	Middleware
PEPS	Pan European Proxy Server
SP	Service Provider
S-PEPS	The PEPS role to attend SP requests
SSL	Secure Sockets Layer (old version of TLS)
SSO	Single Sign-On
STORK 2.0	Secure idenTity acrOss boRders linKed 2.0
TLS	Transport Layer Security
UID	User Identifier
V-IDP	Virtual Identity Provider

Executive summary

This deliverable is an update of D4.5 and contains the final version of the security recommendations. This deliverable proposes some security recommendations related to session handling, e-Identifier coupling (attribute collection from multiple attribute providers for a single user) and attributes aggregation (complexity of interaction and cross-border, cross-entity (SP, AAS, IdP & AP) communication and authentication on behalf of.

Several aspects are relevant to MS-specific functionalities. When applicable to the common code, even to the code provided as example, the chosen implementation was detailed. Here is a summary:

- Session management is reduced to the minimum to not introduce any security risk.
- The solution chosen for e-Identifier coupling solves the problem is the majority of the cases, leaving the responsibility to the SP to use imperfectly matched identifiers.
- Attribute Aggregation being totally out of the scope of the common code, no common decision was taken about its implementation.
- All technical means recommended to protect Authentication on Behalf of were implemented in the common code; these need to be completed with MS-specific one (physical access control, operation processes, audit, etc.).

On top of that, an extensive security review of the PEPS code was performed by an independent team and most of the important remarks were agreed to be implemented. No high security issues should remain; this will be verified before the end of the project, and reported in D4.14 Final version of code quality review. An overview of the result is given in section 6.

1 Introduction

This deliverable is an update of D4.5 and contains the final version of the security recommendations. This report is published by the STORK 2.0 security group in the context of WP4. This document describes some security topics, analyses them and may propose some security recommendations but they are not considered as official recommendations, neither for the core development, nor for the Member States.

STORK 1 already produced several security-related documents that are listed in the table below. Most of them stay relevant for STORK 2.0.

Document name	***Description***
D5.8.3d Security Principles and Best Practices	STORK 1 Security recommendations (https://www.eid-stork.eu/index.php?option=com_processes&Itemid=&act=streamDocument&did=1878) This document is an official deliverable from STORK 1. It is applicable to the common development, as well to the MS-specific development and deployment. Although some aspects may be added and/or refined for STORK 2.0, this document remains applicable.
National Identifier privacy	This internal document describes mechanisms to obfuscate a national identifier in order to respect privacy.
SAML Key Binding	This internal document describes the Man in the Middle attack possibilities on the SAML protocol, and the new SAML profile “Holder of key browser SSO” aimed at solving this.
STORK 2.0 D4.7 Code Quality Review	Analyse performed by an independent team (from Approach) of the general quality of the code and the in-depth security aspects. Detailed recommendations were performed and a verification of the code modification will be performed before the end of the project.

Table 1: STORK 1 security documents

This document has been organized in the following way:

Section 2 provides an overview of the problems related to session handling.

Section 3 highlights problems linked to attribute collection from multiple attribute providers for a single user.

Section 4 identifies the problems associated to the complexity of interaction and cross-border, cross-entity (SP, AAS, IdP & AP) communication.

Section 5 describes problems linked to the process of “Authentication on Behalf”.

Section 6 introduces the main results of the code security analysis.

2 Session handling

Session management as generally managed – including in STORK1 – may introduce functional problems and security vulnerabilities. This section highlights the potential problems linked to section handling, describes the attached threats and proposes some measures to mitigate them. Although performed in the framework of STORK 2.0, all findings can be applied to STORK1 without any changes.

The section begins by describing the problems. It will then go on to the multiple e-Identifier coupling approaches. Finally the proposed solution (to be filled in) is introduced, and the implemented solution is described.

2.1 Functionalities

In order to remember some information between the browser calls, applications usually store the needed data inside a "session" associated to a user/browser. Each session is associated to an identifier ("sessionid") allowing retrieving the right user's session.

Applications sessionid is usually stored in a HTTP cookie or written inside the URL.

Sessions are heavily used in STORK because the browser connects first to the PEPS/VIDP, then authenticates on other national systems and finally comes back to the STORK PEPS/VIDP. After the second connection, the PEPS/VIDP needs to retrieve some data from the first connection – mainly the SP URL to send the response to.

Several problems related to session management are listed below. Most aspects are security-related but we included some other ones as they potentially impact the possible solutions.

[F1] Sessionid in cookie

When storing the sessionid inside a cookie, the user may potentially be attacked by a mechanism called Cross-Site Request Forgery (CSRF)[1]: An attacker may force the user of a web application to execute actions of his choice without the user being aware of it as the cookie is sent automatically with any request to the intended server.

In case of a shared computer, the session cookie may be available for the next user if the browser is not (totally) shut down.

[F2] Sessionid in URL

When storing the sessionid inside the URL, the sessionid is visible on the user's screen and can be seen by a person passing behind the user. With modern smart phones, almost anybody can easily take a picture of the URL with the sessionid. The sessionid could also potentially be exposed in the "Referer" HTTP header.

In case of a shared computer, the sessionid may be available in the history to the next user.

[F3] Parallel sessions

When the sessionid is transmitted into a cookie, it may be overwritten by another one if the user connects another STORK-enabled service with the same browser. This could potentially lead to a session mismatch and thus the wrong information sent to a Service Provider.

[1] https://www.owasp.org/index.php/Cross-Site_Request_Forgery_%28CSRF%29

[F4] Load-balancing

If you have several servers in parallel to serve the incoming requests, you have no guarantee that the second request arrives on the same server as the first one. The session initiated on one server will thus be unavailable on another one.

2.2 Threats

Several potential threats related to session management have been identified:

[T1] Session re-use

In case a computer is shared, a user could potentially retrieve the previous user's session by having access to a stored cookie or a sessionid in the URL in the browser history. By replaying the last STORK HTTP request, he would potentially receive the response (aimed at the SP) containing the personal information; he probably will not succeed in impersonating the previous user at the SP's as the user already passed the authentication process but this should be envisioned. The impact of this attack is very high.

[T2] Session hijacking

During a STORK request, an attacker could grab the sessionid of a user performing an authentication (or other service) and send the "final part" of the request with the victim's sessionid. He would, in this case, receive the response (aimed at the SP) containing the personal information and he could impersonate the victim at the SP's.

Although this kind of attack is not easy because it has to be synchronised with the victim's requests, its impact is very high.

[T3] CSRF

During a STORK request, an attacker could manage to have the victim loading a URL pointing to a STORK URL and executing an unintended request. Although trivial to implement, this attack will only lead to have *the victim himself* executing an unintended action. The main impact we see at this moment would be to potentially break the authentication flow which would lead to an annoyance but no security risks (as the result of any action will only be sent to the expected user).

[T4] Replay

Replay is a form of attack where a malicious entity repeats previously intercepted messages. Although we do not see much impact of this kind of attack, we could imagine some scenarios where a request is served without interaction and leads to requests sent to an IdP/AP or responses sent to a SP; if a high number of requests is sent, this could be used as a Denial of Service entry point for an IdP/AP or a SP.

[T5] Flow break

In case several servers are used in parallel and the browser does not come back on the same server as the initial request, if the session are not synchronised between servers, the server will not find the session corresponding to the sent sessionid and will end-up with an error. This would be an annoyance for the user but would not introduce any security risk. However, this problem may occur on a majority of requests.

[T6] SP mismatch

In case a user opens several STORK transactions in parallel, it is possible that the information aimed at one SP is sent to the wrong one. This could lead to personal information leakage.

The following table shows the mapping between problems and threats:

	***[F1] Sessionid* in cookie**	***[F2] Sessionid* in URL**	***[F3] Parallel sessions***	***[F4] Load-balancing***
[T1] *Session re-use*	X	X		
[T2] *Session hijacking*		X		
[T3] CSRF	X			
[T4] Replay				
[T5] *Flow break*			X	X
[T6] *SP mismatch*			X	

Table 2: Mapping between problems and threats

2.3 Possible solutions

Several possible solutions for the identified threats are listed below:

[S1] Delete session server-side

As soon as the SAML response is created – even before sending it – the current session and all associated must be destroyed on the server. This will forbid any re-use of the session even if the client knows the used sessionid.

Note that destroying the session cookie on the browser is not sufficient.

[S2] Sessionid in URL

Putting the sessionid in the URL eliminates the problems linked to the session cookies although we saw that it introduces other ones (eavesdropping).

[S3] Sessionid in cookie

Putting the sessionid in a cookie eliminates the problems linked to the URL although we saw that it introduces other ones (CSRF).

[S4] Session sharing

Several mechanisms are available to share the sessions between servers:

1. Share memory: a synchronisation of the memory caches is performed between the servers.
2. Session in DB: the sessions may be stored in a shared DB. In case a DB is available, this is usually not a problem although it adds a little overhead; in case no DB is available, this solution is very heavy.

Both techniques render the system more complex to manage.

[S5] Session stickiness

Some mechanisms are available to ensure that one browser always comes back to the same server. You mainly have 3 possibilities to do that:

1. Statically distribute the IP addresses between the servers. This mechanism has a lot of drawbacks (efficiency, does not support changing IP addresses of mobile devices, etc.)
2. Tie the TLS session to a server. As the TLS session may change – especially in STORK where the browser connects the PEPS/VIDP, disappears for a few minutes, then comes back
3. Tie the sessionid to a server. This is only possible if the TLS tunnel is stopped before the application server otherwise the sessionid is encrypted with the whole request.

Session stickiness is definitely possible but is not always easy to support and renders the load-balancing much less dynamical than what is usually expected by system/network administrators.

[S6] Anti-replay mechanism

In order to forbid replay attacks, a mechanism has to be implemented to check, for every new SAML request, if the request id was not already served. A table with all served request id must thus be kept – note that id corresponding to expired requests may be purged from this table.

In case load-balancing is performed, a synchronisation mechanism equivalent to [S4] must be provided; for synchronisation at application level, a DB is probably the easiest solution.

[S7] Check SAML request id in session

In order to ensure that no mismatch can occur between two parallel requests, the SAML request id could be stored in the session and checked when coming back from the IdP. In order to perform that, the IdP either needs to use the request id as returned identifier[2]. In case a mismatch is detected, the process can be stopped to forbid any personal information leakage but this will not be very user-friendly.

[S8] Use SAML request id

The usual session handling uses a mechanism where the server generates a sessionid and stores it in a place where it is sure to retrieve it later – typically a cookie or the URL.

In STORK, because we use SAML requests only (at least in the interfaces, not especially in the back-ends), we always get an identifier: the SAML request id. We could use this identifier as a session key and transport it in the requests payload only – thus eliminating the need to store it in a cookie or the URL.

In case load-balancing is needed, a shared resource between the servers is required, as for [S6], a DB is probably the easiest solution. The same table could be used to maintain the mapping between a request id and the SP URL to send the response to and to check against replay attacks.

[S9] Anti-CRSF tokens

Specific anti-CRSF solutions exist, using “tokens” to enforce a specified workflow. This totally blocks CSRF attacks but imposes a lot of constraints to the whole flow. As the user leaves the STORK infrastructure between the first and last calls, this would impose a lot of constraints to intermediate services (IdP, AP, etc.).

[2] or a mapping between the IdP identifier and the request id has to be maintained (possibly in the session as this is for a check only)

The following table shows the mapping between threats and possible solutions:

	[T1] Session re-use	[T2] Session hijacking	[T3] CSRF	[T4] Replay	[T5] Flow break	[T6] SP mismatch
[S1] Delete session server-side	X					
[S2] Sessionid in URL			X		X	X
[S3] Sessionid in cookie		X				
[S4] Session sharing					X	
[S5] Session stickiness					X	
[S6] Anti-replay mechanism				X		
[S7] Check SAML request id in session	X	X				X
[S8] Use SAML request id	NA	X	X		X	X
[S9] Anti-CRSF tokens			X			

Table 3: Mapping between threats and possible solutions

In order to protect against all threats, we have the following possible combinations:

- [C1] = [S2] + [S4] or [S5] + [S6] + [S7]
- [C2] = [S3] + [S4] or [S5] + [S6] + [S7] + [S9]
- [C3] = [S6] + [S8]

Remarks:

- [S2] and [S3] are mutually exclusive
- [S4] and [S5] are equivalent in their result
- [S6] and [S8] should share the same storage
- [S1] is not really needed but it is a best practice anyway (if sessions are used)

2.4 Conclusions security related to session handling

Because a table should be implemented to protect against replay attacks [S6], we propose to extend it to replace usual sessions by a more limited mechanism [S8] solving most security problems linked to usual session handling. As the implementation would use the same storage, the overhead – both in performance and work-load – should be very low.

2.5 Implemented solution(s)

Sessions are maintained using cookies with the "*HttpOnly*" parameter, thus mitigating the threat of being used by JavaScript for session hijacking.

Parallel sessions are not supported. If a user enters in a new session in the PEPS, the previous one is overwritten in the browser, and thus not usable anymore.

No special measures have been taken for load balancing. Load balancers should implement stickiness, preferably – but not limited to – based on the session cookie.

3 e-Identifier coupling

In STORK1 user attributes are retrieved mainly from a central authority (usually the IdP). Due to pilot requirements, STORK 2.0 will support attribute collection from multiple attribute providers for a single user in a single session. This poses the problem of coupling a user's identifiers at different Identity and Attribute Providers. An e-Identifier subgroup has been established within STORK 2.0 WP4 Security group in order to examine the issue. The objectives of the subgroup are:

- to review and analyse existing e-Identifier coupling approaches and put forward new ones;
- to propose a solution that may be implemented in STORK 2.0.

Requirements of a possible solution for e-Identifier coupling include security and user privacy.

This section first identifies the problem. Then it describes multiple e-Identifier coupling approaches, and finally the proposed solution is introduced.

3.1 The problem and risk

3.1.1 The problem

The problem of e-Identifier coupling arises when a user wishes to retrieve attributes that are stored in different APs. In order to validly combine the attributes from the different APs it should be assured that the attributes retrieved at each AP are for the same user (physical person). That is, in a more technical way, a way to verify that the identifiers that each AP uses to identify the user refer to the same user (physical person) is necessary.

The problem arises both within and across borders. Within borders, there are different APs that store user information under different identifiers. For example, Universities use their own identifiers, tax offices use the fiscal number, and local government may use National Id or Passport number. Across borders the problem arises almost any time that a user wishes to retrieve attributes that are stored in APs at different countries. In this case it is almost certain that the APs will use different identifiers to refer to the user.

An obvious solution to the problem would be the use of common (shared) identifiers across the APs. This could for example be the STORK eID, or any other number that uniquely identify a person. Nevertheless, the majority of APs do not currently use such common identifiers, and we expect this to hold at least for the coming years. As a result it is necessary to study the problem and develop methods to address it.

3.1.2 Use cases

This sub-section describes the scenarios where e-Identifier coupling is required. In the typical scenario a user holds accounts at least to one IdP and several APs in which he can authenticate electronically. The IdPs and APs do not share a common user identifier. Table 4 summarises the preconditions of the typical e-Identifier coupling scenario:

ID	*Description*
EIC-PRE-1	The user holds an account to at least one IdP that he can access electronically in order to authenticate.
EIC-PRE-2	The user has identity attributes to at least one AP, and the attributes can be retrieved electronically without authentication mean.
EIC-PRE-3	The user holds an account to each AP that he/she has attributes, in which he can authenticate in order to retrieve them.

Table 4 : Preconditions for e-Identifier coupling

3.1.3 Threats

Three threats in e-Identifier coupling are identified:

1. Attribute impersonation: The intentional retrieval of another person's attributes.
2. Attribute combination: The intentional combination of two (or more) persons' attributes in one identity. It arises when two (or more) persons intentionally work together when authenticating in APs and couple their identifiers.
3. User tracing/profiling: The combination of a user's attributes stored in multiple APs and its (mis)use by a third party without the consent of the user.

3.2 Possible solutions

In this section a number of e-Identifier coupling approaches are discussed.

3.2.1 User defined coupling at third party

User defined coupling has been proposed in a recent model of attribute aggregation [3] [4]. In this model the user establishes links between disparate accounts, and he/she may link (couple) identifiers at multiple APs. The creation and management of links is handled by a third party service. To link two accounts at different APs and couple the identifiers, the user needs to sequentially authenticate at each AP. The system then maintains the coupling of the identifiers at the two APs and makes them available for future attribute retrieval.

This approach may be easily implemented. However, it does not guarantee that the coupled identifiers refer to the same person; attribute combination risk. Matching basic attribute information (e.g. name, surname, date of birth) would probably provide a sufficient assurance. In addition, the third party service may easily profile the user; user tracing profiling risk. Therefore, this model requires a trust relationship between the user and the service that the latter will not misuse the user's attribute information.

Please note that this approach could also be implemented at the MS-part of C-PEPS instead of a third party. This may be advantageous for the MS and STORK as it will facilitate attribute collection from all APs in the country. However, as the method requires persistent storage of information about the location of users' attributes MSs may decide not to store this information at the PEPS.

3.2.2 Core Id User defined coupling at AP

In this approach a single, core ID is used in order to couple identifiers at multiple APs. Each AP's identifier is linked to the core ID (this could be the STORK eID in our case). To link an account, the user authenticates in the same session in STORK and the AP, and the AP stores the STORK eID (or a derivation) in its systems.

This approach requires APs to store some additional information at their systems and may therefore have a limited application. In addition, it does not guarantee that the coupled identifiers refer to the same person; attribute combination risk - two (or more) persons can work together when authenticating. Matching basic attribute information (e.g. name, surname, date of birth) would provide a higher assurance. Finally, the approach does not exclude user tracing/profiling; APs may combine user information through the use of the core ID. Therefore, the approach requires a trust relationship between the user and the APs that the latter will not misuse user's attribute information.

3.2.3 Core Id User defined coupling at third party

This approach combines features from the two before. A core ID is used to couple identifiers at multiple APs, and each AP's identifier is linked to the core ID. However, the core ID (or the derivation) is not stored at the AP. The AP is requested to provide a persistent pseudonym that will be used for identifying the user and a third party stores the coupling of the pseudonym to the core ID (in our case the STORK eID).

This approach is easier to implement than the last one ("Core Id User defined coupling at AP") as it does not require APs to store additional identifiers. Other than that the two approaches share the same advantages and disadvantages.

Please note that this approach could also be implemented at the MS-part of C-PEPS instead of a third party. This may be advantageous for the MS and STORK as it will facilitate attribute collection from all APs of the country. However, as the method requires persistent storage of information about the location of users' attributes and pseudonyms MSs may decide not to store this information at the PEPS.

3.2.4 AP face-to-face

This approach requires the physical presence of the user at the AP for registration with high assurance. The user has to natural present at the AP with his/her ID card, and the AP, after checking the picture from the id (and possible other documents as well as a STORK authentication), links the ID in the card with the user identifier in its systems. In this way there is a high level of assurance for the coupling of identifiers. Subsequent attribute retrievals do not require the physical presence of the user.

This approach promises a higher level of security but requires additional user and AP action. As a result its application may be limited. In addition it does not exclude user profiling; APs may combine attribute information through the ID card details. Additional checks are needed in order to examine whether the approach promises perfect security.

3.2.5 Third party face-to-face

In this approach a third party is responsible for coupling user identifiers (either between different APs, or an AP and the STORK ID). The user has to physically present at the third party along with ID cards and possible other documentation that verifies his/hers identifiers at the APs and core IdP. An employee/representative of the third party checks the ID cards and documentation and inserts the coupling of the identifiers in the system. The physical presence of the user is required only once and for registration with high assurance, subsequent attribute retrievals do not require physical presence.

This approach promises a high level of security and requires less action from the user and APs compared to the "AP face-to-face" approach (note that the user can link multiple identifiers with one visit). However, the approach requires a third party with physical presence. In addition the third party can trace and profile the user information. Therefore, this model requires a trust relationship between the user and the service that the latter will not misuse the user's attribute information.

3.2.6 Possible solutions summary

In Table 5 the main risks for each approach are summarized. The matching of basic attributes provides a higher assurance for each method, but not a definite one.

	Attribute impersonation	***Attribute Combination***	***User tracing/profiling***
User defined at third party	✓	✘	✘
Core Id user defined at AP	✓	✘	✘
Core Id user defined at third party	✓	✘	✘
AP face-to-face	✓	✓	✘
Third party face-to-face	✓	✓	✘

Table 5: e-Identifier coupling Solutions versus threats

In Table 6 functional and technical features of each approach are summarized.

	User defined at third party	***Core Id user defined at AP***	***Core Id user defined at third party***	***AP face-to-face***	***Third party face-to-face***
Third party (or C-PEPS)	✓	✘	✓	✘	✓
Requires core-Id	✘	✓	✓	✘	✘
Storing additional information at AP	✘	✓	✘	✓	✘
Requires natural presence	✘	✘	✘	✓	✓
Security: Risk of two people authenticating together	✓	✓	✓	✘	✘
High assurance that matching is correct when no cheating occurs	✘	✘	✘	✓	✓
Security: Risk of attacks on matching	High	Moderate	Moderate	Low	Low
Citizen privacy	High	Moderate[2]	Moderate[2]	Low	Low
Notes	[1] Requires trust to the third party [2] With the use of pseudonyms				

Table 6 : e-Identifier coupling Solutions features

In summary, approaches that require physical presence of the person handle both attribute impersonation and combination risks efficiently, while approaches with no physical presence do not; they handle efficiently only the attribute impersonation risk. The matching of basic attributes in the latter provides a higher assurance for each one of these methods, but not a definite one. On the other hand, the natural presence on the former requires significant additional user and AP (or third-party) action. As a result their application may be limited.

Finally, none of the approaches handles the user tracing/profiling risk. In all of them, there is a possibility of a third party or the APs combing the user information. However, not all approaches require the same trust level. “User defined at third party”, “Core ID user defined at third party”, and “Third party face-to-face” requires a trust relationship between the user and the third party. On the other hand, “Core ID user defined at AP” and “AP face-to-face” requires a trust relationship between the user and APs. The former trust relationship, between user and third party, may be easier to establish as it involves two parties, in contrast with the latter that involves many APs.

3.3 Conclusions security related to e-Identifier coupling

The solutions that cover both attribute impersonation and attribute combination risks – that is "AP face-to-face" and "Third party face-to-face" – require an effort from the AP and citizens. Nevertheless, they enhance security and they are the preferable ones for implementing; especially the "Third party face-to-face" that simplifies the trust relationship for handling the third risk, "User tracing/profiling".

However, due to the relevant costs they may have limited applicability, and APs may implement the rest of the solutions. In this case the SP should clearly be informed that the aggregation of attributes has been performed in a lower level of assurance, and that re-authentication has taken place in the case it has.

Of these solutions the preferable appears to be the "Core Id user defined at third party" that handles attribute impersonation risk, has a moderate risk on attacks on matching (in comparison to "User defined third party approach"; see Table 6), simplifies the trust model for handling the user tracing/profiling risk, and requires limited action from APs that may enhance its applicability (in comparison to "Core Id user defined at AP"; see Table 6).

3.4 Implemented solution(s)

Communication between the C-PEPS/A-PEPS with its SPs, IdPs and APs is out of scope for the common parts of STORK2.0. The use of e-Identifier at the A-PEPS as a search mechanism is implemented – the e-Identifier is transmitted from the C-PEPS to the A-PEPS – but the national solution use of this identifier is under the MS responsibility.

However, the demo implementation – that could serve as an example for several MS – implements the following mechanism:

a. The PEPS sends in all requests to the AP the STORK eID, the given name, the surname and the date of birth

b. The matching is performed this way:

   1. The AP searches first the STORK eID in its databases (with a substitution of the destination country in the prefix). This will find users added through the STORK1 platform, if the eID isn't encrypted
   2. The AP searches the STORK eID using the "stripped" eID without the prefix. This will find users added in a manual process with user's presence
   3. The AP searches a match on the combination of the given name, the surname and the date of birth
   4. If no match is found, the AP re-authenticates the user, using its standard mechanisms

c. In case of the identity matching is not ensured (b.3 & b.4), no AQAA is returned and the retrieved given name, surname and date of birth are returned to allow the SP to validate the data – possibly with the "similarity of names" module provided in the common code.

4 Attribute Aggregation

The *attribute aggregation* (referred also as *smart attributes)* is one of the core functionalities of STORK 2.0 workflow, contributing significantly to its core business value. From that standpoint, it is important to consider the issues and consequences in the security and privacy of that process. This section analyses some potential problems linked to attribute aggregation and suggests some solutions which should be applied in order to mitigate the identified issues or to lower the risks involved in the workflow. The recommendations are based on the recommendation to use modified SEMIRAMIS framework [5].

This section considers the aspects which are relevant to attribute aggregation. Other functionality and infrastructure aspects are handled by other documents or sections. The functionality important to smart attributes relies partially on sections[3]. From that standpoint, although it may be related, issues or security analysis of the aspects already covered in those documents are not considered. However, due to the common context and issues, these aspects may be referred in this section.

The section has been organised the following way:

- 4.1: functionalities of the attribute aggregation process are described in the terms of processes or assets involved in the system.
- 4.2: list of identified threats in the domain of this analysis, providing additionally the table describing the relationships between functionalities and threats. These relationships help us to identify common points and their intersection, which could lead us later to select the optimal solution for the identified (potential) issues.
- 4.3: solutions which may be applied in order to mitigate identified threats. It also provides the table depicting the relationships between threats and solutions, which lead us to the optimal and recommended set of solutions to be implemented.
- 4.4: conclusion of the analysis

4.1 Use cases

Before entering in the description of possible issues, the uses cases are firstly discussed. The explanation of use cases uses PEPS terminology, but applies equally to V-IDPs. Only in the MW-MW scenario there is only one V-IDP involved, so it does not transfer control to other systems; the control and processes stay within the boundaries of the same system. Therefore, the issues in this situation may be less threatening.

[U1] IdP-AP SSO

This functionality refers to a connectivity model where a mutual trust relationship between IdP and AP is required. It requires the IdP and the AP to share a user identifier for each identity. The same shared identifier for multiple AP is allowed but a different one may be used.

The process flow looks as follows:

- AP checks if request comes from a trusted entity, in this case IdP3
- AP validates the embedded authentication assertion
- AP checks whether the attributes are requested for the corresponding identity, e.g. identity claimed by the assertion

This model provides a high level of usability but may enable user tracing/profiling.

[3] Chapter 3.3.4, M1: IdP-SSO of [Wp12]

[U2] Virtual SSO

This functionality refers to a connectivity model where a mutual trust relationship between APs and AAS is required. The APs trust that the AAS has validly authenticated the user. The method requires an AP generated shared user identifier (also referred as pseudonym) to uniquely identify the user on the APs. This identifier is not shared between different APs and is unique for each AP-user pair.

The process flow looks as follows:

- The AP receives an attribute request containing:
  - Authentication assertion and pseudonym
  - Pseudonym only
- If available the AP checks the authentication assertion
- The AP checks if the pseudonym is valid. If no authentication assertion is provided only the validity of the pseudonym is checked.
- If all checks passed, the AP sends the requested attributes.

[U3] No SSO

This connectivity model does not require any trust relationships. The user gets redirected to each AP and needs to authenticate separately.

[U4] Shared UID

As described in [U1] and [U2], the APs may require a shared identifier to refer to the corresponding attributes. Shared identifier in this sense refers to the same identifier shared between AP and IdP [U1], or AP and AAS [U2]. In some cases the same identifier may be applied across several AP-IdP or AP-AAS relations. To avoid possible user profiling and tracing across APs resulting from such naming, it is required to generate unique shared(permanent) user identifiers per relationship set, and link them to the corresponding identity account. This way, identifiers would be still shared between AP and IdP [U1] or AP and AAS [U2], but unique identifier would be used for each pairing set of those entities.

[U5] eID Coupling

Although it is out of the scope of this document, it is important to mention that eID coupling needs to be implemented in a way that prevents attribute impersonation or leakage. The problem of eID coupling is discussed in section 3. To summarise, either a manual way with additional human interaction or an automatic processing is to be applied on eID coupling. In the case of automatic processing, the method which does not produce false-positive eID coupling results should be applied, with the additional possibility to switch to manual coupling in the case of false-negative results.

4.2 Threats

In this section we describe the threats to the system and processes.

[T1] User profiling

When the user accesses the particular SP, IdP or AP, as a part of attributes collection and retrieval process, the adjacent entity might be able to collect the different attributes and profile a particular user based on the recurring authentications and attribute submissions. A special variant of this threat might involve the profiling of the users across several entities, which is less probable[4] but has higher impact.

[T2] Replay

This form of threat assumes the malicious entity, a part of the process, being able to resend the previously intercepted messages. Such activity may disrupt the service functionality and quality, leading possibly to a Denial of Service. Although less probable, this attack might be especially expected in a line of user interaction with AAS and APs. It is assumed for the other interaction points, which are included in the common infrastructure, to be already considered in the context of such attacks as a part of respective security assessments (SP – PEPS – AP – AAS interaction).

[T3] Flow break

In the case where no SSO connectivity method between AP and AAS is used, the user might be redirected through several entities (APs) located at various locations, including cross-country redirection. The purpose of this redirection would be to authenticate at particular AP, select and provide requested attributes, and then return back and continue with the process flow.

However, in this activity, due to the complexity of the flow and heterogeneity in infrastructures and approaches, it may happen that particular service is not reachable at the moment, thus breaking the complete attribute collection activity and leaving the session open. Although not primarily security issue, this problem, if not appropriately monitored and handled, might cause annoyance and frustration among the users, affecting overall acceptance and sustainability of the system. Additionally, the leftover sessions and collected but unprocessed data at AAS might introduce minor security issues.

[T4] Attribute leakage

The connectivity methods including IdP-AP SSO and Virtual SSO model assume trust relation between IdP and AP, in the former, and AAS and AP, in the latter case. The trust relation between those entities might be exploited in some scenarios, such as compromise of the system and software, unreliable administrators or through exploiting of unknown software vulnerability. This way, the malicious party may be able to retrieve attributes for arbitrary persons or groups of the persons without the knowledge and consent of those.

[T5] Attribute impersonation

This threat refers to the situation where the user is able to collect the attributes of some other user and include them in its attribute aggregation process. This way, the falsely retrieved attributes may be used to perform some fraudulent activity or obtain the permissions or resources which are not intended to authenticated user.

[4] The probability is related to the incentive and costs involved on the attacker's side. While the costs for profiling the users by one adjacent entity, based on recurrent sessions with the same entity, are relatively low, the costs related to involvement of several unrelated entities may be relatively higher in comparison with the incentive and possible gains resulted from profiling the user in such group attempt.

The following table shows the mapping between the use cases and the threats:

	[U1] IdP-AP SSO	[U2] Virtual SSO	[U3] No SSO	[U4] Shared UID	[U5] eID *Coupling*
[T1] User profiling	X	X	X	X	
[T2] Replay	X	X	X		
[T3] Flow break			X		
[T4] Attribute leakage	X	X			
[T5] Attribute impersonation	X	X	X	X	X

Table 7 : Attribute Aggregation Functions vs. Threats

4.3 Possible solutions

In this chapter the solutions which may help to mitigate the identified threats are proposed.

[S1] Extended audit

The possibility that the retrieval of attributes - based on trust relation between IdP and AP or AAS and AP – does not require direct user interaction nor re-authentication each time the attribute is requested fulfils the requirement of user-friendliness defined in section 3. However, the fact that the user is not able to directly decide whether the attributes are released/retrieved, opens possibility to misuse that trust relationship in the case of system compromise at any of the actors, or in case of some similar fraudulent activity.

In order to increase transparency and auditability of the approach, the system could be extended to enable the user to review the list of recent personal attribute retrievals. Such functionality can be provided either on AAS / IdP side or on AP side. The support at AAS / IdP side would however be more effective due to the fact that the user accesses AAS / IdP during attribute retrieval or may access it independently, while the frequency of accessing the AP's interface by the user is comparably lower[5].

The possibility to review the retrievals of attributes strengthens the overall security of the system[6] and its privacy-conforming perception. It furthers improves the confidence of the users that their data is properly handled.

[5] It should be noted that such functionality might introduce additional security/privacy related risks. These risks should be assessed and appropriately approached. However in this case for accounting purposes we propose the logging of the activities, not the exact values of the transactions (which lowers the potential privacy/security risks).

[6] As the users are able to review attribute retrievals, large scale or intensive user-profiling would hardly pass unnoticed.

[S2] Additional security measures

The usage of central AAS service increases its attractiveness for potential attackers and extends potential impacts of a successful attack. In order to minimise the risk involved with the compromise and misuse of the service, several additional measures can be implemented:

- Integrity and intrusion check: application of data/system integrity audit tools and host-based intrusion detection systems
- System hardening
- Strict and detailed security policies

[S3] Identity correlation

The attributes returned by APs receive at least the following elements from the User's profile: {name, surname, dateOfBirth}. Although not sufficient to uniquely identify a particular person, these elements may be used to perform an identity correlation with the user's identity data from IdP. This way, the risk of including the attributes of some other user will be significantly lower.

The identity correlation should be performed at the AAS, which would assess the similarity of referenced identity among elements provided from the IdP and AP entities and their respective requests and responses. If there is incomplete match of that data, the request to include a particular attribute should be either refused (if there is a substantial discrepancy) or allowed, but with the inclusion of the flag stating the level of discrepancy[7]. In this case, the SP may decide whether to accept or refuse such collection of attributes.

It should be noted that this analysis focuses on limited scope in the terms of eID-Coupling, which refers to reconciliation and linking of different Ids inside one country or across several countries. The issue of eID-Coupling is separately examined in section 3.

[S4] Attribute values encryption

The attribute values may be encrypted for particular receiving SP either by applying public key cryptography facilities or some other approaches. In the case of public key cryptography, the SP may optionally include its public key in the attribute request, which will be then used by APs to encrypt the attribute values for that SP. Hence, the AP will still not be able to profile user or get information about SP[8] (if not intended), while the attribute values could be only read at AP and SP facilities – intermediaries will see only encrypted data.

[S5] Anti-replay mechanism

The mechanism against replay attacks requires inclusion of the request status tracking methods, which are to be used to assess the state of the request. The actors in the process should maintain the tables with statuses of particular requests based on their ids. This way, the particular request is considered only in the case it was not already served e.g. its state conforms to the expected one. This approach would prevent unnecessary or unexpected loops which may lead to decreased user experience (in the case of error) or service disruption (in the case of attacks).

[7] This can be applied particularly in the case where the minor difference between processed data occurs, which may imply that the requests and responses refer to the same person. The example for that may be different handling of diacritic characters or transliteration rules in the names. Additional data describing discrepancy can be delivered with the flag, such as the number representing the similarity measure of the names. The details of possible implementation and included data are part the work of IJS under "Similarity of names" (to be delivered).

[8] The SP may use the different keys across the ranges of users, requests, dates etc.

[S6] Unique identifiers

Based on the similar functionalities related to usage of pseudonyms, as described in [U1] and [U2], the former may be prone to user profiling if the user identifiers applied for the IdP-AP-User relation are not unique per each relationship set. For such case the obligatory usage of unique identifiers per relation is suggested, which is to be understood as sector (actor) specific relationship identifier.

[S7] Graceful session handling

The system should use session handling techniques which are capable to recognize the problems in the process flow and offer graceful continuation or termination of the flow.

For instance, if the user is forwarded to select and submit attributes from the AP which is not reachable, the system should be able to recognize that the user returned[9] from that transaction. This should be done in accordance with S5 and its anti-replay mechanism. When the user comes back to the AAS from such transaction, it should be able to continue the flow[10] or to terminate it. In the latter case all data already gathered during the session, as well as session related information, would be cleaned and the user returned to initiating SP with appropriate status message. This suggestion also includes the option for the user to stop the attribute aggregation at each step of the flow.

The following table shows the mapping between threats versus solutions:

	[T1] User profiling	[T2] Replay	[T3] Flow break	[T4] Attribute leakage	[T5] Attribute impersonation
[S1] Extended audit				(X)	
[S2] Additional security measures				(X)	X
[S3] Identity correlation					X
[S4] Attribute values encryption	X				
[S5] Anti-replay mechanism		X			
[S6] Unique identifiers	X				
[S7] Graceful session handling			X		

Table 8 : Attribute Aggregation Threats vs. Solutions

[9] E.g. when the *back* button in browser is used

[10] In the case non-reachable AP is not critical for the transaction e.g. only optional attributes are delivered from that AP, the session (attribute collection) should be continued and eventually finished with successful delivery of the attributes to the SP.

In order to protect against all identified treats, all security measures should be applied, although [S1] and [S2] are partly redundant (but do not protect completely about the threats).

Remarks:

- [S1] and [S2] cover partial subsets of [T4];
- [S4] and [S6] cover partial subsets of [T1] problem. User profiling is a complex threat which may be exploited on many ways; therefore the more countermeasures are recommended to be implemented in order to increase level of security and privacy;
- [S2] and [S3] also cover different subsets of problems therefore the both are advised;
- [S3] is particularly important to consider due to different national cultures and variations in personal name representations; the implemented measure should provide higher level of security and precision in the process of correlating the personal names represented through similar transliterations;
- [S4] may be unfeasible to implement in the current infrastructure. However it is recommended considering such solution in order to improve the security and privacy related properties of the attribute aggregation process.

4.4 Conclusions security related to Attribute Aggregation

This section provides an overview on security related functionalities, the identified threats and recommended solutions to mitigate them.

It should be noted that eID correlation is important part of reconciliation of multiple identifiers among one or several countries. The solution of that minor problem may affect [T5], which is discussed in more detail in section 3.

Although some of the solutions refer to the same threat, due to the variability of particular threats and sub-optimal coverage of proposed solutions, it is advised applying all security measures to provide higher level of risk mitigation and security.

4.5 Implemented solution(s)

Communication between the C-PEPS/A-PEPS with its SPs, IdPs and APs is out of scope for the common parts of STORK 2.0.

5 Authentication on Behalf of

This chapter analyses the STORK 2.0 use case Authentication on Behalf (AUB) from a technical and operational security perspective. To this end, this section has focused on a simplified security risk management approach in which the security group has tried to identify the main risks this use case could be exposed to and proposed a set of technical controls to mitigate them.

For security analysis a lot of methodologies are available. To keep it practical an approach comparable to the methodology applied in STORK1 is used. In this approach threats for a selected entity or system are found and described. These threats are motivated by known attacks and errors. Then, security principles and objectives are derived from the identified threats and from requirements coming from relevant security policy regulations to by compliant to. Thirdly, security functions are defined that implement the security objectives and principles. Together they counter the threats. Finally, to make the circle round, for all security functions a mapping is made on the attacks and errors. This way a check is done whether all attacks and errors are captured.

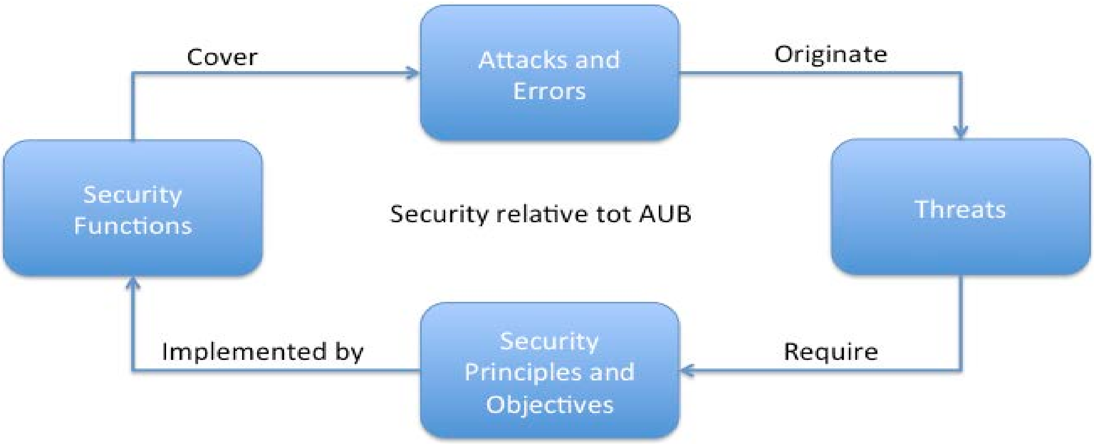


Figure 1: Security Approach

In order to describe security requirements, in the following subchapter this document first gives an overview of the entity in question: 'authentication on behalf', (AUB). It will then go on to the description of the concept and the definition of all components and communication relations of AUB as they are given by the architectural design. The next subchapter depicts the complete security table, in an overall view and in detail. Finally, some conclusions in the security topics on AUB are presented.

Based on the goals of STORK 2.0 and on the content of the identified use cases, it is assumed that the nature of the processes and information related to AUB are implying that:

1. IT support for AUB makes a significant contribution to the activities within the process and/or the production of services.
2. Only with big additional effort continuing the process is possible when IT is not available.
3. Deployment of the AUB system has a positive effect on the effectiveness and efficiency of the organization and/or between organizations within the EU.

Regarding to the methodology and standards applied, the selected countermeasures (Security Functions) are compliant with ISO/IEC 27001:2005, which basically corresponds to the classification of ISO/IEC 17799:2005 and applied in other methodologies like CRAMM, a Risk Analysis and Management Method.

5.1 System Overview

The basic AUB is defined in D4.2 First version of Functional Design [8] as follows: "*is the process that allows a user to access privileged data of the represented person. Usually this process ends with a fully identified user (representative) and represented person, which means that their eID data is transferred to the service provider (SP), and this SP recognises this user as a representative of a known customer, student, partner, or whatever relationship this represented person may have with the SP.*

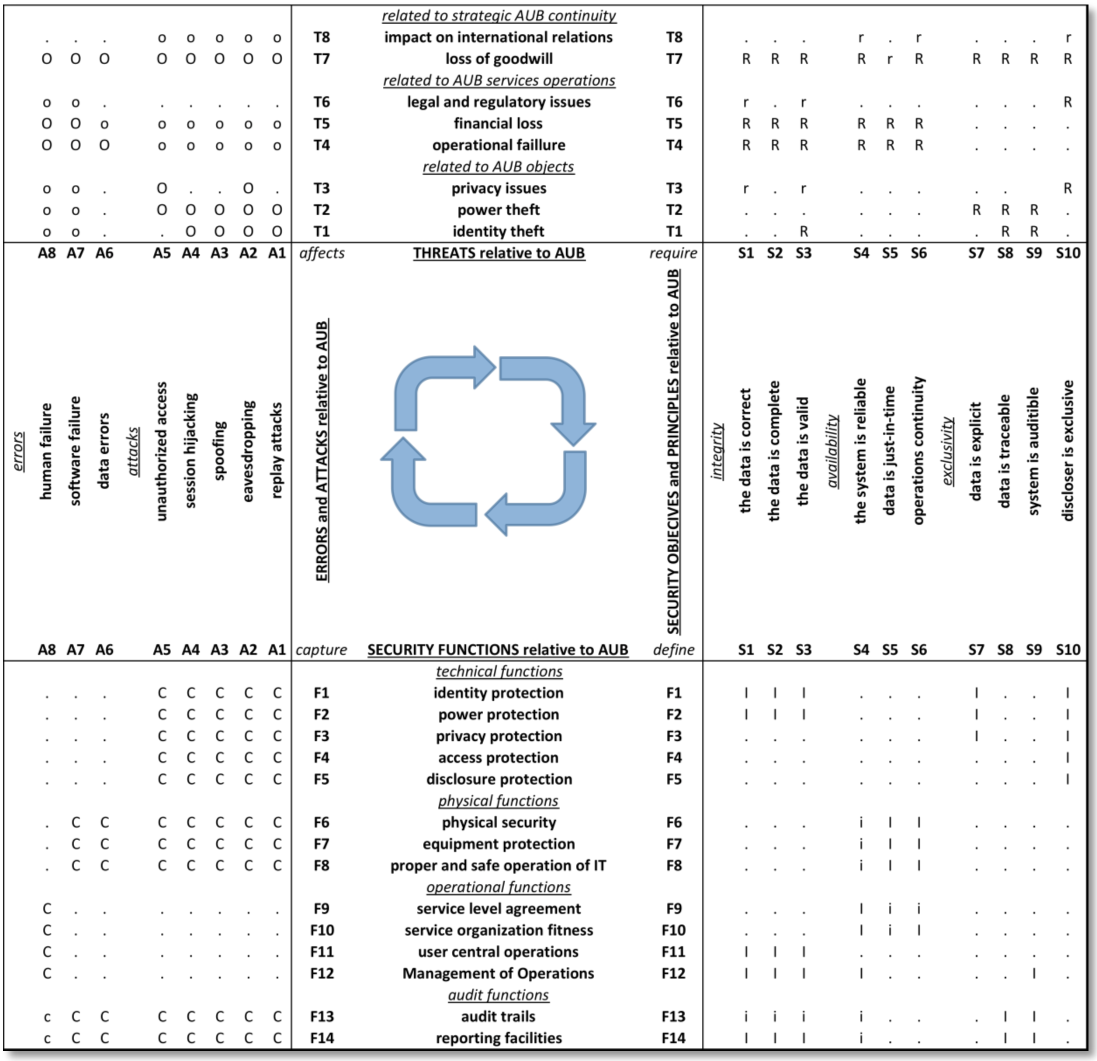
For an extensive system overview, architectures and descriptions on AUB, please refer to the documentation of WP4. The use cases of STORK 2.0 are issued in the pilots' documentation (WP5).

5.2 Security Analysis AUB

5.2.1 Overview

The complete security table is depicted in the table below. It is to be read in the direction of a clock, starting with the attacks and errors affecting certain threats. To prevent the damage caused from the threats all objective and principles required are mapped. After that the security functions are defined to deal with the objectives and requirements.

To close the circle, for each security function the attacks and errors, which are captured, are checked. If necessary more functions are identified if not all attacks or errors are captured.



									related to strategic AUB continuity											
.	.	.	o	o	o	o	o	**T8**	**impact on international relations**	**T8**	.	.	.	r	.	r	.	.	.	r
O	O	O	O	O	O	O	O	**T7**	**loss of goodwill**	**T7**	R	R	R	R	r	R	R	R	R	R
									related to AUB services operations											
o	o	.	.	.	.	.	.	**T6**	**legal and regulatory issues**	**T6**	r	.	r	.	.	.	.	.	.	R
O	O	o	o	o	o	o	o	**T5**	**financial loss**	**T5**	R	R	R	R	R	R	.	.	.	.
O	O	O	o	o	o	o	o	**T4**	**operational faillure**	**T4**	R	R	R	R	R	R	.	.	.	.
									related to AUB objects											
o	o	.	O	.	.	O	.	**T3**	**privacy issues**	**T3**	r	.	r	.	.	.	.	.		R
o	o	.	O	O	O	O	O	**T2**	**power theft**	**T2**	.	.	.	.	.	.	R	R	R	.
o	o	.	.	O	O	O	O	**T1**	**identity theft**	**T1**	.	.	R	.	.	.	.	R	R	.
A8	**A7**	**A6**	**A5**	**A4**	**A3**	**A2**	**A1**	*affects*	**THREATS relative to AUB**	*require*	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**
errors **human failure**	**software failure**	**data errors**	*attacks* **unauthorized access**	**session hijacking**	**spoofing**	**eavesdropping**	**replay attacks**	**ERRORS and ATTACKS relative to AUB**		**SECURITY OBJECIVES and PRINCIPLES relative to AUB**	*integrity* **the data is correct**	**the data is complete**	**the data is valid**	*availability* **the system is reliable**	**data is just-in-time**	**operations continuity**	*exclusivity* **data is explicit**	**data is traceable**	**system is auditible**	**discloser is exclusive**
A8	**A7**	**A6**	**A5**	**A4**	**A3**	**A2**	**A1**	*capture*	**SECURITY FUNCTIONS relative to AUB**	*define*	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**
									technical functions											
.	.	.	C	C	C	C	C	**F1**	**identity protection**	**F1**	I	I	I	.	.	.	I	.	.	I
.	.	.	C	C	C	C	C	**F2**	**power protection**	**F2**	I	I	I	.	.	.	I	.	.	I
.	.	.	C	C	C	C	C	**F3**	**privacy protection**	**F3**	.	.	.	.	.	.	I	.	.	I
.	.	.	C	C	C	C	C	**F4**	**access protection**	**F4**	.	.	.	.	.	.	.	.	.	I
.	.	.	C	C	C	C	C	**F5**	**disclosure protection**	**F5**	.	.	.	.	.	.	.	.	.	I
									physical functions											
.	C	C	C	C	C	C	C	**F6**	**physical security**	**F6**	.	.	.	i	I	I	.	.	.	.
.	C	C	C	C	C	C	C	**F7**	**equipment protection**	**F7**	.	.	.	i	I	I	.	.	.	.
.	C	C	C	C	C	C	C	**F8**	**proper and safe operation of IT**	**F8**	.	.	.	i	I	I	.	.	.	.
									operational functions											
C	.	.	.	.	.	.	.	**F9**	**service level agreement**	**F9**	.	.	.	I	i	i	.	.	.	.
C	.	.	.	.	.	.	.	**F10**	**service organization fitness**	**F10**	.	.	.	I	i	I	.	.	.	.
C	.	.	.	.	.	.	.	**F11**	**user central operations**	**F11**	I	I	I	.	.	.	.	.	.	.
C	.	.	.	.	.	.	.	**F12**	**Management of Operations**	**F12**	I	I	I	I	.	.	.	.	I	.
									audit functions											
c	C	C	C	C	C	C	C	**F13**	**audit trails**	**F13**	i	i	i	i	.	.	.	I	I	.
c	C	C	C	C	C	C	C	**F14**	**reporting facilities**	**F14**	I	I	I	i	.	.	.	I	I	.

Table 9: Depicted overview of security analysis AUB

Legend: a cell indicates

- ‘O’ (or ‘o’): the Error or Attack originates the Threat with major (minor) impact
- ‘R’ (or ‘r’): the Threat requires (may require) a Security Objective or Security Principle
- ‘I’ (or ‘I’): the Security Objective or Principle must (should) by implemented by the Function
- ‘C’ (or ‘c’): the Security Function covers (partially covers) the Error or Attack

In the next paragraphs all segments of the table are elaborated.

5.2.2 Attacks and errors on AUB

5.2.2.1 Attacks

[A1] Replay attacks

They are acts of capturing relevant session’s information and use it in coming actions to gain access to a given power, authentication or access to an e-service.

[A2] Eavesdropping

It is the act of listening a communication to get relevant information about the process. This information can also be used to perform other attacks, for example, to steal the session identifier using eavesdropping to perform session hijacking.

[A3] Spoofing

It is the act of hiding the real identity behind another one. This attack is heavily used at the network level (IP address spoofing) and in e-mail exchanges (origin address spoofing). One of the objectives is, for example, to "spoof" identity of a user that has more rights to gain access to a given power.

[A4] Session hijacking

It is the act of taking over an authenticated session to obtain its right on the system. This attack is done at application level.

[A5] Unauthorized access

It is the act of gaining access to a system without any authorization. This could lead to information leaks, but also execution of disallowed action for the specified user. This attack is done at network, operating system and application level.

5.2.2.2 Errors

[A6] Data errors

Data errors are errors that can occur, in computer data, during writing, reading, storage, transmission, or processing data, which introduce unintended changes to the original data. Data errors are commonly avoided using data integrity techniques.

[A7] Software failure

A software failure is misbehaviour of the program, which ends in an unspecified behaviour. This unspecified behaviour could lead to a software stop, suspended services or, from the security perspective, to a security problem.

[A8] Human failure

It is the act of a human who makes errors by wrong input, wrong manipulations on the system. Human errors refer errors introduced by humans in the system. If a given system has to be punctually manipulated by humans, they can introduce errors that can also lead to misbehaviour of the system. To avoid this type of errors, human interaction should be reduced to the maximum possible.

5.2.2.3 Threats relative to AUB

5.2.2.3.1 Threats related to AUB objects

[T1] Identity theft

Identity theft is the act of obtaining attributes and information of an entity in order to impersonate it. The objective is, for example, to perform privilege, or even illegal or abusive actions, under the theft identity.

This could be done using session hijacking, replay attacks, spoofing or eavesdropping.

[T2] Power theft

Power theft if the fact for an entity to change the assigned powers with ones he is not allowed to. The objective for another entity is to gain unauthorized powers to perform privilege actions.

This could be done through organizational problems, or through replay attacks, eavesdropping, spoofing, session Hijacking and unauthorized access

[T3] Privacy

In STORK1, the privacy definition is harmonised as:

> 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.

So, privacy is more often a disclosure of entity information (personal data, documents, messages, decision, etc.) The objective is to gather as much data as possible in order to profile or trace an entity.

This could be done using eavesdropping and unauthorized access.

5.2.2.3.2 Threats related to AUB services operations

[T4] Operational Failure

Operational failure means that the AUB services system has a disturbing or disrupting effect on the operation of an organization. This could be done particular by human failure, software failure and data errors.

[T5] Financial Loss

Financial loss happens if failure of the system leads directly or indirectly to financial losses (for SP). This could be done in particular by human failure and software failure.

[T6] Legal and Regulatory Issues

Legal and regulatory issues can result in civil proceedings or criminal prosecution. Some causes of this can be human and software failure.

5.2.2.3.3 Threats related to strategy AUB continuity

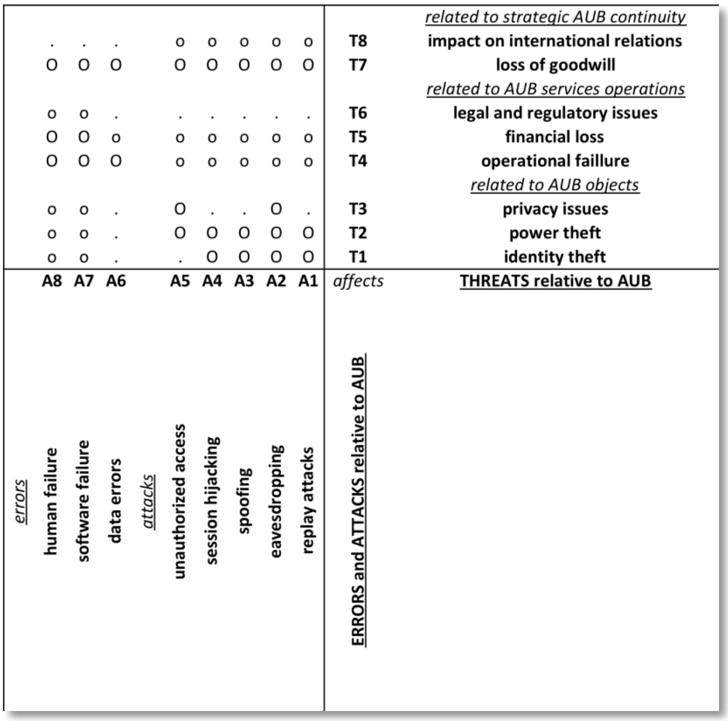
[T7] Loss of Goodwill

The loss of goodwill is related to loss of trust which leads to negative publicity and / or image damage. All errors and attacks can lead to loss of goodwill.

[T8] Impact on international relations

The impact on international relations is a failure that has a negative impact on diplomatic relations between nations. In this case there is some threat from the possible attacks on the system.

5.2.2.3.4 Conclusions from attacks and errors to threats



									related to strategic AUB continuity
.	.	.	o	o	o	o	o	T8	impact on international relations
O	O	O	O	O	O	O	O	T7	loss of goodwill
									related to AUB services operations
o	o	.	.	.	.	.	.	T6	legal and regulatory issues
O	O	o	o	o	o	o	o	T5	financial loss
O	O	O	o	o	o	o	o	T4	operational faillure
									related to AUB objects
o	o	.	O	.	.	O	.	T3	privacy issues
o	o	.	O	O	O	O	O	T2	power theft
o	o	.	.	O	O	O	O	T1	identity theft
A8	A7	A6	A5	A4	A3	A2	A1	*affects*	THREATS relative to AUB
errors			*attacks*					ERRORS and ATTACKS relative to AUB	
human failure	software failure	data errors	unauthorized access	session hijacking	spoofing	eavesdropping	replay attacks		

Table 10: AUB: Relation between attacks, errors, and threats

AUB is exposed to several attacks that could be materialized by exploiting several errors; the higher risk threats are "loss of goodwill", "financial loss", "privacy issues" and "power and identity theft". To mitigate the risk of these threats, several countermeasures as a conclusion for this section are proposed:

- Pack the powers within the SAML signature structure making for the user impossible to modify the data without being detectable. In the process, between the selection and delivery phases, recheck that the user has rights to access the powers the MS is going to issue.
- Include all the information about the representative in the signed SAML section. Also apply countermeasures for eavesdropping and replay attacks that could be used to spoof the represented identity.
- Provide end-to-end encryption within the process, that provides confidentiality of data, and also provide end-to-end data integrity using digital signature schemes.

5.3 Security Objectives and Principles relative to AUB

5.3.1 Integrity

The objective and principle of integrity denotes the certainty about the identity of the sender and the receiver, about the factual correctness, completeness of the information (in meaning, in value and in validity), and about the reliability of the systems that provide the information. In respect of AUB, integrity is subject to a number of requirements.

[S1] Data is correct

Any information transmitted or presented must be correct. By correct, we assume that the format of the data is correct, and neither the value. Those two point must be check in each data transaction. The main goal of this principle is to protect software against invalid data insertion.

[S2] Data is complete

Any information transmitted or presented must be complete. By complete, we assume that the information should not be partial.

[S3] Data is valid

In STORK1 security review, the just in time validity security principle was defined. This also applies to the AUB function in STORK 2.0:

> 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 out-dated (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.

5.3.2 Availability

Availability is to be divided in timeliness (the availability of AUB systems and information on the time that the information is required in a normal, reliable operating environment) and continuity (the availability in the long term, the continuation of the systems and operations after serious disruptions after which there is no longer a normal operating environment). In this respect, availability is subject to three requirements.

[S4] The system is reliable

The system consistently performs according to its specifications. In theory, a reliable system is totally free of errors but in practice, this is almost impossible to achieve, thus vendors express product's reliability as a percentage.

[S5] Data is just in time

Data necessary for the AUB system must be available just during the execution of AUB, not more. The main goal of this principle is to have data available for a good AUB execution and to protect the system against persistent data, or cached data misuse.

[S6] Operations continuity

The system must keep a high level of availability to provide a good reliability. Maintenance strategy on AUB component should be clearly identified to provide a good continuity after serious disruptions.

5.3.3 Exclusivity

Exclusivity is the entitlement to access only to predefined information. In STORK 2.0, context of AUB mandate definition is:

> The power of a legal or natural person to legally act on behalf of another legal or natural person. [...] a mandate is only the power explicitly given through contracts or company statutes, not the powers granted directly by law, like parents acting on behalf of their minor children.

Exclusivity is subject to a number of requirements:

[S7] Data is explicit

The system must show (display) all information of the mandate (description, start and end date, the legal or natural person).

[S8] Data is traceable

The system must provide traceability to be able to see who gave mandate to whom.

[S9] System is audible

The system provides audit capabilities to be able to know which user has which mandate(s).

[S10] Discloser is exclusive

As it is mentioned in the definition, the power must be *explicitly* given to an entity. It means that the user must see and validate the exact mandate he gives. Personal information revealed to an entity should be the minimal for the purpose of the service provided.

STORK1 minimal disclosure should apply for the AUB function in STORK 2.0:

> 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 citizens' life. The case where no persistent identifier is provided will be treated as a special case."

5.3.4 Conclusions on objectives and principles to threats

	SECURITY OBJECIVES and PRINCIPLES **relative to AUB**	*integrity*	**the data is correct**	**the data is complete**	**the data is valid**	**the system is reliable**	*availability*	**data is just-in-time**	**operations continuity**	*exclusivity*	**data is explicit**	**data is traceable**	**system is auditible**	**discloser is exclusive**
related to strategic AUB continuity														
impact on international relations	**T8**		.	.	.	r		.	r		.	.	.	r
loss of goodwill	**T7**		R	R	R	R		r	R		R	R	R	R
related to AUB services operations														
legal and regulatory issues	**T6**		r	.	r	.		.	.		.	.	.	R
financial loss	**T5**		R	R	R	R		R	R		.	.	.	.
operational faillure	**T4**		R	R	R	R		R	R		.	.	.	.
related to AUB objects														
privacy issues	**T3**		r	.	r	.		.	.		.	.		R
power theft	**T2**		.	.	.	.		.	.		R	R	R	.
identity theft	**T1**		.	.	R	.		.	.		.	R	R	.
THREATS relative to AUB	*require*		**S3**	**S3**	**S3**						**S1**	**S1**	**S1**	**S2**

Table 11: Relation between threats & security objectives

The table shows that all threats relative to AUB are covered by at least 3 security principles.

5.4 Security Functions relative to AUB

5.4.1 Technical Functions

[F1] Identity protection

Any entity (legal or natural person) must be clearly identified, and its information must be protected. For that:

- All the information should be included into the signed SAML section;
- Each entity should be authenticated and protected against replay attacks;
- Session must respect the just-in-time validity security principle (timeout in the signed section of the data exchange that indicates the session will expire in a reasonable amount of time such as 5 to 10 minutes),
- The system must provide end-to-end encryption within the process.

[F2] Power protection

Powers resulting of the "authentication on behalf" process should be protected. For that:

- The powers should be packed into the SAML signature structure making for the user impossible to modify the data without being detectable;
- Between the selection and delivery phases, the system must recheck that the user has rights to access to the power the MS is going to issue.

[F3] Privacy protection

The system must guarantee the minimal disclosure security principle. For that:

- User data transmitted over the network during the AUB function must be restricted to the minimum, and encrypted;
- User must be presented all requested information.

[F4] Access protection

Any access to any part of the system should be protected and restricted only to the authorised users. For that:

- Each user who wants to access any functions of the system must present its credentials;
- Each credential must correspond to the user and must be valid.

[F5] Discloser protection

Protecting the information and the supporting infrastructure by prevention unauthorized disclosure, modification, removal or destruction of assets and interruption of business, by controlling access to information, ensuring access for authorized users and unauthorized prevent access to information and by protecting the confidentiality, authenticity or integrity of information using cryptographic means.

5.4.2 Physical Functions

[F6] Physical Security

The prevention of unauthorized physical access, damage or disturbance of the site and the information of the organization, by:

- Physical security of the environment
- Physical access security
- Protection against external threats

[F7] Equipment Protection

By preventing the loss, damage, theft or compromise of assets and business interruption through placement and protection equipment utilities and maintenance of equipment.

[F8] Proper and Safe operation of IT

By proper facilities with documented operating procedures, change management procedures, segregation of duties, and separation of facilities for development, testing and production.

5.4.3 Operational Functions

[F9] Service Level Agreement

Accomplishing a suitable level of information and services by implementing, monitoring, maintaining and evaluating them in accordance with the contracts for services by a third party.

[F10] Service organization Fitness

By implementing awareness, education and training regarding information security, disciplinary measures, termination of responsibilities, blocking of access rights.

[F11] User Central Operations

The user performs the processes. The user is in control of the whole process, knows the information that is being transferred and is able to stop the process in any point. He transfers his relevant information from one point of the system to the following one. No side-channels are used to recover user owned information (i.e. personal data).

[F12] Management of Operations

Policy for Information Security, approved by a Board and published and known to be made to all employees and relevant external parties.

Management commitment to information security, actively given the organization support by giving clear direction to show commitment and explicitly assign responsibilities for information security and to recognize.

Coordination of information security, by representatives from different parts of the organization with relevant roles and functions. All responsibilities for information security must be clearly defined.

Approval process for IT facilities for new ICT facilities be identified and implemented.

Confidentiality Agreement, which are a reflection of the needs of the organization to protect.

5.4.4 Audit Functions

[F13] Audit Functions

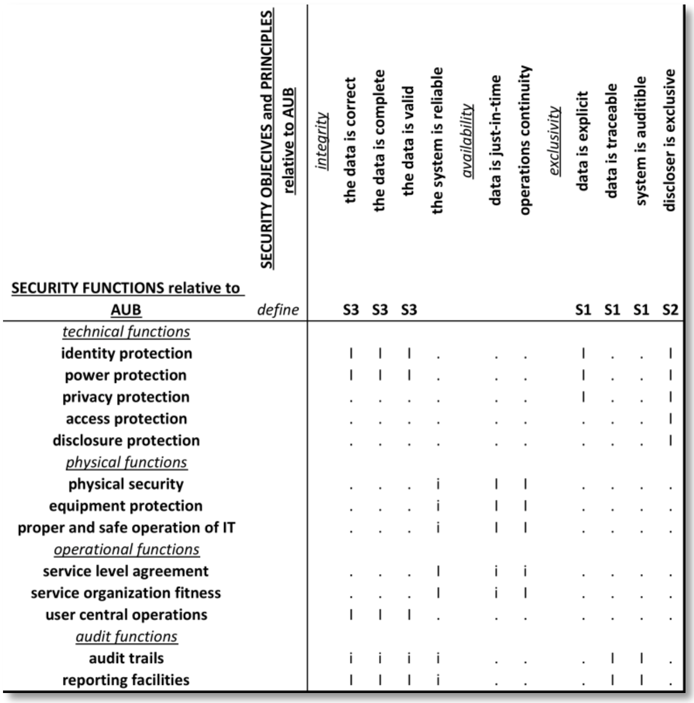
Independent reviews on information security, by the approach of the organization to manage information security and its implementation (i.e. management objectives, management, policies, processes and procedures for information) must be independent and planned intervals are assessed, or when there are significant changes arise in the implementation of security.

Screening, by the verification of the background of all candidates for employment, hired staff and external users to be carried out in accordance with relevant laws, regulations and ethical considerations, and being proportionate to the business requirements, classification of the information to which access is granted

[F14] Reporting Facilities

Ensuring that information security events and weaknesses associated with information systems such known be made that timely corrective action can be taken.

5.4.5 Conclusions of Functions to Objectives and Principles

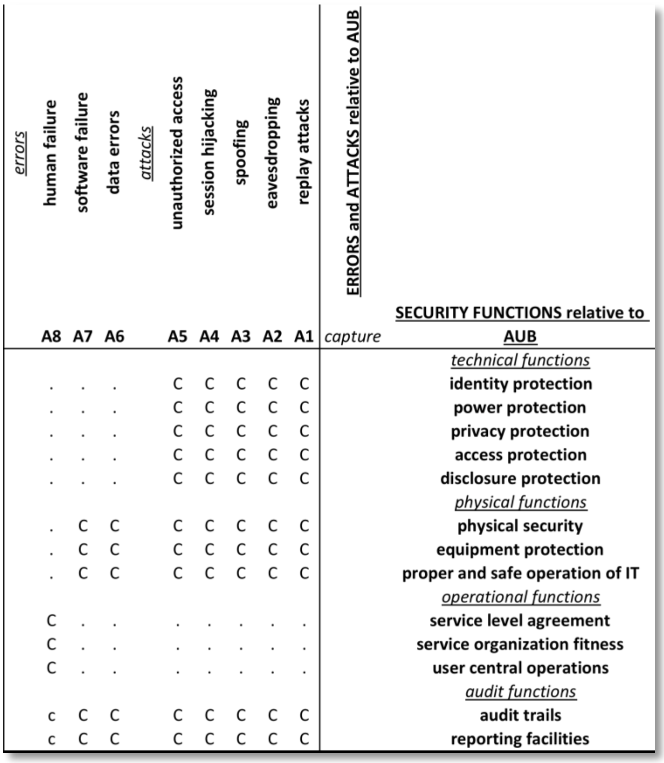


	SECURITY OBJECTIVES and PRINCIPLES relative to AUB	*integrity*	the data is correct	the data is complete	the data is valid	the system is reliable	*availability*	data is just-in-time	operations continuity	*exclusivity*	data is explicit	data is traceable	system is auditible	discloser is exclusive
SECURITY FUNCTIONS relative to AUB	*define*		S3	S3	S3						S1	S1	S1	S2
technical functions														
identity protection			I	I	I	.		.	.		I	.	.	I
power protection			I	I	I	.		.	.		I	.	.	I
privacy protection			.	.	.	.		.	.		I	.	.	I
access protection			.	.	.	.		.	.		.	.	.	I
disclosure protection			.	.	.	.		.	.		.	.	.	I
physical functions														
physical security			.	.	.	i		I	I		.	.	.	.
equipment protection			.	.	.	i		I	I		.	.	.	.
proper and safe operation of IT			.	.	.	i		I	I		.	.	.	.
operational functions														
service level agreement			.	.	.	I		i	i		.	.	.	.
service organization fitness			.	.	.	I		i	I		.	.	.	.
user central operations			I	I	I	.		.	.		.	.	.	.
audit functions														
audit trails			i	i	i	i		.	.		.	I	I	.
reporting facilities			I	I	I	i		.	.		.	I	I	.

Table 12: Relation between security objectives & security functions

This table shows which Security Objective or Principle the given Security Function must implement. As the table shows, every objective or principle is covered by at least five security functions, where most of them are distributed over technical, physical, operational, and audit functions.

5.4.6 Conclusions of Functions to Attacks and Errors



errors			*attacks*					ERRORS and ATTACKS relative to AUB
human failure	software failure	data errors	unauthorized access	session hijacking	spoofing	eavesdropping	replay attacks	SECURITY FUNCTIONS relative to AUB
A8	A7	A6	A5	A4	A3	A2	A1	*capture*
								technical functions
.	.	.	C	C	C	C	C	identity protection
.	.	.	C	C	C	C	C	power protection
.	.	.	C	C	C	C	C	privacy protection
.	.	.	C	C	C	C	C	access protection
.	.	.	C	C	C	C	C	disclosure protection
								physical functions
.	C	C	C	C	C	C	C	physical security
.	C	C	C	C	C	C	C	equipment protection
.	C	C	C	C	C	C	C	proper and safe operation of IT
								operational functions
C	.	.	.	.	.	.	.	service level agreement
C	.	.	.	.	.	.	.	service organization fitness
C	.	.	.	.	.	.	.	user central operations
								audit functions
c	C	C	C	C	C	C	C	audit trails
c	C	C	C	C	C	C	C	reporting facilities

Table 13: Relation between security functions & attacks

This table shows which Security Functions cover (partially covers) a specific Error or Attack. As can be seen, all errors and attacks are covered by a number of security functions. Especially for the attacks, the countermeasures are distributed over technical/physical as well audit functions. The errors are mostly covered by operational functions.

5.5 Conclusions security related to AUB

This section analyses the STORK 2.0 use case Authentication on Behalf (AUB) from a technical and operational security perspective. To this end, this section has focused on a simplified security risk management approach in which the security group has tried to identify the main risks this use case could be exposed to and proposed a set of technical controls to mitigate them.

Through this section the security issues relative to AUB have been analysed. We have stepped from a threat-focused approach to the attacks and errors that motivate them and also considering the security objectives and principles that should be provided through a security functions.

The study has identified the main risks and discussed the errors and attacks that can take advantage of these risks and finally how to apply countermeasures by clearly defining a set of security objectives and principles supported by a set of security functions.

Finally, given the study result, it is foreseen that it will not be difficult to add the security functions relative to AUB in the STORK infrastructure as far as they are minor modifications and, in most of the cases, the function is already provided.

5.6 Implemented solution(s)

The common code includes all relevant recommendations – that is, the ones described in section 5.4.1. Other sections are MS specific responsibilities, thus out of scope for the common functionalities

[F1] Identity protection

- All information is included in signed SAML tokens.
- SAML tokens are only valid once
- SAML tokens have a validity of 5 minutes
- The system uses TLS encryption, just like in STORK1, and in order to maintain the compatibility.

[F2] Power protection

- All information is included in (doubly) signed SAML tokens.
- A Session ID is used during the authentication and the delivery. This has to be the same to not break the flow.

[F3] Privacy protection

- User data transmitted over the network during the AUB function is restricted to the minimum and encrypted
- User is presented all requested personal information and needs to consent sending it in order to complete the transaction.

[F4] Access protection

- Access to the application before authentication is restricted to trusted Service Providers (signed SAML requests)
- Access to the authenticated parts is restricted to authenticated users (by calling MS-specific authentication code)
- Access to the administration of the system is out of scope for the common functionalities

[F5] Disclosure protection

- Disclosure protection is mainly a MS specific responsibility, which is out of scope for the common functionalities, however, no personal information is logged by the common implementation

6 Code security

An extensive security review of the PEPS code was performed by an independent team and most of the remarks have been agreed to be tackled. No high security issues should remain at the end of the project; a new review will validate this.

Here are the main remaining recommendations:

- All input – mainly SAML attributes – syntax should be better validated and in a more consistent and systematic way. Note that input (after digital signature verification) is always “trusted” because it comes from an authenticated partner; this mitigates the risk.
- Exception handling should be rewritten in a more consistent and systematic way.
- Logging/debugging features can lead to (limited) personal information disclosure to the PEPS system operators.
- Logging should be encoded to prevent some potential attacks.
- Demo code should follow all best practices to not give a bad example to a developer starting from this.
- *Struts* is known as a not very secure framework. If the project wants to keep on using it, people should enforce the Struts configuration, and add some security validation code.

7 References

[1] Session Management, OWASP, https://www.owasp.org/index.php/Session_Management_Cheat_Sheet

[2] Cross-Site Request Forgery (CSRF), https://www.owasp.org/index.php/Cross-Site_Request_Forgery_%28CSRF%29

[3] Chadwick, D., and Inman, G. Attribute Aggregation in Federated Identity Management. *IEEE Computer*, 2009, 42 (5), 33-40.

[4] Chadwick, D., Inman, G., and Klingenstein, N. A conceptual model for attribute aggregation. *Future Generation Computing Systems*, 2010, 26, 1043-1052.

[5] Semiramis Project. https://joinup.ec.europa.eu/software/semiramis/description

[6] Chadwick, D., Inman, G., and Siu, K. Expression of Interest – Improving Identity Management on the Internet. W3C Workshop on Identity in the Browser, May 2011, Mountain View (USA). http://www.w3.org/2011/identity-ws/papers/idbrowser2011_submission_12.pdf

[7] B. Zwattendorfer, D. Slamanig. Privacy-Preserving Realization of the STORK Framework in the Public Cloud, SECRYPT 2013

[8] STORK 2.0 D4.2 First version of functional design, https://www.eid-stork2.eu/index.php?option=com_phocadownload&view=file&id=23:d42-first-version-of-functional-design&Itemid=174) .