EUROPEAN PATIENTS
SMART OPEN SERVICES

Smart Open Services for European Patients

Open eHealth initiative for a European large scale pilot of patient summary and electronic prescription

Deliverable: Work Package Document

WP3.7

D.3.7.2. FINAL SECURITY SERVICES SPECIFICATION DEFINITION
- Section II - Security Services -

WORK PACKAGE	**3.7**
DOCUMENT VERSION	0.4.21
DATE	**16/06/2010**

COVER AND CONTROL PAGE OF DOCUMENT	
Document name:	The epSOS Security Services
Distribution level*	PU
Status	Final
Author/ person responsible: **Contributors:**	Eliberto Albertini / Giorgio Orsi (Lombardy)

* Distribution level: PU = Public, PP = Restricted to other programme participants, RE = Restricted to a group specified by the consortium, CO = Confidential, only for members of the consortium.

Sub-Project Identification		
Work Package	WP3.7	(Security Services)
Working Group	EWG C	(epSOS Security Services)

History of Alteration

Version	Date	Status Changes	From	Review
V0.1	11-30-2009	Draft	G. Orsi & E. Albertini (LOMBARDY)	FHGISST,GEMATIK, ELGA,IZIP,ASIP, MEDCOM,CLM,NICTIZ
V0.2	31-12-2009	Final (Internal)	G. Orsi & E. Albertini (LOMBARDY)	ELGA, CLM
V0.3	21-01-2010	Final (QR)	G. Orsi & E. Albertini (LOMBARDY)	ELGA,CLM,ANDA, NHS,GEMATIK,SALAR, FHGISST
V0.4.1	01-03-2010	Final (after QR)	G. Orsi & E. Albertini (LOMBARDY	ELGA,GEMATIK,ASIP, FHGISST
V0.4.21	30-04-2010	Final	G. Orsi & E. Albertini (LOMBARDY	ELGA,GEMATIK,ASIP, FHGISST

INDEX

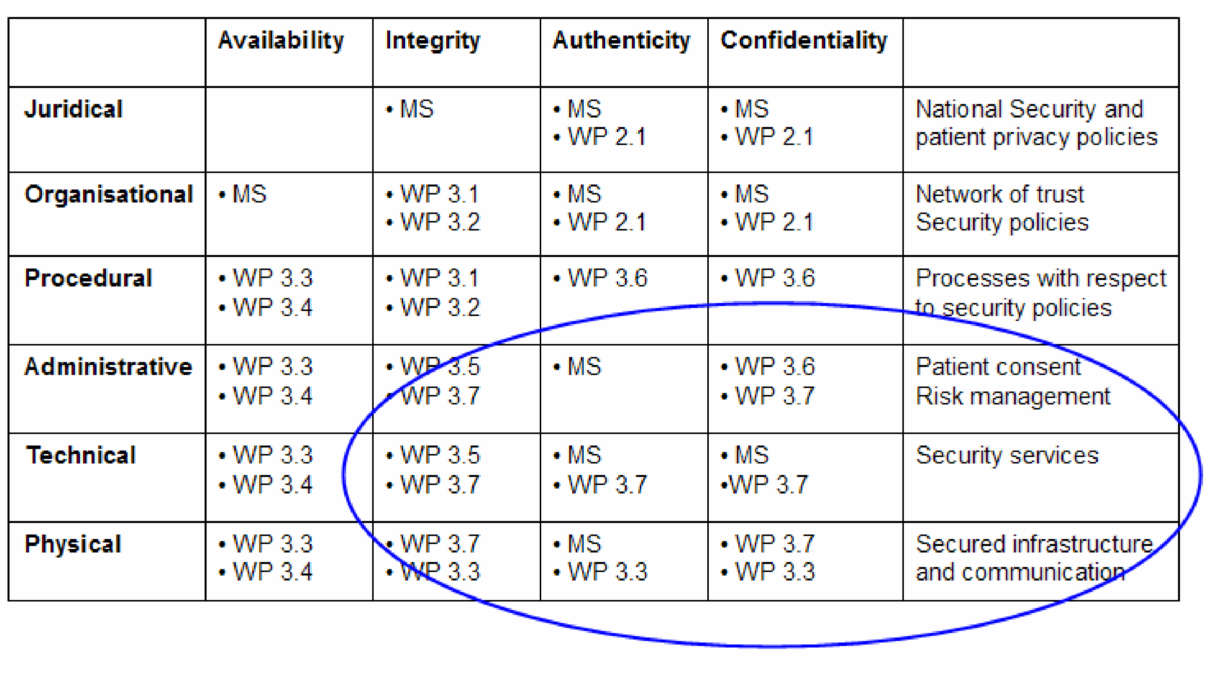
INDEX OF THE FIGURES

INDEX OF THE TABLES

1 INTRODUCTION

1.1 Security Service Objectives.

This document describes the security measures that must be put in place to grant the security (confidentiality, Integrity, authenticity and availability) of epSOS cross border communication of Health data. In particular the security service description has to provide evidence that all the epSOS-NCP security objectives are fully met.



	Availability	Integrity	Authenticity	Confidentiality	
Juridical		• MS	• MS • WP 2.1	• MS • WP 2.1	National Security and patient privacy policies
Organisational	• MS	• WP 3.1 • WP 3.2	• MS • WP 2.1	• MS • WP 2.1	Network of trust Security policies
Procedural	• WP 3.3 • WP 3.4	• WP 3.1 • WP 3.2	• WP 3.6	• WP 3.6	Processes with respect to security policies
Administrative	• WP 3.3 • WP 3.4	• WP 3.5 • WP 3.7	• MS	• WP 3.6 • WP 3.7	Patient consent Risk management
Technical	• WP 3.3 • WP 3.4	• WP 3.5 • WP 3.7	• MS • WP 3.7	• MS •WP 3.7	Security services
Physical	• WP 3.3 • WP 3.4	• WP 3.7 • WP 3.3	• MS • WP 3.3	• WP 3.7 • WP 3.3	Secured infrastructure and communication

Table 1-1: Security Model

As shown in the table above WP3.7 deals with several aspects which are part of "Security Management Systems". The following descriptions should help to give an overview about the focused themes and the appropriate duties of WP3.7.

Integrity

The original content of documents – regardless of where they come from, what they are used for or where they are shown - must always be the same and changes of the content must be consistent and traceable.

From the physical perspective these requirements can be achieved by using mechanisms that protect the transmission of documents and guarantee both entities (sender and requestor/receiver) unchanged content. This protection has to be included in the basic infrastructure of communication interfaces between different systems.

The technical aspects of integrity regard services which support the integration of "Security Management". These services have to respect at least two requirements. The first one is based on semantic issues which arise when the content of a document has to be translated or recoded (this issue is covered by WP3.5). The second one is important for WP3.7 because it covers tasks like prevention of altered medical information usage and protection against unauthorized access.

The administrative part of integrity covers accompanying measures of "Security Management Systems".

Authenticity

When health-related data are modified or exchanged, both sides (sender and requestor/receiver) involved in such a transaction must be assured that all entities in the communication process (in terms of data transfer) are authorised to execute the designated process steps. The technical aspects such as non-repudiation, auditing of “who requested/changed what”, etc. are under the conceptual responsibility of WP3.7.

Confidentiality

This important domain within epSOS is covered by different “Work Packages” with different approaches.

From the WP3.7 point of view the physical layer of this issue can be satisfied by usage of encryption methodologies which prevent data from being humanly readable. Appropriate infrastructures and mechanisms have to be established by the MS themselves but the requirements are defined within WP3.7.

Security services include the technical issues needed to achieve a high level of confidentiality. These services are not only an active support of confidentiality; they may also have passive components. An active part of these services is the separation of personal data (e.g. logon credentials or demographic data) from medical data storing (anonymization). On the other hand to achieve patients' satisfaction, a passive and proactive service which checks against “unauthorized and unwanted” access of any medical data (e.g. “emergency cases without patient consent”) can be established by each MS.

1.2 Security Services description method adopted

To describe each security service, the following approach has been adopted:

- **“Security Service Objective”,** making reference to the security requirements, this highlights the services goals.
- **“Security Service Scope”,** making reference to *actors*, transactions and UC, this describes the service scope.
- **“Requirements and Constraints”,** this points out the existence of specific or legal constraints.
- **“Risk Evaluation”,** in this paragraph the security service is analyzed to discover the presence of vulnerabilities. For each vulnerability found, the possibility of exploiting the vulnerability is studied and then an acceptable risk level is defined. Based on the results of the vulnerabilities analysis, and risk level definition, the paragraph identifies the security measures that may implement the security service – (this activity takes into account the experience realized by each MS). A Risk Analysis “baseline approach” model was adopted and an Excel form for a fast Risk calculation was proposed (no mandatory usage).

A more technical part of the description of the security service includes:

- **“Identification of Measures”,** Technical description of identified measures or reference to common standards to be adopted,
- **“Option Definition & Variant Analysis”.**

1.3 **Security Services not detailed by WP 3.7.**

1.3.1 Identification & Authentication

Identification and Authentication are usually the first security measures described in a security service document. In epSOS it was recognized that I&A of Patients and HCP (EU Citizen) was so a complex task and also that there were so many legal aspects and interactions with other EU Projects, that a specific WP was dedicated to dealing with this argument.

According to architecture defined, an Identity Provider is grouped with each epSOS-NCP and issues a profiled Identity Assertion.

Apart from HCP and Patients I&A, in epSOS there are two more classes of active entities to be Identified and to be authenticated: epSOS-NCP and epSOS-NCP users (technical staff and security officer).
The former, epSOS-NCP I&A in a epSOS-NCP to epSOS-NCP communication, is a common matter of a chosen secure protocol (TLS or SSL), the latter (epSOS-NCP users I&A), is a common utility usually provided by an operating system or data base.

For these reasons the WP 3.7 security services does not detail this Security Service.

1.3.2 Data Availability

Data availability is a security service. However in the epSOS piloting phase, a real threat due to a voluntary attack is not foreseen, moreover we consider that:

- Health data availability~~,~~ must have no impact on Patients health treatment. (*Health data availability makes the treatment based on a better foundation*).
- Denial of service (DoS) on communication is a common matter of secure protocol and networking architecture.

For these reasons WP3.7 security services does not detail any **Disaster Recovery Plan** nor describes any fault tolerance platform. In case of future evolution of the epSOS project the needs of a detailed data Availability security service shall be reconsidered.

WP3.7 makes a requirement to WP3.3-WP3.4. to provide a NCP platform in a such a way as to respect the SLA as defined by WP3.1. and WP3.2 Non Functional Requirements NFR01 and NFR03.

WP3.7 also makes a recommendation to WP4.2. to provide in each MS participating in the piloting phase a NCP back-up procedure.

1.3.3 Time Synchronization

Time synchronization has relevant impacts on security, it is important to resist a replay attack, and it is also important for a coherent timing localization of the same event in different epSOS-NCP.
Nevertheless the possibility offered by a standard product (i.e. NTP or SNTP) to minimize to a few milliseconds the time difference between epSOS-NCPs, makes meaningless to describe Time Synchronization as a Security Service when the acceptable timing tolerance should be ½ second.
This same issue is addressed in the “Time Synchronization Profile” by WP3.4.

1.4 Circle of Trust

1.4.1 epSOS Trust Model

Motivation:
A service has to require that an incoming message proves a set of claims, such as NCP authentication, HCP attributes, patient identifier, HCP role and patient consent – as defined in WP 3.1, 3.2 and 3.6. From the perspective of a web service implementor, trusting a peer in a web service message exchange is based on some security service, like the proof of possession of a cryptographic key applied to transport and message level security.
However, from the business perspective a wider view has to be taken to encompass *all* trust-related use cases, such as contract establishment, key management or software deployment. Therefore, all security-relevant factors need to be scrutinized with the questions “who trusts whom?”, and “on what basis is trust established?”.

There are several categories of security issues that need to be questioned:

- Integration of legal and technical perspectives: Because of a mutual lack of understanding between management and technical people, trust for processes crossing the boundary in between them need properly managed trust relationships.
- Security infrastructure integration challenge: Different components are enforcing different aspects of security, as there are epSOS and national identity and authentication systems, and a communication protocol stack with a sizeable number of libraries and subsystems. In a typical environment, these security components are not well understood by a single integrator or administrator and therefore a certain number of specialists and/or contractors needs to be trusted.
- Security management challenge: From a management perspective, there are multiple islands of administration specific to products and usually prone to error, and lack of coordination. Security policy management must not be isolated to a business unit, application, or product. This makes for a challenge in an epSOS environment where trust is based on the enforcement of consistent policies across multiple components of the architecture.
- Compliance to internal or regulatory policy is difficult to check. There is no automated way of knowing if the security events that are being generated by the individual components or system configuration parameters are indicating compliance with the policy or not.

Trust Management:
Trust management addresses trusted relationships between entities (organizations, actors, security domains, and systems). These relationships can be system-to-system, business-to-business, and so on.

- Trust management is occupied with three aspects, namely business, organization and technology. The business aspect deals with two legal entities agreeing upon a set of rules to conduct business.
- The organization aspect is concerned in establishing and managing security procedures, like the obligations of personnel to organization or how far administrators are trusted.
- The technology aspect deals with managing the infrastructure that supports the capability for establishing trust by cryptographic and other technical methods. These include key management, and other technical considerations like audit trails.

The following figures shown the trust model architecture (figure 1.1) and the trust bootstrapping for NCP (A) (figure 1.2).

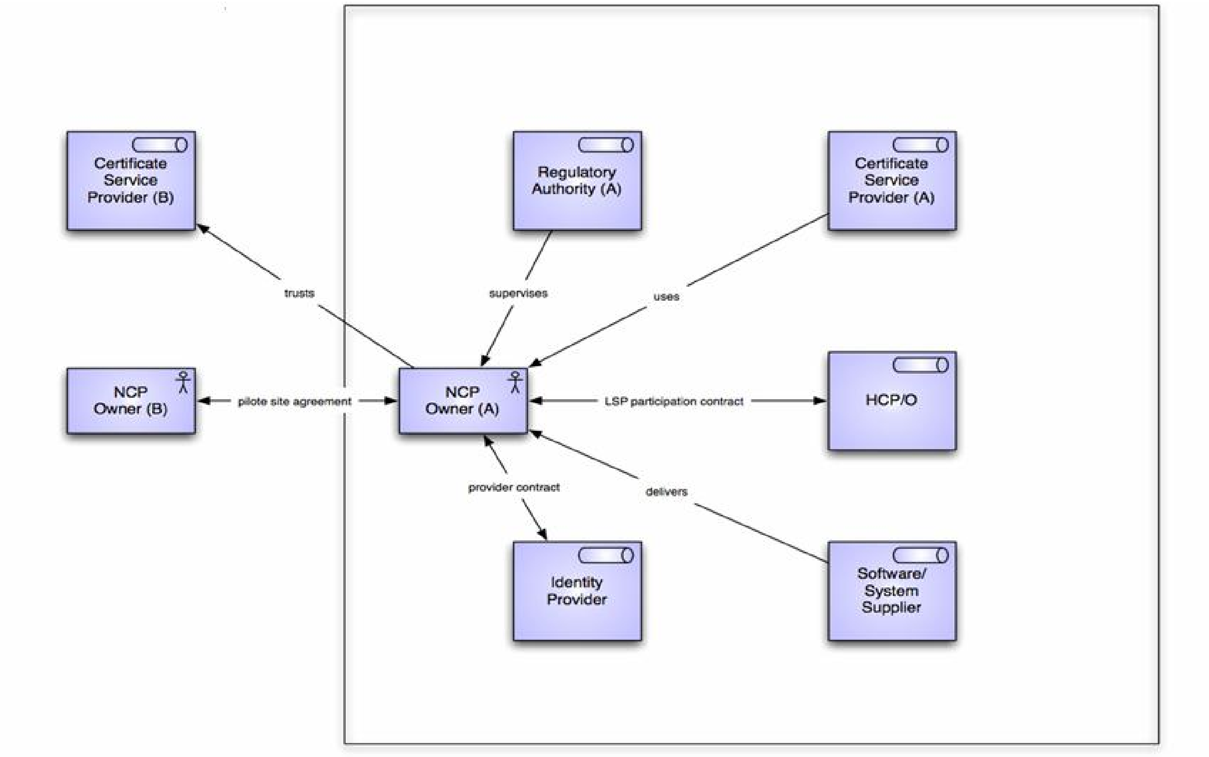


Figure 1-1: Trust Model Architecture (business view)

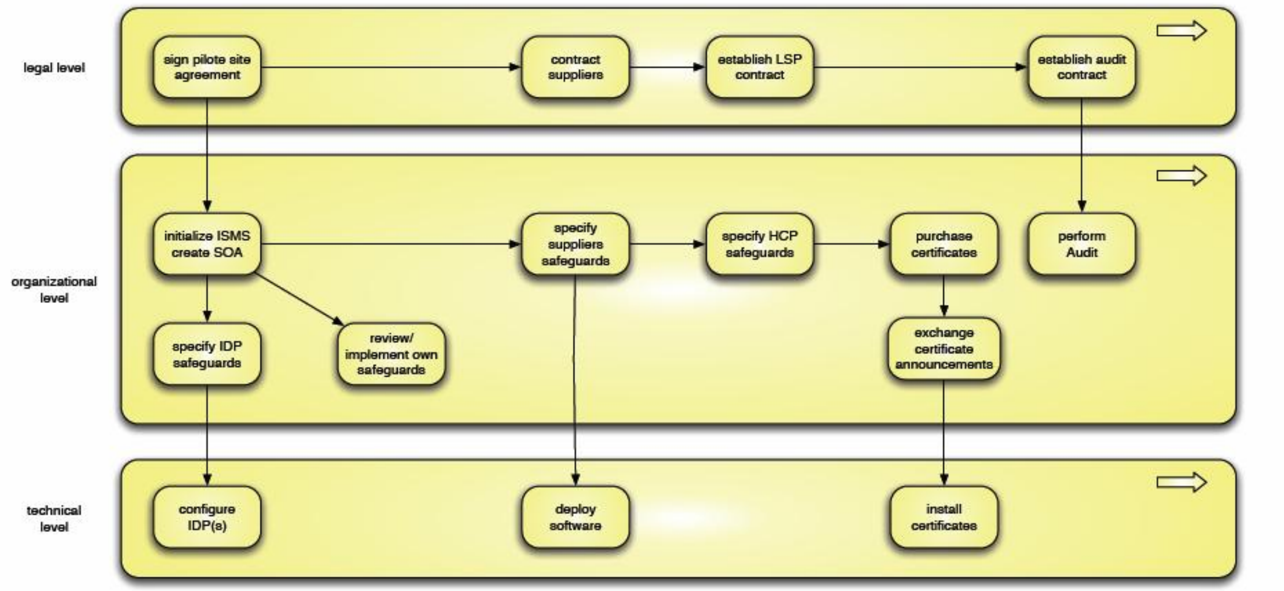


Figure 1-2: Trust Bootstrapping for NCP

Figure 1.2 highlight the relevant process.

WP 2.1 shall coordinate the definition of the legal Process.

Process at organizational level shall follow the security policy as defined by “WP3.7_D3.7.2_SECTION_I_Security_Policy_V07” and shall be tuned by each MS accordingly to the NCP implementation. Particularly a specific procedure shall address the

CoT secure initialization and shall describe the relevant organizational measures and all essential requirements. As there is no central system in epSOS, the off line procedure to exchange certificate (root certificate) shall prevent the –man in the middle- attack.

Process at technical level shall adopt the following prescriptions:

- WP 3.4 “Initialisation of the epSOS Circle of Trust” description (ie. the NCP Service Status List (NSL) definition)
- WP 3.7 security services description, this document, to protect the NSL stored in the NCP in such a way to avoid using it if altered (i.e. verifying the NSL integrity before the start of NCP on-line activities);
- for each MS the recommendation to install, protect and to use the purchased digital certificates of the chosen CA (Certification Authority).

2 Access Control Security Service

2.1 Introduction

An Access Control System (ACS) is a set of actors or services cooperating together with the aim of enforcing a policy given by a user to a resource. A user is willing to protect a resource by avoiding accesses from outside its own trusted domain (persons, applications, circumstances). The definition of the user's trusted domain is written in a policy document, in a specified formalism that an ACS can process and enforce.

Many of those formalisms for the definition of policies were defined (see for example Access Control Bibliography [**1**],[**2**]), together with many corresponding theories, from first order logics to type unification algorithms. Usually newer models do not fix deficiencies in the security provided by the earlier models, but they address changes in organizational structures, technologies, organizational needs and relationships. Access control methodologies can be factorized in four main paradigms, according to the existing healthcare standards [**3**]:

- Discretionary Access Control (DAC): the resource access decision is solely performed on a concrete identity of subjects and group membership. Subjects who possess rights over a resource may pass those to other subjects. A typical example is the UNIX file system access control mechanism and the POSIX ACL format.
- Mandatory Access Control (MAC): this mechanism makes use of sensitivity levels/labels, rules and policies. Users are usually assigned to a clearance level, and resources to sensitive levels. Rules determine how these levels can correlate for an access control decision. Rights delegation is also possible using rules.
- Role Based Access Control (RBAC): this model, and the Attribute Based Access Control (ABAC) and Policy Based Access Control (PBAC) evolution, does not assign a resource directly to identities, but to roles. A role is a semantic definition for a set of functionalities held by the user. An example is ``give access to the file only if the user is a dentist''.
- Context aware Access Control: this mechanism goes one step further by not assigning a subject to a role and a resource to a policy, and by defining a mechanism for enforcing rules, as the name suggests, aware to the context.

In the epSOS security model, an access policy consists of patient policies (policies that the patient defines for giving his/her privacy consent) and MS policies defined for alignment with the MS' local legislation.

From the results of the questionnaire (ref. WP3.7 D3.7.2 “Master document”) there is evidence to suggest an approach based on a mixed RBAC and Context-aware access control methodologies. Other approaches do not fit in the epSOS domain. An approach based on ACL (Access Control List) for instance, is difficult to manage where many people need to have different levels of access to many different resources [**4**]. An ACL refers to a particular identity and it is resource-focused and adding, deleting and changing ACLs on individual resources is time-consuming and error-prone. Unlike ACL, in the RBAC model access to a resource is determined based on the requester's role previously agreed among the MS and allows group individuals into categories of people who fulfil a particular role. The main disadvantage of the RBAC model arises from the latest assertion: dividing people into role based-categories makes it more difficult to define granular access controls for each person, but according to the questionnaires' answers this is not problematic for this epSOS security service. Taking into account the epSOS philosophy where the pilots must not require any changes to national health systems, each member state must self-rule the policies according to the local legislation, and each patient must be able to freely write the consent without any limitation introduced by the ACS technology suggested. More attributes can be added to enrich the role of the requester, such as the purpose of use for the data retrieved in a query. This leads to a more detailed approach named Attribute Based Access Control (ABAC), a superset of the RBAC that includes all the functionalities and adds the possibility of using more attributes in the access decision rather than the simple role. However, in an ABAC-only model, the attributes required to gain access to a particular resource are determined on a local level and can vary greatly from one member state to another. An evolution of the ABAC model is the Policy Based Access Control (PBAC) where every resource or a group of resources have a policy governing the accesses. This is the model suggested by the Access Control Security Service.

The expressivity of PBAC models leaves the possibility of making other deployments possible. A common case may be that there are more ACSs already existing in a Member State for enforcing local access decisions made upon local legislations and mechanisms (such as RBAC). In this example, the ACS grouped with the NCP is used to enforce the epSOS patient consent and then the NCP acts on behalf of the requester to access the national infrastructure. The access request to patient data can now be enforced by local ACSs.

The OASIS extensible Access Control Markup Language (XACML) specification was developed as a way to specify access control policy in a machine-readable format. XACML is a set of standards defining a policy exchange format, in order for systems to exchange or share authorization policies, even if the policies are translated into a proprietary or native policy language prior to the actual execution of the policy. XACML means that access control policies do not have to be tightly linked to the systems that they govern, but can spread across enterprises and organizations [**6**]. A policy is a set of targets (the resource to be protected) and rules (such as read/write).

In XACML, policy editors are free to write their own policy model. This makes XACML an ideal solution to the epSOS access control security service.

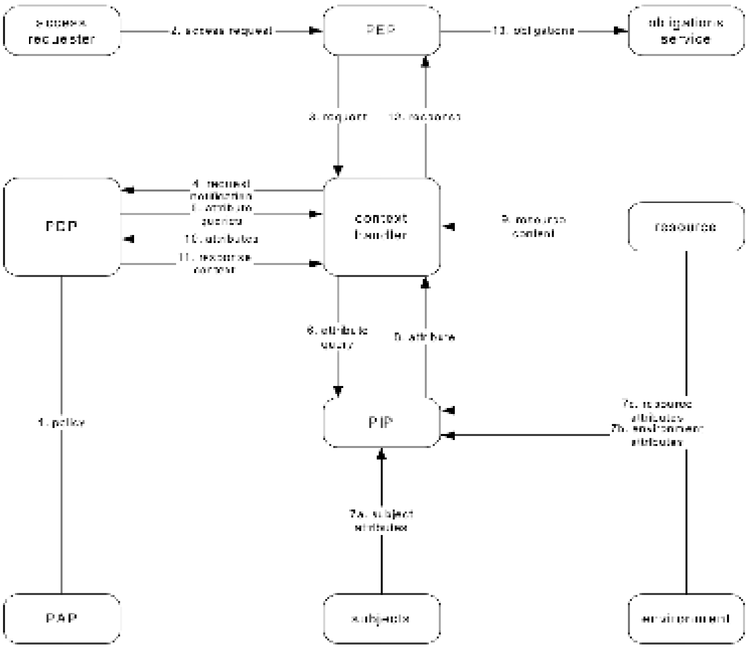


Figure 2-1: The XACML Data-Flow model

2.2 The XACML flow

The XACML specification [**6**], defines a policy format, a request format and a policy decision format. Policies can be defined in policy sets recursively[1]. The policies inside a policy set are combined together using a policy-combining algorithm, the procedure for combining the decisions and obligations for multiple policies. User defined algorithms are accepted by the XACML specification.

This mechanism SHALL be used by the epSOS access control security service to couple the patient policies together with the MS policies. The entire XACML data flow logical model is depicted in Figure 2-1. The Policy Access Point (PAP) makes the first action in the data flow. PAP contains policies (or it retrieves from a policy repository (PR)) and feeds the Policy Decision Point (PDP) (1). A request arrives at the Policy Enforcement Point (PEP) (2). The Context Handler (CH) manages the request for access to a resource (3) in its native request format, optionally including attributes of the subject, resources, actions, and environment. Now, the CH constructs a XACML request context and sends it to the PDP (4) for obtaining a decision request (permit/deny access to the resource). The PDP may request additional subject, resource, action and other categories to CH (5). CH forwards the queries to a Policy Information Point (PIP) (6), that obtains the requested attributes (7) and returns to the CH (8). CH can optionally include the resource in the context (9). CH sends the obtained attributes to the PDP (10) that can now evaluate the policy. PDP returns the response context with the authorization decision to CH (11) who translates the response to the native format of the PEP (12). Now PEP fulfils the obligations (3) (e.g. send an SMS if someone accessed the EHR) and if access is granted, the resource is given.

A PDP decision is one of the following:

- Permit
- Deny
- Indeterminate
- Not Applicable

The PDP decision is based on truth tables, defined by the XACML specification. When the request targets (i.e. the resources) ``match'' using a set of known functions with the policy conditions satisfied, then the decision will be the value of the rule, one of “permit”' or “deny”. All the other scenarios will result in Indeterminate or Not Applicable. This enables two versions of PEPs, a permit-based PEP and a deny-based PEP. The difference is when to permit access in case of Indeterminate or Not Applicable. The PEP model endorsed by epSOS SHALL be a deny-biased PEP.

By making reference to the Audit and Access Accountability security service, all the transactions passing through the NCP must be properly audited. In particular the event to be audited is

- HCP Authorization, when the access control decision has been made, by providing the consent assertion (the policy) and the authorization request as defined by the various use cases.

2.3 epSOS Access Control Security Service

2.3.1 Security Service Objective

The objective of the access control security service is to provide a means for the member states to enforce access controls on resources without interfering within the local legislation, at the NCP level. Each actor involved in the access control security service (PEP, PDP, PR, and PAP) will be grouped with the NCP.

[1] A policy set can be part of another policy set built on top of policies.

2.3.2 Security Service Scope

In "WP3.7 D 3.7.2 "Master document", the following specific security requirements have been identified for this Security Service:

1. **EpSOS-Req#3.7.4 (Access Control)**
2. **EpSOS-Req#3.7.5 (Access Control, privilege management and HCP authorization)**
3. **epSOS-NCP-Req#3.7.4 (Access Control)**

The Access Control Security Service fulfils requisite EpSOS-Req#3.7.4 because it is enforcing an access control decision prior giving access to a resource. The epSOS-Req#3.7.5 is satisfied because the access control decision is based upon the role played by the HCP (as per PBAC/RBAC mechanism). The epSOS-NCP-Req#3.7.4 is satisfied because the actors involved in the ACS are grouped within the epSOS-NCP, so the epSOS-NCP is providing an Access Control mechanism.

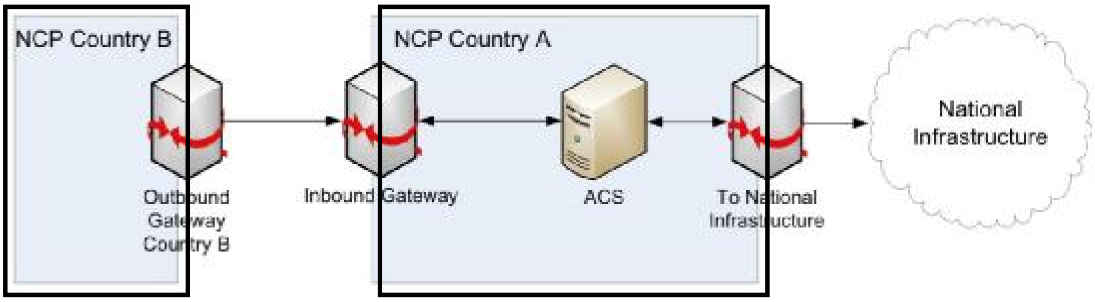


Figure 2-2: Architecture. The ACS filters each request coming from the other epSOS epSOS-NCPs

The epSOS' initial scope document together with Annex I, defines use cases for the large-scale pilot. One use case is defined for the ePrescription and two for the Patient Summary. Here, both scenarios are analyzed. The main idea is to provide an ACS in front of each epSOS-NCP that intercepts all queries and enforces policies, as depicted in Figure 2-2.

2.3.3 ePrescription use case

Use case for ePrescription has been introduced in Annex I, then defined in the Initial Scope document and further analyzed in the WP 3.1.

- UC1: Medicine already prescribed in Country A

In UC1[2], the NCP of Country A receives a request for validation for a given couple (patient | patient id). NCP loads the policy at this stage. In the second transaction of UC1, the `get_available_prescriptions()` message is sent. This message contains the identity of the dispenser with attributes such as the role ("dispenser"), the purpose of use (``Healthcare Treatment") and the epSOS trusted domain. When the ACS intercepts this message, it extracts the authentication token and generates a XACML request for the resource "prescriptions" for the given patient. For each prescription an access control decision is made and a new list of prescriptions (where the decision is "permit") is then created and sent back to the dispenser.

[2] The scenario is depicted in epSOS LSP: WP 3.1 Final doc.

The access control security service SHALL also be used for enforcing local legislation for accessing the patient medication summary.

2.3.4 Patient Summary Use Cases

Use cases for the patient summary are introduced in Annex I, then defined in the Initial Scope document and further analyzed in the WP 3.2. From the security service point of view, the use cases can be grouped together. By making reference to *Figure 2: Sequence diagram for the use case 1&2, Patient summary occasional and regular visits in the Document D3.2.2_v0.4.pdf",* the only transaction to be filtered by the ACS is the `get_available_patient_summary().`

2.4 Implementation Details

As already discussed in the previous sections, the transactions that need to be grouped with the Access Control Security Service are:

- get_available_prescriptions()
- get_available_patient_summary()

Both messages SHALL contain the identity of the HCP / Dispenser and the identity of the Patient, according to the outcome of WP 3.6. Such an identity is encoded as SAML authentication assertion containing all the attributes defined by WP 3.6 and taken as the basis for the access control decision.

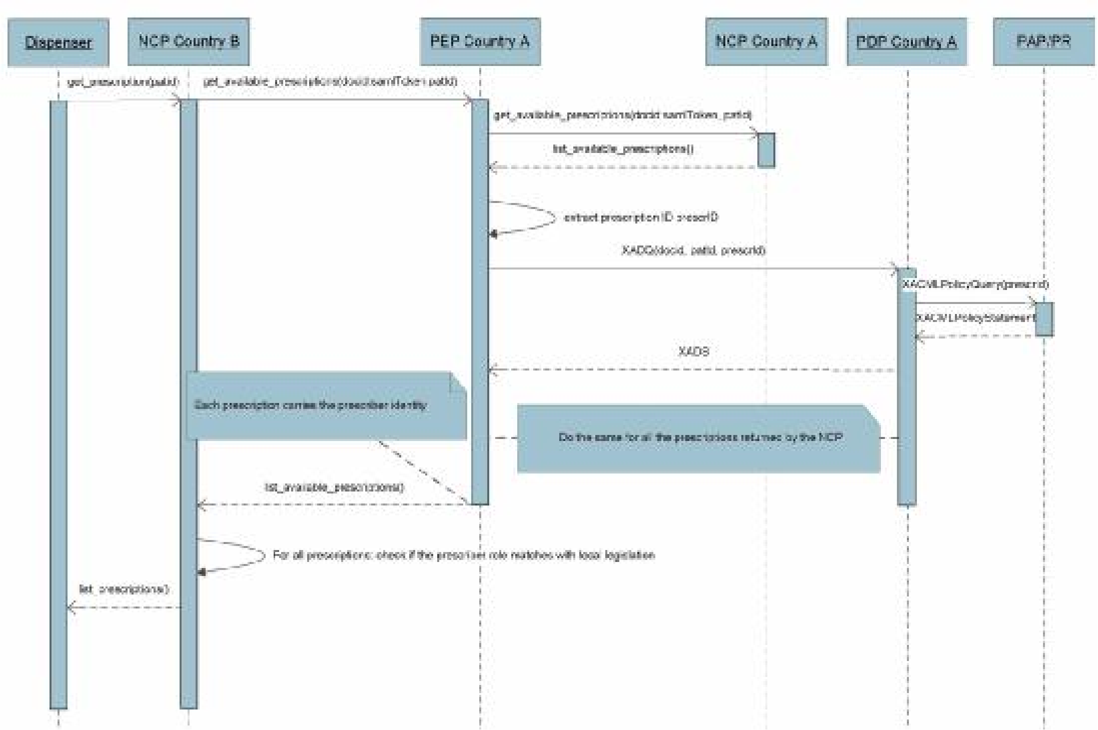

Dispenser
NCP Country B
PEP Country A
NCP Country A
PDP Country A
PAP/PR
get_prescription(patId)
get_available_prescriptions(docId:samlToken patId)
get_available_prescriptions(docId samlToken, patId)
list_available_prescriptions()
extract prescription ID prescrID
XADQ(docid, patId, prescrId)
XACMLPolicyQuery(prescrId)
XACMLPolicyStatement
XADS
Each prescription carries the prescriber identity
Do the same for all the prescriptions returned by the NCP
list_available_prescriptions()
For all prescriptions: check if the prescriber role matches with local legislation
list_prescriptions()

Figure 2-3: Implementation details for use case ePrescription

By making reference to Figure 2-3, the process is analyzed. The figure depicts the ePrescription use case. The message SHALL be conformant with [**7**]:

- The `get_prescription(docId:samlToken, patId:String)` message is sent by a dispenser in order to obtain all the available prescriptions for the patient. This message contains the identity and attributes for the HCP (i.e. in case of doctor, *docId,* as SAML token) and the identity of the patient, *patId*, according to WP3.4 and WP3.6 definitions. This server-side flow is sent through NCP-B to NCP- A.

- NCP-A validates the SAML assertion (methods defined in WP3.6, WP3.4).
- NCP-A obtains the prescriptions for the given patient ID and extracts a list of prescription ids (*prescrId*) to be used as *XACML Resources* for authorization decisions.
- For each prescription id contained in the list, NCP-A requests an authorization decision from the ACS actors (PEP, PDP, PAP/PR) for the *prescrId* and the identity contained in the *docId* SAML token to be used as request *Subject*.
- PEP in Country A creates a *XACMLAuthzDecisionQuery* SAML request according to [7] for the PDP in Country A. In this case, the PEP plays the role of implicit CH. The XACML request inside SHALL map the attributes contained in the SAML authentication assertion. The <Subject/> element of the request MUST have an attribute `urn:oasis:names:tc:xacml:10:subject:subject-id` as the SAML assertion subject. The patient MUST be inside the request. The epSOS defined role MUST be contained in the `urn:oasis:names:tc:xspa:1.0:subject:role`.
- If the policy is not cached[3], PDP in Country A performs a *XACMLPolicyQuery* to the PAP/PR in Country A for obtaining the policy governing the given prescription ID.
- Once the PDP in Country A gets the policy, it performs the policy decision and replies to the PEP with the *XACMLAuthzDecisionStatement*.
- A deny-based PEP enforces the decision from the PDP and informs the NCP-A.
- After all access control decisions have been made, the NCP-A creates the "filtered" list of available prescriptions. Each prescription MUST contain the prescriber functional role in the prescriber identity as defined by epSOS.
- The list of available prescriptions is sent back to the NCP-B.
- For each prescription, the NCP-B enforces local legislation duties using a similar mechanism. It checks if the prescriber role matches with the permitted roles in the local legislation, according to the Example 3 of the document *The epSOS Trusted Domain(s), Consolidation of Concepts*.

A similar scenario applies for the Patient Summary use case, where the transaction is the `get_available_summary().`

2.5 Emergency use case

Particular focus is placed on the access of a patient summary in an emergency situation i.e. when the patient is unconscious. Since the ACS intercepts each transaction passing through the NCP, all MS MUST provide a special policy covering this case providing emergency access to data. However, giving such unconstrained access can potentially leave the possibility to professionals of completely bypassing the policies defined by the patient. Taking into account these requirements, the emergency use case message flow is defined as follows:

- The HCP logs onto the system and selects a PS retrieval for emergency access. The user interface SHALL inform the HCP about the audits related to such an access.
- The SAML assertion containing the identity of the HCP MUST have the attribute urn:oasis:names:tc:xspa:1.0:subject:purposeofuse set as "EMERGENCY'.
- The HCP performs a normal PS query to the NCP-B that forwards it to Country A.
- The PEP located in the NCP-A looks at the #purposeofuse attribute value. If the "EMERGENCY" is found, it enforces the emergency policy defined by the MS by performing the XACML flow.

[3] The mechanisms for caching policies and policy decision are discussed in *WP 3.4: Common Component Specification.*

- The identity of the HCP is stored in a table. After a number of accesses made by the same HCP identity in a defined amount of time, the access control decision will be “Deny’’ and the reason SHALL note that too many attempts for an emergency access have been made.
- The ACS logs the event in the Audit Record Repository.

The limit is introduced to avoid the possibility of a malicious user gaining unauthorized access to the patient summaries by using the emergency use case. The limit SHALL be defined by each member state.

2.6 Requirement and Constraints

Each MS is required to use XML language to describe the access policies.
.

2.7 Risk Analysis

It is unfeasible to find an acceptable list of possible threats. Extensive literature exists with the aim of formalizing security protocol by assuming “perfect” cryptographic primitives (see [**8**] for example). From the security service point of view, the policy generation and protection is not a risk to be evaluated, since all the actors covered in this document (PEP, PAP, PR, PDP) are recommended to be grouped in the epSOS-NCP and each MS is responsible for epSOS-NCP protection. However adequate protection against policy modification and tampering MUST be in place.

To avoid policy misunderstandings, the use of *policy templates* is suggested (e.g. a XACML encoded template where patients fill the necessary values such as resources, subjects according to the MS laws).

2.7.1 Attack Analysis

No attacks were found. Due to the reason that each actor covered in this document is grouped within the epSOS-NCP, all the attacks can be factorized by general epSOS-NCP protection.

3 Data Integrity Security Service

3.1 Introduction

Integrity is one of the basic information security requirements. To guarantee the integrity of data means to introduce the measures against undetectable modification of data. In a broader sense the integrity covers the technical systems consistency/robustness.
This document provides requirements that address protection of epSOS data while it is stored within containers controlled by the epSOS system or while being transferred within the epSOS system. Integrity errors may affect epSOS data stored in memory, or in a storage device. This document does not deal with protection of the epSOS data from integrity errors while being transferred within the epSOS system (for this safeguard ref. to “Data Exchange security service”).

3.2 Security Service Objective

In “WP3.7 D 3.7.2 “Master document” the following specific security requirements have been identified for this Security Service:

- ***NCP-Req#3.7.08 (System and data integrity).***

The objective of Data Integrity Security Service will be to provide epSOS-NCP with a method that can find out if any unauthorized change to health data was applied.

3.3 Security Service Scope:

The epSOS-NCP Data Integrity scope, coherently with the epSOS project is restricted to the transactions defined to exchange Patient Summary as described by UC 1–2 and ePrescription - eDispension as described by UC 1-2.

3.4 Classification of epSOS data

WP 3.7. introduced the following classification schema for epSOS data and specified
the impacts of damage on the confidentiality, integrity, availability, authenticity and non-repudiation of the data in data classes.

Healthcare-related Data: Healthcare Data of a patient e-prescription or patient summary used for the medical treatment of a patient. Healthcare related data also contain:

- data about the patient’s consent and
- log data which provide information about the access to healthcare data in epSOS. They do not contain any healthcare data.

Critical Meta Data: Data needed to control the exchange of healthcare related data between epSOS-NCPs and between epSOS-NCPs and PoCs, respectively. Critical Meta Data include authentication, identification and administrative data to clearly identify patients and PoCs, HCPs…

Non-critical personal data: Personal data which do not belong to the special categories of data according to Article 8 Directive 95/46/EC and do not belong to the critical meta data.

Administrative Data: They do not contain any personal data. They are used for the administration or configuration of technical components.

Data class	Confidentiality	Integrity	Availability	Authenticity	Non-repudiati
Healthcare related Data	High	High	Low	High	High
Critical Meta Data	High	High	Low	High	High
Non-critical personal data	Moderate	Moderate	Low	Moderate	Moderate
Administrative Data	Low	Low	Low	Low	Low

Table 3-1: Classification of epSOS data types

3.5 Risk “analysis”

The detail risk analysis requires at least the description of the system and of its security environment, the list of requirements following from legislation, security policies, etc. This information is not available and therefore only generic threats can be taken into consideration.

Assumptions:

1. epSOS data are stored in systems (memories) which are adequately physically protected against:
   a. natural hazards,
   b. unauthorized access.
2. the epSOS system is being run by qualified and trained personnel
3. operators, administrators of the epSOS system will not intentionally manipulate the system.

Threats against integrity of data

1. technical failure which causes the modification or destruction of data
2. human error
3. malicious software
4. intentional attack

Vulnerabilities

1. Bad configuration or configurations errors
2. weakness of the security mechanism used
3. hidden errors.

Risk assessment

With a reference to the method of Risk calculation adopted the value was evaluated to medium.
To reduce this Risk, architectural, technical and organizational/procedural measures shall be implemented.

3.6 Measures

1. proper maintenance of the system and of its technical infrastructure;
2. regular system maintenance and upgrades (software);
3. system and data backup;
4. configuration management;
5. firewall;
6. installation and regular updates of antivirus software;
7. IDMS.

Besides the aforesaid measures, a technical measure based on cryptographic mechanism shall be implemented and two possible solutions could be adopted:

1. Use of document fingerprints;
2. Use of digital signatures.

While both solutions effectively resist unintentional errors, the use of digital signatures could better resist intentional attacks. However considering that Technical staff (administrator) may have direct access to the health data in the epSOS-NCP, the use of digital signatures becomes less secure due to the possibility for these users to have access to the cryptographic keys.

Acceptable algorithms to calculate the message digest for both solutions could be: SHA-1 or SHA-2.

In the case of a digital signature solution the key length should be no less than 1024 bit

3.7 Option analysis

In order to guarantee the integrity of the documents and messages between NCPs, a couple of alternatives are available, each with their own merits. In the following, these different options are described, and their pros and cons are explained.

1. Combination with digital signatures.
   Since digital signatures are used to implement non repudiation aspects of communication between NCPs, these signatures allow excellent opportunities to incorporate data integrity of the messages. Namely, in order to efficiently compute the signature, a hash function is used to compute the unique small digest value, before the signature is actually computed. This digest value also provides an automatic way to protect the integrity of the message.

2. Signed check sum.
   Integrity is often ensured by using some kind of checksum, which implies that for each message a (small) checksum is computed which is appended to the message. When the message is modified underway, the original checksum doesn’t match anymore, so the receiver of the message can detect this. In order to prevent the checksum from being modified accordingly, it should be protected somehow. One way of doing this is by using the available PKI and signing the checksum with the private key of the sender.

3. Separately encrypted checksum.
   This option is similar to the previous option, with the difference that the checksum is

not signed by using a PKI, but by using a symmetric key. Since NCPs don’t share a symmetric key for other purposes, such a key would have to be introduced and managed for the sole purpose of data integrity. One could use a different symmetric key for each NCP pair, but probably one key for all NCPs is good enough.

Option	1	2	3
Pros	o No extra key management o Functionality already available	o No extra key management	o Independent of digital signature, so could be used when the PKI is temporarily unavailable
Cons	o Only available in combination with digital signature	o Only available in combination with digital signature o Extra computations for checksum and signing/verifying o Additional message field needed for the checksum	o Extra computations for checksum and encryption/decryption o Extra key management o Additional message field needed for the checksum

Table 3-2: Data Integrity Options Summary

Since messages can only be exchanged when the PKI is available and each message is digitally signed, there is no real disadvantage for using option 1. This makes option 1 the clear choice within epSOS for assuring data integrity.

4 Data Confidentiality Security Service (*of stored data*)

4.1 Introduction

Confidentiality is one of the foundational concepts of information security. The meaning of confidentiality is defined in the ISO-27002 specification as "ensuring that information is accessible only to those authorized to have access". Confidentiality can be achieved using different mechanisms, such as access control techniques, encryption and anonimization. The *epSOS confidentiality security service* must keep the data secret (healthcare and demographic) stored in the epSOS-NCP persistent memory. The aim for this security service is to suggest a methodology for the member states for achieving confidentiality using encryption and pseudo-anonimization. Access control methodologies are defined in the *epSOS Access Control Security Service.*

4.2 Security Service Objective

Objective of the *epSOS confidentiality security service* is to provide confidentiality for data stored by the epSOS-NCP in its persistent memory (e.g. hard-disks, tapes) for a (un)limited amount of time.

4.3 Security Service Scope

The epSOS confidentiality security service addresses the following requisites, according to the definition of EpSOS Requirements and NCP Requirements defined in the “WP3.7 D.3.7.2 “Master document”.

- ***EpSOS-Req#3.7.06 (Confidentiality);***
- ***epSOS-NCP-Req#3.7.05 (Confidentiality);***
- ***epSOS-NCP-Req#3.7.07 (Protecting Data Storage).***

The requisites EpSOS-Req#3.7.06 and epSOS-NCP-Req#3.7.05 are satisfied because this security service is suggesting the use of encryption for data stored on persistent memory. If a member state decides to use pseudo-anonymization, then the epSOS-NCP-Req#3.7.07 is satisfied.

4.4 Confidentiality Mechanism

The encryption of a given message is a *transformation* applied to data using an *encryption key* that returns another message unintelligible to other systems that does not own the key. Two important encryption methods exist: *symmetric encryption* and *asymmetric encryption*.

Symmetric encryption was the first tentative for achieving the confidentiality goal. The main advantage of symmetric encryption is the reduced computational costs of the modern algorithms. The main disadvantage is the key management and protection. Brute force attacks covers usually the key length. Nowadays current symmetric encryption algorithms includes:

- Data Encryption Standard (DES), now deprecated, was adopted in 1977 with a key length of 56 bits.
- Triple DES (3DES), uses a DES schema with three keys for addressing attacks based on the reduced key length, and meet-in-the-middle. Key is 168 bits but can be reduced to 112 bits
- Advanced Encryption Standard (AES), is the current symmetric encryption algorithm,

officially adopted in 2001. Based on the Rijndael method, the key length can be 128/192/256 bits.

Symmetric encryption is widely used with the AES algorithm. Kerberos and TLS are application examples. However key distribution is still an open problem.

4.4.1 Use of XML Encryption

Another methodology for achieving data confidentiality aligned with the goals previously mentioned (strong encryption and low computational cost), is to use XML Encryption. This can partially mitigate the risk of using a single AES key encrypting all the files or the whole file system. The idea is to forge a new AES key for each file, and use it to encrypt. Then use the private key of the epSOS-NCP provided by the epSOS PKI security service to encrypt the AES generated key, in an XML structure defined as the XML-ENC W3C recommendation.

```
<xenc:EncryptedData xmlns:xenc="..."
    Type="http://www.w3.org/2001/04/xmlenc#Content">
    <xenc:EncryptionMethod
          Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-cbc" />
      <ds:KeyInfo
       xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
       <xenc:EncryptedKey>
         <xenc:EncryptionMethod
          Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5" />
          <xenc:CipherData>
           <xenc:CipherValue>
             TtljPAbr5...
            </xenc:CipherValue>
          </xenc:CipherData>
         </xenc:EncryptedKey>
       </ds:KeyInfo>
      <xenc:CipherData>
      <xenc:CipherValue>
       64oNEBHk365z
      </xenc:CipherValue>
     </xenc:CipherData>
</xenc:EncryptedData>
```

The algorithm for producing such a structure is defined as follows:

1. Base64 encode the object that needs to be encrypted
2. Generate the AES key
3. Encrypt the object using the key
4. Create the XML-Encryption structure by placing the AES encrypted object as EncryptedData/EncryptionMethod/CipherValue/ and by assembling a EncryptedData/EncryptionMethod/KeyInfo that contains the AES key encrypted with the epSOS-NCP private key using RSA encryption as CipherValue element content[4].

The decryption algorithm is defined as follows.

1. Obtain the keys used for encrypting the AES key.
2. Decrypt the AES key
3. Using the AES key decrypt the content of the EncryptedData/ EncryptionMethod/ CipherValue/
4. Base64-decode the obtained value

More information can be obtained in *XML-Encryption, Syntax and Processing, W3C Recommendation 10 December 2002.*

[4] A method for obtaining the certificate used for encryption MUST be in place in case the NCP has multiple key pairs, by placing the corresponding certificate or the modulus/exponent pair, that can also be used to retrieve it using mechanisms such as XKMS.

4.5 Risk Assessment

The AES algorithm with a key of 128 /192/ 256 bits is strong enough to provide sufficient security, unless a key is being compromised (i.e. discovered). If decryption is not possible due to key corruption or key lost, data availability is affected.
It is not known which data is going to be stored inside the epSOS-NCP persistent memory. This risk assessment will take the worst case, covering healthcare related data.

With a reference to the method of Risk calculation adopted the value was evaluated to medium.

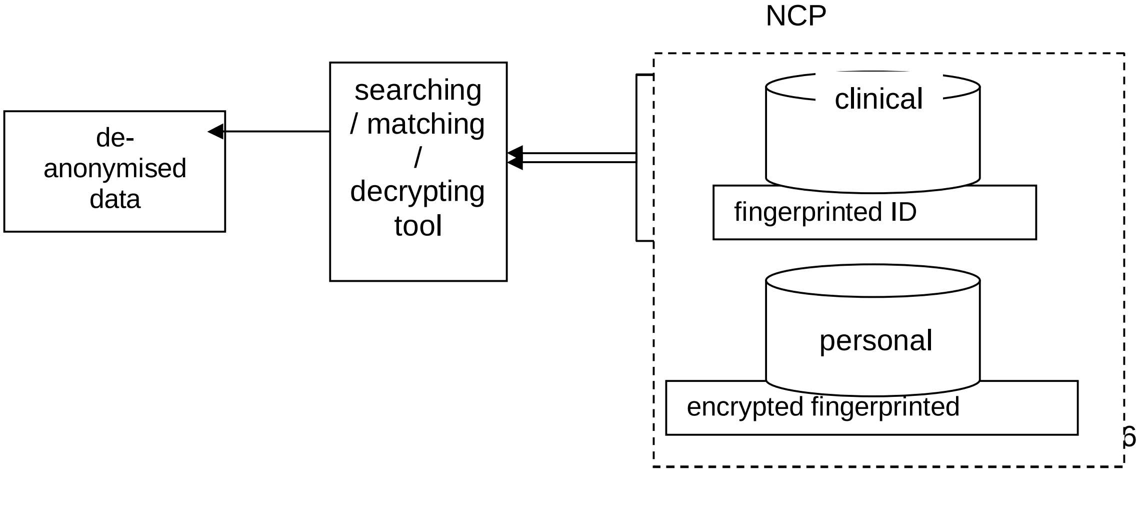
4.6 Pseudoanonymisation

The pseudo-anonymistation enables (in the context of epSOS) the separation of clinical data from personal (or other) data for storing purposes and their recoupling at the moment of need / authorised request. This prevents unauthorised data handling. In combination with encryption it creates the environment for highly secure storing of sensitive data. In case of break-in (hack) or physical stealing of the repository nobody can match the clinical data to any particular person without having the “keys”.

In the concept of epSOS the application of such security means is considered only at NCP level as it is the boundary of the project scope.

4.6.1 Applied pseudoanonymisation

Pseudoanonymisation will be applied in case the NCP has the role and mandate to store the clinical data (either temporarily or permanently). Provided that there is the NCP repository of patients’ data using pseudoanonymisation (and encryption) for the purpose of achieving a higher level of data security, the process of re-coupling – deanonymisation - (and decrypting) is initiated by a received request from the opposite NCP. This request must, of course, contains the unique identification element of the concerned patient (PID).
Each separate part of the stored data (clinical and personal) contains the attribute represented by meaningless identification of the person concerned within the repository, which could be for security reasons generated by several elements including the PID, name, country, ... (the complexity should be defined) and this all is “fingerprinted”. At least at one level (either of personal or clinical data) this fingerprinted attribute must be encrypted to prevent unauthorised matching of fingerprinted attributes.
Separated data are put together in one file and sent off to the authorised requester (in the scope of epSOS it is always the opposite NCP) without being stored; only logged for audit trail.



4.7 Data Confidentiality – Option Analysis

From the results of the questionnaire, pseudoanonymisation is a mechanism not accepted in almost all member states. Therefore, confidentiality SHALL be achieved by means of encryption. Each member state is free to choose one of the two methods proposed by this security service.

	Asymmetric Encryption	Symmetric Encryption
Key length	With RSA the length can be up to 1024,2048,4096 bits.	With AES the key length can by up to 128, 192, 256 bits.
Performance	Poor performance, used mainly to exchange encrypted symmetric keys	Better performance
Availability of already existing products	Libraries exist for RSA encryption for all the programming languages. Need to build a custom software for epSOS.	• XML Encryption: libraries exists for all the programming languages, a custom program MAY be implemented for epSOS. • Disk Encryption: file system encryption or full disk encryption hardware and software implementations exist and can be adopted.
Security	Strong	Strong[5]

Table 4-1: Data confidentiality Options Summary

Two key concepts must be taken into account by each Member State while deploying the confidentiality security service:

- Advanced encryption algorithms must be used for providing ``strong” encryption.
- Low computational cost: encryption and decryption algorithms must be efficient

The suggestion for epSOS LSP during the pilot phase is to use symmetric encryption using AES with a key of 128/192/256 bits for each atom of data (i.e. files). In this case the AES key MUST be adequately protected and often changed, in order to prevent brute force and social attacks. If the key is compromised, the risk of loosing data confidentiality is high. Thus, a secure and reliable storage for the key MUST be adopted, a security access policy to limit the access to the key to authorized Technical Staff only MUST be defined, and specific key management procedure, according with the epSOS security policy concepts, MUST be implemented.

Disk encryption can also be used, with an adequate assurance to have backups.

As alternative XML asymmetric encryption may be used. This alternative is discouraged because the method it is not yet wider adopted.

[5] Nowadays, if the key is sufficiently secured, both methods can guarantee the same level of security.

5 Data Exchange Security Service

5.1 Trust Zones of epSOS

Certain components of the overall IT infrastructure (e.g. applications, systems and networks) are protected by security measures (e.g. firewalls, gateways) and form a reliable environment that can be referred to as “Trust Zone”.

Depending on the measures in place, information with different protection requirements can be processed inside these environments. Therefore a trust zone describes a specific IT environment that corresponds to specific protection requirements categories (normal, high, very high). A trust zone can be a collection of applications, IT systems, the underlying network and the used infrastructure or just a single application or IT system embedded in a surrounding infrastructure. The following illustration shows a simplified network plan of the epSOS infrastructure that contains three general trust zones (which will have to be further refined):

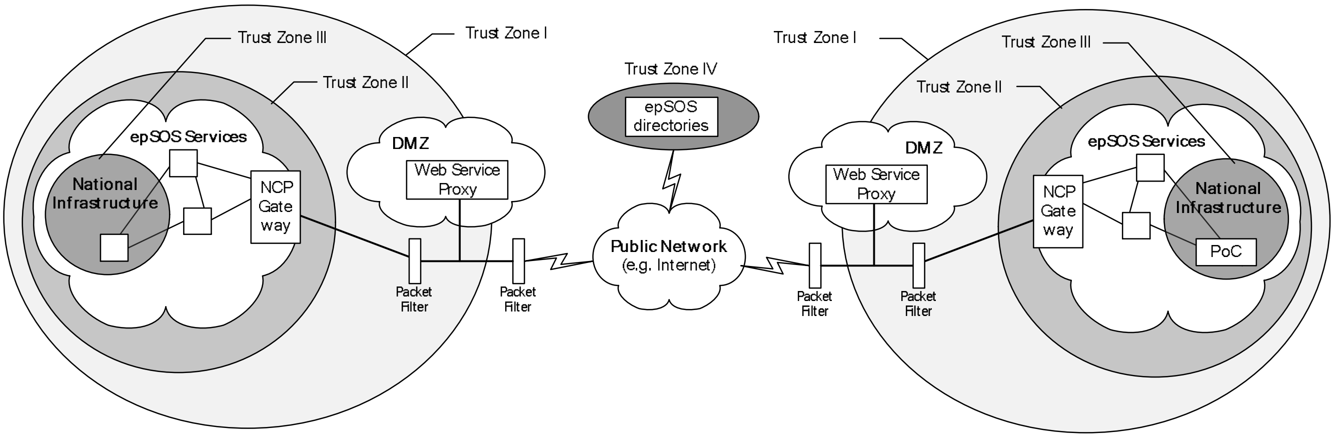


Figure 5-1: General epSOS Trust Zones

Each trust zone can be characterized independently. The following table provides a coarse-grained characterization of the epSOS trust zones:

Trust Zone I	The first zone is formed by the whole internal network of a country. Basic security measures (packet filters and a dedicated internal network infrastructure) protect it from outside threats.
Trust Zone II	The second trust zone is formed by a subset of the first zone. Internal security measures like a logical and physically separated network that is only accessible through a gateway, protect the environment even from internal threats. For epSOS applications epSOS-NCP gateways act as the only entry and exit point from this zone.
Trust Zone III	Trust zone III is a subset of trust zone II. In addition to the measures that are in place to secure the second zone, additional protection is provided. This trust zone corresponds to the existing national eHealth infrastructures which are considered to be secure by definition.
Trust Zone IV	All common epSOS directories and services (e. g. for reporting and management) are located in a separated trust zone. (*This zone was not included in epSOS LSP)*

Table 5-1: Trust Zone characterization

The security measures that are implemented for each of the trust zones correspond to different protection requirement categories. The following table gives an overview on which protection category can be adequately addressed by which trust zone in the case of epSOS:

	Confidentiality	**Integrity**	**Authenticity**	**Non Repudiation**
Trust Zone I	Normal	Normal	normal	n. a.
Trust Zone II	High	High	high	normal
Trust Zone III	very high	very high	very high	very high

Table 5-2: Mapping protection requirements for TrustZones

5.2 End-2-End-Security

The epSOS architecture is characterized by the presence of reusable, loosely coupled services that fulfil the specific business requirements of ePrescription and Patient Summary. The coupling of these services is realized by sending and receiving standardized messages over existing network infrastructures. Message exchange within these infrastructures can be exposed to threats concerning information confidentiality, integrity, liability (non-repudiation) and availability. Particularly with regard to the transfer of messages over public networks like the internet, mechanisms have to be identified that guarantee the complete and continuous protection of the transferred information from the sender of a message to its recipient. Continuous in this context means that no intermediary is able to compromise information security (e. g. read the content, change the content without being noticed) while the message is in transit. This concept is often referred to as “End-to-End Security”.

By settling the ends of “End-to-End Security” on different abstraction layers, the requirements on safeguards and the reachable levels of security change dramatically. For the purposes of epSOS, four levels of “End-to-End Security” will be considered:

- object lifecycle: protected objects are secured throughout their lifecycle from their creation to their consumption without any gaps. E. g. in order to gain lifecycle end-to-end confidentiality for a ePrescription it must be encrypted just after signing and can only be decrypted by an authorised end user. In epSOS the ends of object lifecycle end-to-end security are the HCP in country A, who created a medical document and the HCP in country B, who used this document. Object lifecycle end-to-end security is best suited for matching very high protection demands.

- business transaction: protected objects are secured all the way between two communicating business level applications. E. g. in order to gain business transaction end-to-end confidentiality for a request to an ePrescription managing system, the ePrescription must be encrypted by the managing system using a key known to the end user. In epSOS the ends of business transaction end-to-end security are the HCP in country B and the data managing service in country A. Depending on the additional environmental security measures on data processing and persistence, business transaction end-to-end security can require high and even very high protection demands.
- Message exchange: protected objects are secured when they are persistently stored and when they are exchanged among communicating services. Communication can terminate on any ISO layer as long it is ensured that the way up to layer 7 is properly safeguarded. In epSOS the ends of message exchange end-to-end security are the epSOS-NCPs of country A and B. Depending on the additional environmental security

measures on data processing and persistence, service call end-to-end security can require normal and even high protection demands.

- node-to-node: protection of data transmission between two network nodes, on-node processing, and persistence is considered to be independent from each other. This abstraction is only suited to fulfil normal protection demands.

The layers of end-to-end security can directly by mapped onto the Trust Zones (Figure 5-2):

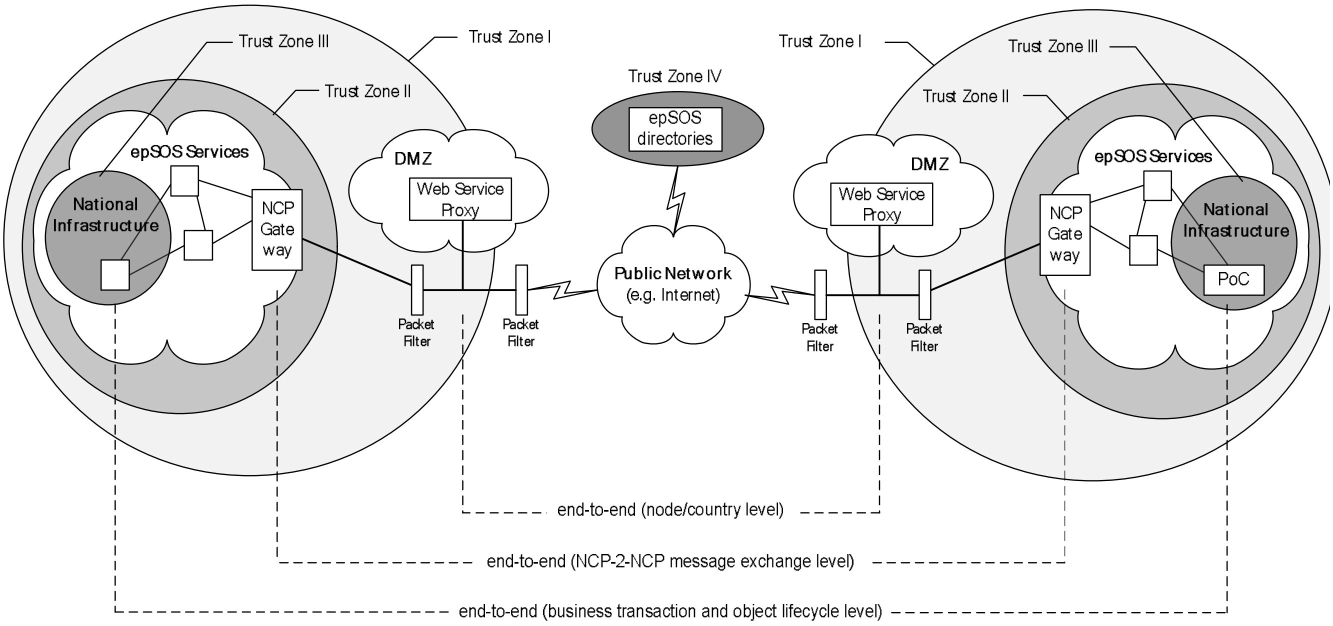


Figure 5-2: End-to-End security and Trust Zones

Given the mapping of protection levels onto trust zones, this leads to the following propositions (see analysis of protection demands):

- Medical Data Integrity: object lifecycle end-to-end security MUST be used if very high protection demands are to be fulfilled (decision making). Both ends MUST terminate in Trust Zone III (*Medical Data used for decision making*) or in Trust Zone II (Medical Data *used for decision support)*.

- Identifiable Data Confidentiality: at minimum message exchange end-to-end security MUST be provided. Communication MUST terminate in Trust Zone II or III.

- Communication Liability: node/country level end-to-end security is sufficient. Communication MAY terminate in Trust Zone I.

- Data Availability: point-to-point security is sufficient. Communication MAY terminate in Trust Zone I

- Entity Authentication: at minimum message exchange level end-to-end security MUST be provided (users do not alter medical data, therefore identity theft does not impose any threats on data integrity). Issuance and verification of Authenticity MUST be located in Trust Zone II or III.

- Originator Authenticity: at minimum message exchange level end-to-end security MUST be provided (contributes to liability). Issuance and verification of Authenticity MUST be located in Trust Zone II or III.

- Access Control: at least at minimum message exchange level end-to-end security MUST be provided (users do not alter medical data, therefore unauthorised access does not impose any threats on data integrity). Attribute provisioning and policy decision/enforcement must be located in Trust Zone II or III.

- Non-repudiation of origin: node/country level end-to-end security is sufficient. Communication MAY terminate in Trust Zone I.

- Non-repudiation of delivery: node/country level end-to-end security is sufficient. Communication MAY terminate in Trust Zone I.

End-2-end protection demands on the lifecycle and business levels must be covered by dedicated security services. End-2-end protection demands on the message exchange and node levels should be covered by Data Exchange Security Services.

Data Exchange Security Services MUST serve confidentiality (including mediation of entity authentication), communication liability (including non-repudiation and originator authenticity) and access control. This can be done by either implementing respective security mechanisms as part of the data exchange protocol or by securely exchanging security tokens through the data exchange protocol.

An analysis of the protection demand of the managed objects can now be done easily by reflecting the already mentioned core protection demands. E. g. a patient identifier has high protection demands on integrity and confidentiality (alteration might provide unauthorized access to another patient's data, disclosure might give information on which physician is treating the patient)

5.3 Data Exchange Security Services

The analysis of the data exchange security services and the underlying security mechanisms and security objects is organised according to the steps of the baseline security process as defined in the ISO/IEC-27001 standard.

5.4 Scoping (IT Network)

The epSOS topology of epSOS-NCPs is a peer-to-peer network of autonomous application level gateways. Each gateway corresponds to a set of service endpoints which mediate requests into the national healthcare infrastructure. Data exchange security services provide message exchange end-to-end security between the sender and the receiver of a message which are both located on the application level (ISO layer 7). In the case of epSOS these are the epSOS-NCP gateways of Country A and Country B.

5.5 Analysis of Alternatives

5.5.1 Security on Different Layers of the OSI Reference Model

The OSI reference model defines an abstract model that describes the layered communication between systems over network protocols. It identifies different layers that use services of underlying layers and provide services to layers on top of them *[ISO7498-1].*

Message Protection (e.g. encryption) must be implemented at least on one of those layers. Which layer is appropriate highly depends on the identified security requirements.

The following illustration shows the integration of security mechanisms on the application, transport and network layer.

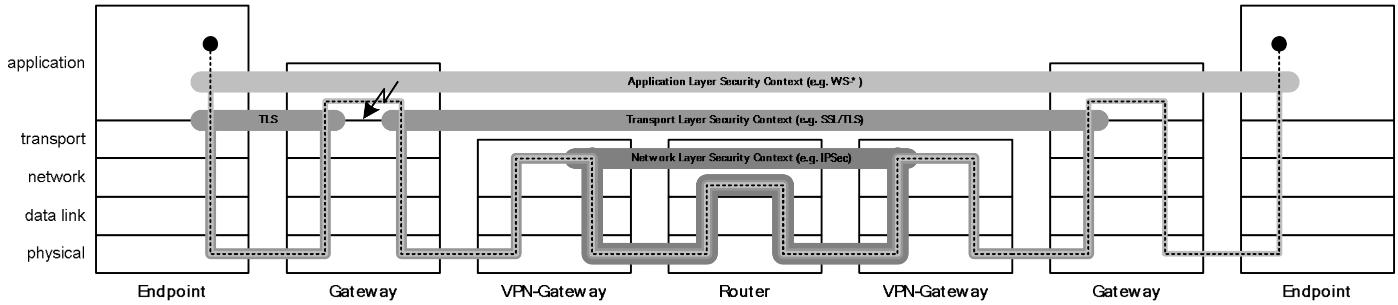

application
transport
network
data link
physical
Application Layer Security Context (e.g. WS-*)
TLS
Transport Layer Security Context (e.g. SSL/TLS)
Network Layer Security Context (e.g. IPSec)
Endpoint
Gateway
VPN-Gateway
Router
VPN-Gateway
Gateway
Endpoint

Figure 5-3: Security Realized on Different Layers of the OSI Reference Model

The number of systems that can be bypassed/bridged by the security context is highly dependent on the layer that is chosen to protect the communication. Information is only protected up to the layer that the security protocol resides in. A processing of information by a protocol in a higher layer makes it necessary to unprotect (e.g. decrypt) the message, process it and protect it again. The security context is interrupted. It is no longer possible to speak of end-to-end security.

The next chapter will provide a detailed overview about specific security mechanisms and their advantages and disadvantages with regard to the layer that they are implemented in.

5.5.2 Mechanisms for realizing End-to-End Security

epSOS builds upon an architecture that is dominated by the exchange of SOAP messages over existing (public) networks. With regard to providing a secure message exchange, it is therefore necessary to identify and evaluate the mechanisms that are appropriate for a web service environment and that provide the requested level of end-to-end security.

The following widely adopted mechanisms will be part of the comparison:

- IPsec (Layer 3)
- Transport Layer Security (SSL/TLS) (Layer 4+)
- SOAP Message Security / Web Service Security (Layer 7)
- Application Based Security (Layer 7)

5.5.2.1 IPsec (Network Layer Security)

Layer / Description:

IPsec is a composition of protocols that protects traffic on the network layer. Especially the Encapsulation Security Payload (ESP), the Authentication Header (AH) and the Internet Key Exchange (IKE) Protocols are suitable for providing protection with regard to most of the identified security objectives. *[RFC4301].*

Confidentiality Mechanisms:

Confidentiality can be guaranteed by using security functions provided by the ESP protocol. *[RFC4301]*

Integrity Mechanisms:

Integrity can be guaranteed by using security functions provided by the ESP or AH protocol.

[RFC4301]

Authenticity Mechanisms:
Authenticity can be realized by using the Internet Key Exchange Protocol. Many authentication methods including one based on public key cryptography and certificates are supported. *[RFC4306]*

Non-Repudiation Mechanisms:
IPsec can secure the communication channel but not single SOAP messages as they are sent between web services implemented within the SPOT framework. In consequence the security objective of non-repudiation cannot be addressed.

Security Scope:
The security context is restricted to communication on the network layer. Therefore only protocols and systems can be bridged that work on the same layer or below. (e.g. switches and routers) If information must be processed on higher layers of the protocol stack the security context must be terminated by unprotecting (e.g. decrypting) the information.

Advantages:
IPSec is well understood and implemented by many software manufacturers. It can be implemented without affecting the source code of the application that relies on the provided security mechanisms. (Security by Declaration)

Drawbacks:
IPsec offers no solution for addressing the security objective of non-repudiation. In addition it is not possible to authenticate endpoints on the level of single web services. Only authentication on the network service level is provided.

The protocols underlying IPsec tend to be complex. This makes it hard to integrate them properly.

Relevant Standards:
RFC4301, RFC4302, RFC4303, RFC4306, ... , PKIX

5.5.2.2 Transport Layer Security (SSL/TLS)

Layer / Description:
The TLS protocol is integrated above the transport layer (Layer 4) and can secure TCP based communication. TLS consists of two layers: the TLS Record Protocol and the TLS Handshake Protocol.

Confidentiality Mechanisms:
Confidentiality can be guaranteed by using symmetric encryption functions. (e.g. AES, DES, RC4, etc.) These functions are implemented by the TLS Record Protocol.

Integrity Mechanisms:
Integrity is provided by a security service of the TLS Record Protocol. This service provides a keyed message authentication digest on the basis of MD5 or SHA.

Authenticity Mechanisms:
The peer's identity can be authenticated using asymmetric or public key cryptography (e.g., RSA, DSS, etc.). This authentication can be made optional, but is generally required for at least one of the peers.

Non-Repudiation Mechanisms:
TLS secures the information channel but not single messages. Digital signatures that might state non-repudiation are only used for authentication. *[Gallop06]* So there are no appropriate mechanisms in place for securing the communication in the context of this security objective.

Security Scope:
Communication security can be provided on a relatively high level. Protocols and systems that work on lower levels (e.g. switches, routers, and proxies) can easily be bridged/tunneled. However, message processing on higher layers makes it necessary to terminate the security context and unprotect the transferred information.
With TLS the security context will always be terminated by a network service that resides in an IT system. But especially in the case of web service communication it might be necessary to terminate the context directly at the web service endpoint.

Advantages:
Transport Layer Security can easily be integrated for providing a secure communication between two peers. Most web service frameworks include this functionality by default. Because of this TLS security can be applied without changing the source code of the application relying on the protocol. (Security by Declaration)

Drawbacks:
TLS offers no solution for addressing the security objective of non-repudiation. Only the information channel is secured but not single messages. Because of this it is almost impossible to prove, that a certain message has been sent. Furthermore only network services can be authenticated by using public key cryptography. An authentication of specific web service endpoints is not realizable. *[Iacono08]*

Relevant Standards:
RFC2246, RFC4346, RFC5246, PKIX

5.5.2.3 SOAP Message Security / Web Service (Layer) Security (WSS)

Layer / Description:
SOAP Message Security (Web Service Security) describes enhancements to SOAP messaging to provide message integrity and confidentiality. *[WSS1.1]* Based upon mechanisms described in [WSS1.1] several specifications and profiles have been published that refer to all of the security objectives identified in chapter 1.1.

Confidentiality Mechanisms:
Message confidentiality leverages XML Encryption in conjunction with security tokens to keep portions of a SOAP message confidential. *[WSS1.1]*

Integrity Mechanisms:

Message integrity is provided by XML Signature in conjunction with security tokens to ensure that modifications to messages can be detected. *[WSS1.1]*

Authenticity Mechanisms:
Authentication is based on the evaluation of security tokens that are embedded into SOAP messages. Several different security token profiles have been specified. (e.g. username tokens, X.509 tokens etc.) *[NIST07]*

Non-Repudiation Mechanisms:
Web service security is implemented on the application layer. This provides the possibility to of auditing and evaluating specific message content that is relevant in the context of non-repudiation. For example the existence of a specific security token (e.g. X.509 tokens) and its usage in conjunction with the XML signature standard can provide the proof that a message was send by a specific subject. *[NIST07]*

Security Scope:
Implementing security mechanisms on the SOAP message level provides the advantage that the security context spans above all systems and services that takes part in transmitting the message from its sender to its recipient. (e.g. routers, proxies, application level gateways) Even systems like intermediate web services that have to process certain parts of the SOAP message can do this without terminating the security context.

Advantages:
Many of the current web service frameworks provide the possibility of enabling SOAP message security by configuring the runtime environment appropriately. (Declarative Security) This makes it relatively easy to address the security needs of certain web services. The implementation of security at the SOAP message level makes it possible to process parts of messages at intermediate nodes even without the need to terminate the security context.

Drawbacks:
A comparison with other mechanisms like TLS shows that the integration of security on the SOAP message level is more expensive i with regard to performance. *[Ekelhart07]*
The realization of security on the application level can only protect information that has its seeds on that level. Additional information generated after protecting the message (e.g. addressing information, routing information) won't be secured and can therefore be analyzed and/or compromised.

Relevant Standards:
WS-Security, WS-SecurityPolicy, WS-Trust, WS-ReliableMessaging; WS-SecureConversation, WS-Federation, SAML, XACML, PKIX, XMLEnc, XMLDSig, XKMS. An overview and description of Web Service Security Standards is given in [SPOT_WSS].

5.5.2.4 Application Layer Security

Layer / Description:
Message security is provided on the application layer (7) of the OSI reference model by protecting the message payload with the help of cryptographic functions.

Confidentiality Mechanisms:
Confidentiality can be guaranteed by individually protecting the content/payload of a message. (e.g. with the help of cryptographic functions for information encryption)

Integrity Mechanisms:
Integrity can be guaranteed by individually protecting the content/payload of a message. (e.g. with digital signatures)

Authenticity Mechanisms:
Authenticity can be guaranteed by individually protecting the content/payload of a message. (e.g. with digital signatures that are based on certificates/keys corresponding to a specific subject)

Non-Repudiation Mechanisms:
Non-repudiation can be guaranteed by individually protecting the content/payload of a message (e.g. with digital signatures that are based on certificates/keys corresponding to a specific subject) in conjunction with auditing mechanisms.

Security Scope:
The scope of this security mechanism is really broad. Information can be secured across any system participating in the communication. (e.g. Proxies, Application Level Gateways, Web Services). Security can even be maintained while information is in storage.

Advantages:
The implementation of security on the level of messages content provides the advantage that mechanisms can be integrated that are matching the specific needs of underlying application or data. Security mechanisms are not specific to data in transit. They can also be used to protect information in storage.

Drawbacks:
Security mechanisms are specific to certain applications and can hardly be reused for other purposes. Security is often applied programmatically. This makes it necessary to change the source code of an application. If the integrated functionality is not based on common standards it’s hard to evaluate if the integrated mechanism addresses relevant security objectives properly.

Relevant Standards:
XMLEnc, XMLDSig, PKCS#7, PKIX, PGP

5.6 Evaluation of the Alternatives

The preceding sections described mechanisms that are able to protect communication on different layers of the OSI reference model. In this section they are analysed with respect to the epSOS requirements on the different security objectives.

5.6.1 Confidentiality

End-to-end message confidentiality can be provided by transport layer, message layer and application layer security mechanisms. The drawback of message and application layer mechanisms not protect addressing information is of no importance for epSOS because communication is always only epSOS-NCP-to-epSOS-NCP.

	IPSec (Network Layer)	SSL/TLS (Transport Layer)	SOAP/WS Security	Application Layer
Appropriateness of provided mechanisms	-	+	+	+
Matching of terminating Trust Zones	-	-	+	+

Table 5-3: Comparison of pros/cons for confidentiality mechanisms

5.6.2 Integrity

Even though the integrity of medical data has to be addressed by specific means, a violation of the integrity of other arguments (e. g. a patient identifier in a query) might lead to unauthorized disclosure of medical data. Therefore the protection demand on the integrity of message exchange is high; communications ends must be located in Trust Zone I or II.
End-to-end message integrity can be provided by network layer, transport layer, message layer and application layer security mechanisms.

	IPSec (Network Layer)	SSL/TLS (Transport Layer)	SOAP/WS Security	Application Layer
Appropriateness of provided mechanisms	+	+	+	+
Matching of terminating Trust Zones	-	-	+	+

Table 5-4: Comparison of pros/cons for integrity mechanisms

5.6.3 Authenticity

If end-to-end authenticity is required, mechanisms on the transport, message and application layer are appropriate. But as the epSOS security policy links certain guarantees with the signature of the epSOS-NCP it is important that epSOS-NCP authenticity is precise, which means that a epSOS-NCP gateway MUST have its own and exclusive service certificate and the only matching termination points are epSOS-NCPs. This cannot be realised by IPSec or TLS, which rely on node certificates.

	IPSec (Network Layer)	SSL/TLS (Transport Layer)	SOAP/WS Security	Application Layer
Appropriateness of provided mechanisms	-	-	+	+
Matching of terminating Trust Zones	-	-	+	+

Table 5-5: Comparison of pros/cons for authenticity mechanisms

5.6.4 Liability (Non-repudiation)

Non-repudiation can only be provided by mechanisms on the message and application (message payload) level. Based on certain requirements it must be decided which mechanism is appropriate. The realization of this security objective on the application level provides the advantage of keeping non-repudiation information beyond message transfer (e.g. in storage) without the necessity of protocolling whole SOAP messages.

	IPSec (Network Layer)	SSL/TLS (Transport Layer)	SOAP/WS Security	Application Layer
Appropriateness of provided mechanisms	-	-	+	+
Matching of terminating Trust Zones	-	+	+	+

Table 5-6: Comparison of pros/cons for non-repudiation mechanisms

5.7 **Conclusions**

Security mechanisms on the message layer (WSS) as well as on the application layer are appropriate to provide complete end-to-end communication security for each of the identified security objectives. While SOAP/WSS security terminates at Trust Zone II, application layer mechanisms can even be tied into Trust Zone III.

In general it is easier to implement SOAP message security (web service security) by just declaring the specific security needs inside the web service framework (declarative security). In contrast application based mechanisms make it necessary to adjust the source code (programmatic security).

5.8 **Option Presentation.**

Taking into account that end-to-end node communication at MS country level, due to public network interfaces, must in every case be protected by TLS/SSL protocol, and under the constraints that in each MS domain communication between Trust Zone I and Trust Zone II should be protected by dedicated VPN, two possible solutions are analyzed also considering operational and economical factors.

Option	WS-*	TLS	VPN	Confident.	Non-rep.	Maturity	Complexity	Performance	Segr. of Duty
“A”	enc/sig	enc	-	+	+	-	-	-	-
“B”	sig	enc	enc	+	+	+/-	+/-	-	+

Table 5-7: Comparison of pros/cons for (WS-*enc/sig +TLS) vs (WS-*sig +TLS+VPN)

The main advantage of option “A” is related to the lack of confidentiality protection needs between Trust Zone I and Trust Zone II communication. This fact may greatly reduce the complexity of security conformity tests of each NCP implementation.

On the other hand, option “B” presents advantages in performance, in development complexity and it is a more mature solution. Moreover the segregation of duty in operational environments is easier to achieve, because the VPN is on network level which is frequently administered by different staff from the Webservice infrastructure.

5.9 Option Selection.

From the security point of view both options “A” and “B” guarantee an adequate level of protection.

WP3.4 SHOULD analyse the options from 5.8.

As the above mentioned solutions just apply to the message payload level, the following additional mechanisms MUST be provided:

- To reach the required level of liability, audit trails MUST be written to accompany the reliable transmission of data.
- To support the security objectives of access control and HCP authenticity, respective security tokens MUST be exchanged securely as part of the SOAP Security Header.

5.9.1 TLS version to be adopted.

Three versions of TLS protocol are available on the market, with a different level of diffusion: (TLS 1.0 rfc2246, TLS 1.1 rfc4346, TLS 1.2 rfc5246).

- TLS 1.0 using SHA-1 algorithm, is the most widely adopted.
- TLS 1.1 was updated from the previous version 1.0, adding, as major significant difference, the protection against Cipher block chaining (CBC) attacks.
- TLS 1.2 significantly improves the previous versions, the *cryptographic* algorithm and provides support for 256 bit Hash functions (SHA-2).

Having in mind the timeframe of epSOS pilot phase, to be concluded within 2012, TLS with SHA-1 may be considered as sufficient . Therefore the indication given is to adopt the TLS version more widely implemented (TLS 1.0 RFC 2246).

6 Auditing& Accounting Security Service

6.1 Security Service Objective

This section describes the security services objective. The first part of this paragraph deals with the definitions of Audit and Accountability and the reason why they are needed in the epSOS scenario.
The second part will explore the transactions between the epSOS-NCPs and the consequent events and information to record, referring to ATNA (Audit Trail and Node Authentication contained in IHE IT Infrastructure (ITI) Technical Framework Volume 1 (ITI TF-1) Integration Profiles Revision 5.0 and RFC3881 (Security Audit and Access Accountability Message XML Data Definitions for Healthcare Applications).

6.2 Security Service Definition and need

Audit is defined as the analysis of actions and states of a system or a user. This implies gathering complete and time-referenced information about the actions related to the system that has to be audited.

The means used to keep track of this information is called “Log”. Logs are therefore evidence of any kind that can provide support to the audit.

Because during the analysis of the Logs, when applicable, it’s important to clearly identify “who” has performed the action (accountability), to make it possible, the user’ (active epSOS entity) must be identified and authenticated.

EpSOS provides processing and transmission of health data that has legal implications for the actors involved in the system, for these reasons it’s also important to audit transactions to identify any deficiencies related to security.

In “WP3.7 D 3.7.2 “Master document” the following specific security requirements have been identified for this Security Service:

- ***EpSOS-Req#3.7.10 (Accounting);***
- ***EpSOS-Req#3.7.11 (Auditing);***.
- ***EpSOS-Req#3.7.12 (Fraud detection***
- ***NCP-Req#3.7.11 (Accounting and Control***;
- ***NCP-Req#3.7.12 (Auditing);***
- ***NCP-Req#3.7.13 (fraud detection);***
- ***NCP-Req#3.7.14 (Continuously Logging);***
- ***NCP-Req#3.7.15 (Securing Access to Audit/Account Logs);***
- ***NCP-Req#3.7.16 (Logging Transactions);***
- ***NCP-Req#3.7.18 (Minimum Content of Accounting Logs);***
- ***NCP-Req#3.7.19 (Reporting Every Access to medical information – notifications included-).***

The objective of Audit and Accounting security service is to provide a history of transactions and to ensure that it is possible to trace who has performed any action involving a epSOS-NCP transaction.
The service MUST also ensure to be tamper-proof to avoid the historical data to be deleted or tampered with, even by Technical Staff (system administrators).

The following picture illustrates the structure of the system and how the dataflow between epSOS-NCPs and its MS Information Systems converge towards the audit system.

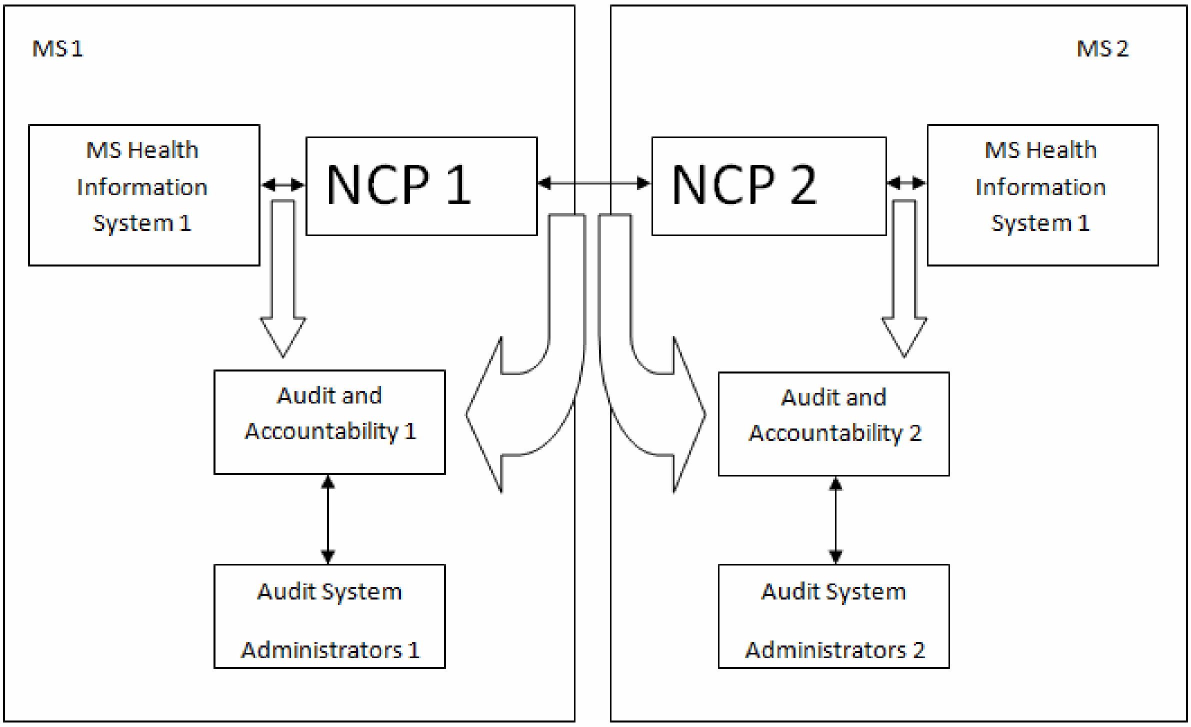

MS 1
MS 2
MS Health Information System 1
NCP 1
NCP 2
MS Health Information System 1
Audit and Accountability 1
Audit and Accountability 2
Audit System Administrators 1
Audit System Administrators 2

Figure 6-1: System Data Flow to the Audit System

6.3 Security Service Scope

The scope of this security service, coherently with the epSOS project is restricted to the transactions defined to exchange Patient Summary as described by UC 1–2 and ePrescription - eDispense as described by UC 1. (ref. WP3.1/ WP3.2 Final documents)

In order to ensure the registration of all activities, every NCP secure audit system has to Log and account the following events, referring to Data Exchanges of epSOS Architecture in WP 3.3:

- HCP Authentication
- Patient Identification and Authentication
- HCP Authorization
- Medical data Query and Retrieve
- Medical Data Update (Dispensation notification)

All these events MUST be logged independently for the result (successful or not) of the corresponding action.

Moreover the NCP secure audit system has to LOG the following system and network infrastructure events:

- NCP / Security Service start up/shut down;
- Usage of secure Audit log (other than audit log record creation);
- Access policy change (ie. Network, file system);
- User account event (ie. Create account);
- Configuration changes;

- Partial failure;
- PKI event;
- Timing synchronization;
- Authentication (NCP, Technical Staff) failure and success.

As previously mentioned, the goal of this security service is to record every transaction to and from the epSOS-NCP, both in the country of the member state and the other one.
This includes every request and response, limited to the events described above, so all information of all profiles involved in the transactions will be audited.

Timing services ensures that there is a synchronization between the records of the various countries, so that they are aligned if it is necessary to make a cross-check between records recorded in two or more countries.

6.4 Requirements and Constraints

Taking into account the objectives of the service, it’s important to notice that to achieve them it’s not necessary to analyze and audit the content of requests and responses but just the identity of the sender and receiver, and other elements relating to the transaction itself (e.g. timing references, validation result, etc...)

For this reason, even if the transaction’s payload contains “healthcare-related data”, they WOULD NOT be stored in any way in the Log.

For the same reasons, a similar handling should be foreseen for personal identifiers, with the exception of a mapping registry in the epSOS-NCP of the same country domain as the actor's origin.

The case of transmission of non-explicitly healthcare-related data, but data that by its nature could lead to the identification of the health status of the citizen has to be considered. In such a case the use of anonimization mechanisms MUST be adopted.

The data Log of each epSOS-NCP service MUST be self-consistent.
As regards a scenario in which it would be necessary to verify whether the data from a epSOS-NCP has been tampered with or not, in that case a cross-check should be needed with other epSOS-NCP engaged in transactions’ subject to the alleged irregularities.

Another important consideration to bear in mind is the need to separate the LOG system from the epSOS-NCP that has to be audited, and the same goes for the administrators of each system that must be users distinct among them.

6.5 **Risk Analysis**

The Audit & Accounting security service has been analyzed to discover the presence of vulnerabilities. For each vulnerability found the possibility of exploiting the vulnerability is studied and then an acceptable risk level is defined.
Based on the results of the vulnerabilities analysis, and risk level definition, the security measures that may implement the security service are identified and described.

6.5.1 Vulnerability Assessment

Two attack scenarios might be identified: online and offline attacks.
The former are perpetrated at the time of the transaction and are designed to prevent it from being recorded, the latter are intended instead to alter or delete records which have already occurred.

- Audit Data missed/lost due to :
  - Malfunctioning of storing system
  - Incorrect administrator operations
  - Malicious administrator operations

Audit Data stolen, tampered with or deleted or modified with an added false entry due to:

- Malfunctioning of Access Control
- Incorrect administrator operations
- Malicious administrator operations

6.5.2 Attack scenarios analysis.

Possible attack scenarios are analyzed to highlight the risks correlated to the secure audit trail.

Audit data missed may be used to hide illicit actions by epSOS Users: (HCP and Technical staff)

The interest for HCP in hiding illicit transactions made by him could be a motive to tamper or delete audit records or, if the action was done with premeditation, HCP may want to prevent the registration of transactions, for example, in order to achieve insurance fraud.

To achieve this aim, the HCP could find and exploit a vulnerability of the audit system or try to bribe the administrator. This second hypothesis is probably the most accredited and the easiest to implement.

As a matter of fact, the audit system Administrator could have the motive to steal, delete or tamper with data on commission or in collusion with HCP.

6.5.3 Acceptable risk level definition

By making reference to the Risk calculation method adopted, and taking into account that the possible attacks are not very likely for all the vulnerabilities, a level of medium Risk was evaluated.

To decrease this Risk, the Audit & Accounting security service, shall implement specific architectural, technical, and organizational security measures.

6.6 Identification measures:

The following requirement shall be taken into consideration when implementing this security service:

- the audit systems (Log data server) must be segregated from the audited system (epSOS-NCP) itself;
- the epSOS-NCP has to be provided with an agent which sends audit data in real time to a secured Log data server;
- Log data format will follow indications in RFC3881 (Security Audit and Access Accountability Message XML Data Definitions for Healthcare Applications).
- the secured Log data server may store received data and relative hash or digital signature on a worm (Write Once Read Many) device, or equivalent method, to prevent offline tampering and deleting;
- the system administrators and the HCPs must be distinct people with no hierarchical relationship between them ;
- On the Log data server, a statistical analysis via business intelligence or data mining products, of the Logged data has to be provided in order to allow early discovery of any misused or fraudulent use of epSOS. (using commercially available products, to be customized, such as Oracle Data Mining or Business Objects)

The data to be logged for: HCP Authentication, Patient Identification / Authentication and consent, HCP Authorization transactions: follow the definition of WP 3.6.

The data to be logged for: Medical data Query and Retrieve and Medical data Update transactions: follow the definition of WP 3.4.

The data to be logged for System and network infrastructure events: follow the definition of WP 3.3 & WP 3.4.

Every data Logged Must in every case be compliant with the following requirements:

- For every transaction, stored information may include information contained both in the request and in the response.
- For every record created by the Audit system, a timestamp of date and time must be recorded from time services, which ensures synchronization between different countries.
- HCP and Patient IDs are not immediately associable with the personal information of their owners because this kind of information is not stored on the secure audit system. Should any information stored in the audit system lead to patient identification, such information MUST be kept in separated tables from the ones containing the IDs identifying transactions and operations.

- The purpose of the audit system is not to store healthcare-related data within the transactions, which is why this type of data should not be stored in any way on the systems.
- However, since the system may be able to provide evidence that a transaction has been requested or not, a cryptographic function (ie hash or the encryption in a way that the decrypting key should not be under the direct control of the NCP) of the health/health-related transmitted data may be stored.
- The data Log MUST NOT be used for different purposes than:
  - to analyze or to prevent a breach of security;
  - to respond to the right request of the data subject (patient) or to "National entities" legally authorized, with the aim of providing the evidence necessary to the case.
- Finally, to ensure that logs would not be tampered with, a sign mechanism to sign every log record may be adopted. On top of this a sign mechanism of the whole daily list of log entries can be put in place in order to ensure integrity of the entire dataset.

6.7 Secure storage of audit trails -Technical Requirements

The system devoted to audit collection must guarantee a high degree of security itself, both from the data transmission and data storing point of view. In other words a tamper-proof system has to be designed.
In order to design such a system**,** WP 3.3 WP 3.4 and WP 4.2 must satisfy the following technical requirements.

1. the PHYSICAL system(s) hosting the audit collection processes and the audit data (logs)must be physically protected (closed environment, access control);
2. the MACHINE(s) hosting the audit collection processes and the audit data (logs) must be UNACCESSIBLE by Technical staff users;
3. users allowed/entitled to access the audit system will ONLY have the right to access in READING the logs, without having access to other system functions
4. audit data (logs) must be stored on non-modifiable supports (WORM) or equivalent method;
5. communication between epSOS-NCP and the Audit trails secure storage should be secured and both systems should also authenticate each others.

For Example, technical req #2 can be met, setting a machine and deleting all the privileged users (local admins) account. In such a way there will be no possibility of altering the machine configuration. In case this option (change configuration) is raised, the machine has to be re-installed.

6.8 Audit & Accounting – Option Analysis

ATNA accepts two possible solutions as Audit record transportation between epSOS NCP and the data Log server:

- Transmission of syslog message (Log record) over TLS (RFC5425) with the syslog protocol (RFC5424).
- Transmission of syslog message (Log record) over UDP (RFC5426) with the syslog protocol (RFC5424).

The advantage of the former solution is related to the security (confidentiality and integrity) of the TLS protocol, but as contra, in case of protocol failure, delay or block of the epSOS NCP primary functions may occur. To avoid this situation a specific recovery procedure must be adopted with complexity and increasing cost.

The latter solution has the advantage of being easier to implement, and having a better performance, UDP being a stateless protocol, any fault of the protocol has no impact on epSOS NCP.
As a drawback this solution does not guarantee the confidentiality and the integrity of the message (a log record may be missed without notice). The drawback may be reduced by implementing a read after write strategy, and adopting a dedicated point to point connection between epSOS-NCP and the data Log server.

The adoption of UDP protocol with the suggested safeguards seems to be a better solution for the piloting phases of the epSOS LSP**.**

6.8.1 Pilot phase simplification

To reduce the cost and the complexity of the epSOS NCP technical implementation, the requirement for a separate secure storage of audit trails could be relaxed.
The waiver of a separate secure storage system must be compensate by specific organizational and procedural security measures, and must be confirmed by a risk analysis.

7 Non Repudiation Security Service

7.1 Introduction

Repudiation is one of the fundamental security issues existing in paper-based and electronic environments. Dispute of transactions is a common issue in the business world. Transacting parties want to seek a fair settlement of disputes, which brings the need for non-repudiation services in their transactions. The motivation for non-repudiation services is not just the possibility that communicating parties may try to cheat each other. It is also the fact that no system is perfect, and that different and unexpected circumstances can arise in which two parties end up with different views of something that happened. Network failure during the protocol run is a representative example.

We define a basic transaction as the transferring of a message M (e.g., electronic goods, electronic cash, or electronic contracts) from user A to user B, and represent this event with the following message flow: A → B : M. Thus, typical disputes that may arise in a basic transaction with a deadline T could be:

- A claims that it has sent M to B, while B denies having received it;
- B claims that it received M from A, while A denies sending it; and
- A claims that it sent M before T, while B denies receiving it before T.

Fair non-repudiation can be considered as an extended fair exchange problem in which non-repudiability is made an integral requirement of the exchange (in general, it may not be required). We can find various instances of the general exchange problem in different types of commercial activities: purchase, contract signing, certified mail, or, more generally, in any barter conducted by means of digital networks.

An exchange is said to be fair if, at the end of the exchange, either each player receives the item it expects or neither player receives any additional information about the other's item. For instance, in payment protocols, fair exchange can ensure that a customer receives digital goods from a vendor if and only if the vendor receives payment from the customer.

The features of the transaction will decide the type of non-repudiation service to be deployed. For any non-repudiation service, evidence is a crucial object, and the processing of evidence usually involves the assistance from Trusted Third Parties (TTPs). There are different activities at each phase of processing. The non-repudiation policy defines the behavior of these activities. Finally, the eventual success of non-repudiation depends upon technical and legal supports.

Non repudiation is, thus, one of the essential security services in computer networks defined by the ITU in X.813 [ITU-T X.813 1996]. In this document we establish the characteristics of this security service and also analyze this service when multiple entities are involved.

7.2 Specific Non-Repudiation Services

Non-repudiation services help the transacting parties to settle possible disputes over whether a particular event or action has taken place in a transaction. We define a non-repudiation protocol as a message flow in which entities exchange digital evidence in order to provide such non-repudiation services.

In an electronic transaction, message transfer is the building block and there are two possible ways of transferring a message.

- The originator O sends the message to the recipient R directly; or
- The originator O submits the message to a delivery agent D which then delivers the message to the recipient R.

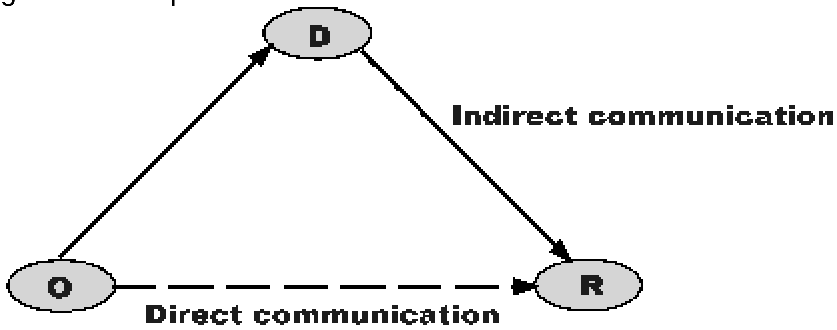


Figure 7-1: Models of Message Transfer

In the direct communication model, because the originator and the recipient possibly do not trust each other, the originator is not sure that the recipient will acknowledge a message it has received. On the other hand, the recipient will only acknowledge messages it has received. In order to facilitate a fair exchange of a message and its receipt in which neither party will gain an advantage during the transaction, a TTP will usually be involved. Of course, the extent of the TTP's involvement varies among different protocols, which allows the provision of a protocol distinction.

To establish accountability for the actions of originator and recipient, the following non-repudiation services are required:

- Non-repudiation of Origin (NRO) is intended to protect against the originator's false denial of having originated the message. Evidence of Origin (EOO) is generated by the originator or a TTP on its behalf, and will be held by the recipient.

- Non-repudiation of Receipt (NRR) is intended to protect against the recipient's false denial of having received the message. Evidence of Receipt (EOR) is generated by the recipient or a TTP on its behalf, and will be held by the originator.

In the indirect communication model, a delivery agent is involved to transfer a message from originator to recipient. In order to support the settlement of possible disputes between originator and delivery agent or between originator and recipient, the following non-repudiation services are required.

- Non-repudiation of Submission (NRS) is intended to provide evidence that the originator submitted the message for delivery. Evidence of Submission (EOS) is generated by the delivery agent, and will be held by the originator.

- Non-repudiation of Delivery (NRD) is intended to provide evidence that the message has been delivered to the recipient. Evidence of Delivery (EOD) is generated by the delivery agent, and will be held by the originator. Similarly, we should be aware that evidence provided by this service cannot be used to make further deductions about the delivery status without some sort of assumption on the communication channel.

7.3 **Evidence**

The evidence is the data or information that can be used if a dispute arises. It can be either generated and stored by the local user or by a third party. Its format depends on the cryptographic mechanisms agreed in the service, such as digital signatures (public-key cryptography) and secure envelopes (secret-key cryptography). Whichever the format is, this evidence has to be based on common information that helps to clearly identify a transaction and thus resolve a possible dispute in a more deterministic way. Some of these common elements are:

- non-repudiation service to which evidence is related;
- non-repudiation policy identifier;
- originator identity;
- recipient identity;
- third-party identity if evidence generator differs from the originator;
- message or a digital fingerprint;
- message identifier;
- information needed for verifying evidence (i.e., digital certificate, symmetric secret key information) if it is not publicly available;
- TTP’s identifier and role (see 4) when involved in the service;
- unique evidence identifier; and
- time information (time and date in which evidence was generated, expiry date, etc.). If this data is certified by a Time-Stamping Authority (TSA), it could include a timestamp service identifier.

When a secure envelope is used to provide evidence, data is stamped with a secret key known only by the TTP, thus being the generator and verifier of evidence as requested by users.

TTP participation can be relaxed through the use of smartcards or manipulation resistant modules [ITU-T X.813 1996] in which secret keys are properly installed. In this case, the smartcard plays the role of a distributed TTP. The generator smartcard is used for evidence generation, while the verifier’s is only for validation. The latter cannot be used to generate evidence with the secret key (even if it is the same one), so that only the user who owns the generator smartcard could have created the evidence. This is achieved by correctly installing the secret key and the module which controls whether the user can use its smartcard for generation or verification. This module is tamperproof and different for the generator and the verifier so that it performs just one of the two possible functions.

The secure envelope maintains integrity of the information using a digital fingerprint (i.e., hash function) and confidentiality (e.g., symmetric cipher with the secret key). When a digital signature is used to provide evidence, information is enclosed in a data structure digitally signed by a Certification Authority (CA) such that only the CA can sign the data and other participants (recipients and TTP) can verify it. Unforgeable digital signatures provide a clear statement of the essential components of handwritten signatures, namely, a user’s ability to sign by itself, a universally agreed verification procedure, and the assertion that it is unfeasible (or at least very hard) to selectively forge signatures in a manner that passes the verification process without being detected.

In order to bring all of this into reality, digital signatures used as evidence in a non-repudiation service need an infrastructure backing them up. There will be a third party certifying participants' link between their identity and public key. Only in this way can any recipient verify the digital signature. Digital signatures introduce a new disrupting element in the non-repudiation service, as the link certified by the CA (often referred to as digital certificate) may have an expiry date. This fact has to be checked when evidence is verified either by the recipient or a TTP (e.g., an adjudicator). If this link has expired, evidence will be valid only if it was generated before. For this reason, time information has to be included in the evidence generated.
In general, it is more efficient, in terms of computation, for users to use secure envelopes with symmetric encryption techniques, since the TTP (or smartcard) is in charge of the generation/verification process. Nevertheless, in this case, the following holds.

- Principals have to unconditionally trust a third party for evidence generation and verification;

- TTP's online availability is needed in order to participate in the service when requested; and

- if users are to relax the TTP participation as stated previously, then they need to use dedicated hardware to avoid the TTP becoming a bottleneck.

So, users would likely prefer to use digital signatures for the following reasons.

- It only needs an implicit trust over the CA computing the digital certificates. But this trust can be relaxed with legal agreements between users and authorities, audited registration processes, and a quite advanced standardization [ITU-T X.509 2000].

- Trust imposed over the CA is less critical than in the former case, since it certifies the existence of a binding between a user and a public key, verifying at the same time that this uniquely corresponds to a private key. But this TTP need not know the key itself. So, there is no danger of this entity accessing the content, or even being able to generate it (as with secure envelopes).

Additionally, Maurer [2004] proposed a novel view of digital evidence called digital declarations, based on a digital recording of a wilful act indicating agreement to a document or contract. This proposal tries to address some of the problems mentioned before that digital signatures bring with them. It also includes new elements in the digital evidence (as wil~~l~~ful acts) to augment the concept of evidence, bringing it nearer to that used in human judgments. Among all the concepts introduced by Maurer, the semantics of certificates is very important. He proposes that when registering the public key, the user must explicitly commit to be liable for signatures with respect to this public key. Evidence confirming this commitment, designated as change, has several important implications for us.

- The certificate has absolutely no value as evidence in court, only the commitment declaration does.

- Only the recipient of a signature (evidence) must trust the CA.

- An expiration date stated on the commitment declaration must be interpreted differently. It specifies until what time evidence can be presented as valid, regardless of when generated. In other words, evidence expires, not public keys. As a consequence of this view, the validity period of evidence should be kept short.

- A commitment declaration cannot be revoked. Revocation of a public key is impossible (not needed).

Actually, with these definitions, the signature seems to be more insecure than in the traditional view when revocation is possible while the commitment declaration is valid. But, on the other hand, it seems to be closer to the business model if we consider the discussed users' liability in the traditional approach. Maurer [2004] proposes the concept of delegation signatures (digital signatures assisted by TTPs) to strengthen its security. Furthermore, this digital declaration and commitments comprise a new approach to digital evidence, with no implications on how non-repudiation protocols handle the evidence.

7.4 Role of the Trusted Third Party

One of the main features which allows us to classify TTPs is their role in a non-repudiation service. A TTP which does not participate actively in the non-repudiation service (i.e., it will be invoked only when there is something wrong in a transaction) is referred to as offline TTP. An online TTP participates in the generation and verification of evidence throughout the protocol instance. An inline TTP acts as an intermediary in all interactions among users. There is a difference between third parties that are used only in case of exceptions and those that are actively involved in a protocol. Obviously, the first type is preferred if efficiency is the major concern, but in some situations and e-commerce applications, to have a delivery agent or intermediary could be the best practical solution.

Other roles have appeared as a consequence of research achieved in the domain of exchange protocols. These new approaches aim at eliminating the involvement of the TTP completely, but need strong requirements; either all involved parties must have the same computational power, as in gradual exchange, or fairness depends on the number of protocol rounds, as in probabilistic protocols.

Finally, an additionally existing trusted third party is the adjudicator. This is the party which drives a resolution process to a conclusion, depending on evidence presented by the entities and, optionally, contacting the TTP which participated in the protocol. In order to facilitate its task, a well-defined dispute resolution process in accordance with the non-repudiation policy must exist. This dispute resolution process has to take into consideration the legal framework in which it is defined. New or established Online Dispute Resolution (ODR) processes can be used.

7.5 Non-Repudiation Phases

Non-repudiation services establish accountability of an entity related to a particular event or action to support dispute resolution. Provision of these services can be divided into different phases, such as generation, transfer, verification, storage, and dispute resolution.

7.5.1 Evidence Generation

Evidence generation is the first phase in the provision of a non-repudiation service. Depending on the non-repudiation service being provided and the non-repudiation protocol being used, evidence could be generated by the originator, the recipient, and/or the TTP. The elements of non-repudiation evidence and the algorithms used for evidence generation are determined by the non-repudiation policy in effect. When NRO and NRR services are required, evidence of origin and receipt are usually generated by the originator and the recipient, respectively, if digital signature is used for evidence generation. When NRS and NRD services are required, evidence of submission and delivery will be generated by a TTP, like a notary or a delivery authority. If a secure envelope is used for evidence generation, it should always be generated by a TTP on behalf of the originator or recipient.

A TTP may also generate and provide supporting evidence in a non-repudiation service. For example, in a fair non-repudiation protocol, the notary will digitally sign the message key provided by the originator and make the confirmed message key available to both originator and recipient. The confirmed message key will serve as part of non-repudiation evidence to prove that the message key was sent from the originator (via the notary), and is available to the recipient.

7.5.2 Evidence Transfer

Evidence transfer is the most challenging phase in the provision of a non-repudiation service. It mainly consists of the sending and reception of evidence among participants. Actually, it represents the core of a non-repudiation protocol. It is greatly influenced by the communication channel properties. The different options are as follows.

- The communication channel is unreliable. In this case, data can be lost.
- The communication channel is resilient (also called an asynchronous network). In this case, data is delivered after a finite but unknown amount of time.
- The communication channel is operational (also called a synchronous network). In this case, data is delivered after a known, constant amount of time.

An unreliable channel will in most cases be transformed into a resilient channel by the use of appropriate transport protocol (e.g., retransmissions).

7.5.3 Evidence Verification

Newly received evidence should be verified to gain confidence that the supplied evidence will indeed be adequate in the event of a dispute arising. The verification procedure is closely related to the mechanism of evidence generation.

If evidence is generated through a secure envelope, it should be verified by a TTP at the request of the user because the secret key for evidence generation and verification is only held by the TTP. Obviously, the extra communication between user and TTP will cause a substantial delay, which might be unacceptable for many online electronic transactions.

7.5.4 Evidence Storage

Because loss of evidence could result in the loss of a future possible dispute resolution, verified evidence needs to be stored safely. The duration of storage will be defined in the non-repudiation policy. For extremely important evidence aimed at long-term non-repudiation, it could be deposited with a TTP.

7.5.5 Dispute Resolution

Dispute resolution is the last phase in a non-repudiation service. This phase will not be activated unless disputes related to a transaction arise. When a dispute arises, an adjudicator will be invoked to settle the dispute according to the non-repudiation evidence provided by the disputing parties and the non-repudiation policy in effect. This policy should be agreed in advance by the parties involved in the service.

The adjudicator needs to verify the evidence, probably with assistance from other TTPs, for example, from a notary when evidence was generated through a secure envelope. Nowadays, different online arbitrator platforms exist, which allow for dispute resolution being processed through document and evidence transactions, as well as the cooperation of online parties. The dispute resolution process can either be registered in one of these platforms and use its services, or use its own rules for the definition of an online arbitrator.

7.6 Non-Repudiation Requirements

Different targets of each non-repudiation service may influence the protocol design. Nevertheless, there are several common requirements in the design of a good non-repudiation protocol, described next.

- **Fairness**. Repudiation can only be prevented when each party is in possession of proper evidence and no party is in an advantageous position during a transaction. The reliability of communication channels affects evidence transfer. Moreover, a dishonest party may abort a transaction, which could leave another party without evidence. Various fair non-repudiation protocols with different features have been proposed. A protocol fulfills strong fairness if, when the exchange is completed, A can at least prove to an arbitrator that B has received (or can still receive) the item, without any further intervention from A. On the other hand, a protocol fulfills weak fairness if, when the exchange is completed for A, it can prove to an arbitrator that B has received (or can still receive) the item, or otherwise an affidavit can be presented to demonstrate that B misbehaved or a network failure occurred.

- **Efficiency**. TTPs will usually be involved in non-repudiation services and this involvement will be essential in order to determine the efficiency of the protocol. Fair non-repudiation protocols, proposed in literature, meet the criterion of efficiency and are often called optimistic protocols. Some authors define this property as effectiveness; that is, if no error occurs and no party misbehaves, then the TTP should not intervene.

- **Timeliness**. This is also desirable in evidence transfer. For various reasons, a transaction may be delayed or terminated. Hence, the transacting parties may not know the final status of a transaction on time, and would like to unilaterally bring a transaction to completion in a finite amount of time without losing fairness.

- **Policy**. This has to perfectly define all parameters needed by the service, some of which can be rules for evidence generation and verification, as well as for evidence storage, evidence use, and the dispute resolution process.

7.7 Non-Repudiation Security Service

7.7.1 Objective

In “WP3.7 D 3.7.2 “Master document” the following specific security requirements have been identified for this Security Service:

- ***NCP-Req#3.7.10(Non Repudiation).***

Objective of the non-repudiation security service is to provide a mechanism for the member states to be able to prevent parties from claiming not to have requested certain information. Non-repudiation is established through the use of electronic signatures in sending requests.

7.7.2 Scope

The epSOS-NCP Non Repudiation scope, coherently with the epSOS project it is restricted to the transactions defined to exchange Patient Summary as described by UC 1–2 and ePrescription - eDispense as described by UC 1.

7.7.3 Electronic Signature

An electronic signature is a set of electronic information attached to or logically associated with an electronic document and used for authentication: and provide a certainty on the identity of the signer. Within the epSOS architecture we opt to use an advanced electronic signature with a "qualified certificate" (the signature certificate). It is proposed that the advanced electronic signature will satisfy the following requirements:

- It is uniquely linked to the signatory;
- It allows identification of the signatory;
- It was created using means that the signatory can maintain under his (sole) control;

7.7.4 Electronic Signature Process

This section describes the processes involved in the use of electronic signatures within the context of epSOS. In the descriptions the actors, roles and objects from “WP 3.7 D.3.7.2 Master document” are used to create the processes.

The processes described in this section, are generic processes. These generic processes serve as a template for applying specific use cases such as ePrescription and Patient Summary. The figure below gives an overview of the procedure when using the electronic signature. The figure shows that the process consists of seven activities.

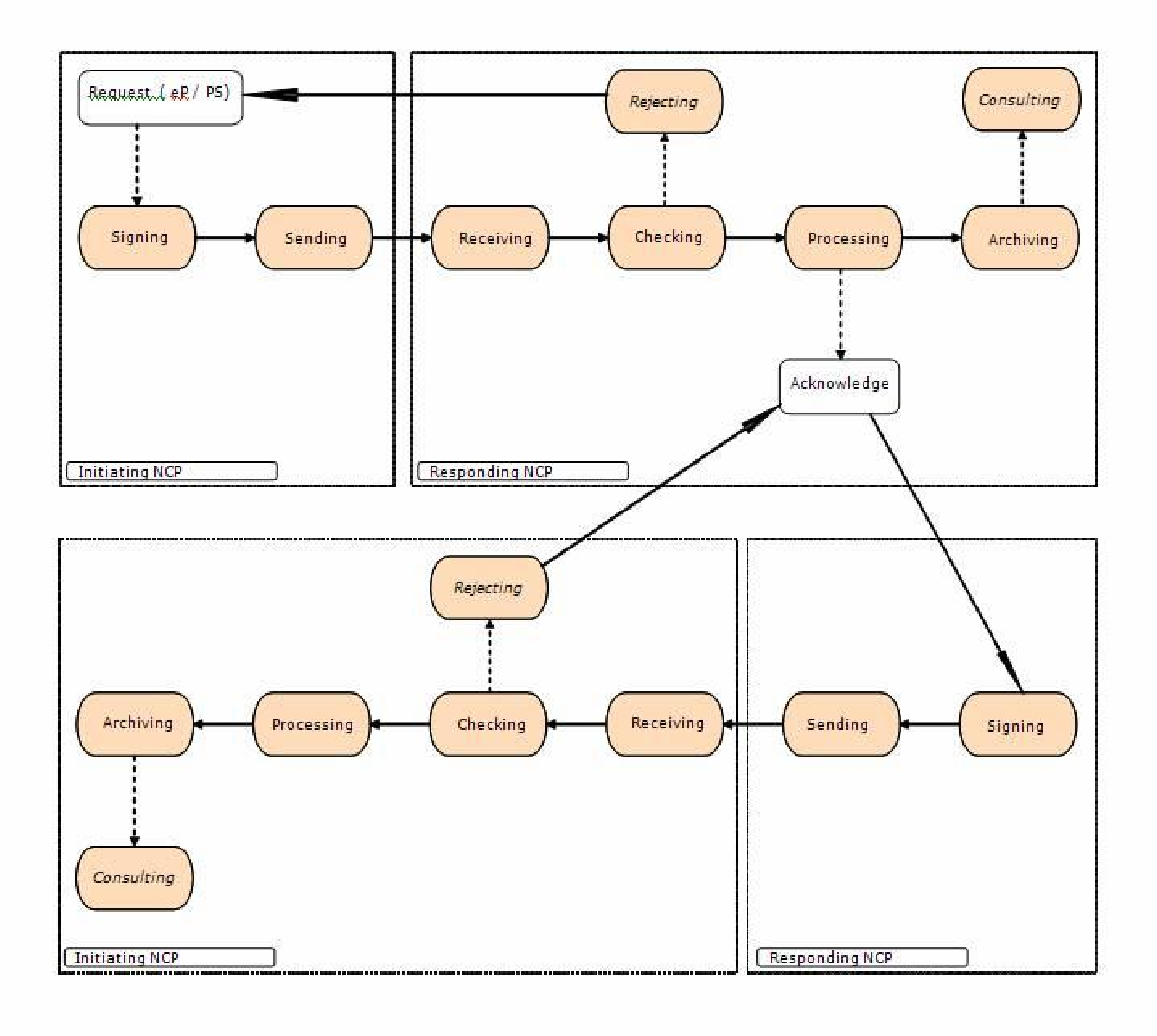


Figure 7-2: Digital Signature Process

7.7.4.1 Signing

The first step is that of the signer signing a specific patient related request, with the aim of connecting its name as responsible for the patient piece. By signing, the signer first takes note of the request it will sign. If the signer believes that such an information request is accurate, it then places its signature. The signed request and the signature cannot be modified without invalidating the signature, either by the signer or by another, if the signature is placed.

> **REQUIREMENT:** The to-be signed subject is the request of the piece of patient information. There is no direct relationship between the signature and the fact that the signature or the used patient document is used for the exchange of relevant patient information. The main reason is that a message usually contains additional information that is important for transport and delivery of the message, but is not relevant for the signature.

This requirement makes it possible to use electronic signatures in various forms of information exchange.

7.7.4.2 Sending

The second step consists of sending the signed request to a particular destination (Responding epSOS-NCP). The request is thereby packed into one message. It must remain fully intact (i.e. the signed information must be sent as a whole and the shipment does not change, the content, cohesion and order), because otherwise the risk of misinterpretation arises of which the signatory may not be aware.

It could be useful for the sender to keep a copy of the sent requests in a local archive. This archive can be useful if the sender doubts on the correctness of the requests or if a difference of opinion arises between sender and receiver on exchanged messages.

7.7.4.3 Receiving

This will normally be an automated step in which the request is unpacked i.e. stripped of transportation-related message elements and the content, with signature, will be prepared for the target processor. The signed data or the signature may not be modified upon receiving. This would lead to a valid signed request being rejected later when it is being checked. The processing of the request can be done by someone other than the recipient.

7.7.4.4 Checking

Checking the validity not only includes verification of the signature itself but also the procedural aspects like checking permission of the signatory and whether the correct data is used in the signed request.

Verification of the signature itself includes several controls: the request is signed, the signature and the request belong to each other, the signature was created by a person with the right authorization and that the content, coherence and order of the request have not changed since the signature was created. The signature must also be issued by a trusted party (trusted CSP). This CSP provides services to validate the trust mechanisms. The verification of the signature itself can be an automated step in which the system of the responding epSOS-NCP verifies the request and only offers correctly signed requests to the final processor.

> **REQUIREMENT:** Verifying the signed request should be made fully automated.

Checking whether the sender of a signed request was authorized to sign is a responsibility of the responding epSOS-NCP. In practice this means that the processor typically assumes that the signer was authorized to do so, has signed with a valid mechanism and requests information which it is authorized to request. Only in case of doubt, will the processor make an additional check.

> **REQUIREMENT:** The processor of a signed request message should itself determine if he accepts the authorization (to sign) or that verification needs to take place. The responding epSOS-NCP should not assume that authorization is done earlier in the chain.

7.7.4.5 Rejecting

If the verification step shows that the signature is invalid, then the received request message is not processed. The refusal is fed back to the sender and is reported to the intended processor.

If desired, a system manager can be informed, to create a technical analysis of any problems in compliance with security and privacy. It is up to the target processor to decide on how to act. Depending on the situation, he can choose to ignore the message, contact the sender or, if necessary under the circumstances act on the part of the information that is considered reliable. The responding epSOS-NCP acts on its own responsibility and competence. It is also possible to inform authorities in cases of suspected malpractice.

With an invalid signature it is possible that the signed request is mutilated. In special circumstances it is possible that the request has been addressed or delivered wrong. The target processor must therefore, when encountering a non-valid signature, determine whether the possibility exists that the request is not for him but for another epSOS-NCP. In that case the responding epSOS-NCP may not actually take note of the signed request.

> **REQUIREMENT:** On rejection of the received request as a result of an invalid or missing signature, the requesting epSOS-NCP should receive a notification of this event.

7.7.4.6 Processing

If the signature part is deemed valid in the monitoring step, the Acknowledge is sent back to the initiating epSOS-NCP (following the same digital signature process[6]). then the received request is processed in accordance with the agreements. Typically this implicates that the request is validated and transaction of e.g. a medical summary can take place.

7.7.4.7 Archiving

ETSI makes certain recommendations on archiving [CWA14171].
In the epSOS LSP the need for long term archiving is not foreseen and for this reason no specific requirements on archiving were provided. Every epSOS-NCP is responsible for deciding what a reliable archive for storing requests and signatures would look like. In general when electronic signatures are included in messages that are archived, additional measures are recommended for ensuring long term reliability of the archive, in such a way that later on the validity of archived signatures may be demonstrated.

> **REQUIREMENT:** The basic infrastructure does not have (central) facilities to support long-term secure archiving of messages and / or electronic signatures.

The creation of such an archive infrastructure is however recommended for each MS (or epSOS-NCP).

7.8 Option analysis

In order to guarantee non-repudiation of origin and receipt between NCPs within epSOS, a couple of options are available, each with their own merits. Each option will require some

[6] number of parameters (repetitions, time outs, etc...) have to be carefully considered during implementation

kind of time stamping mechanism or sequence numbering to avoid message replay, which is similar for each option and therefore not important for the ultimate choice and left out in this analysis. The non-repudiation of receipt requires some kind of acknowledgement message, irrefutably connected to the sent message. Since non-repudiation of receipt can be considered as non-repudiation of origin with respect to the acknowledgement message, we only focus on non-repudiation of origin. Furthermore, we assume a PKI is available for the NCPs to ensure e.g. data confidentiality. In the following we describe the different options and describe the pros and cons of each option.

1. Digital signatures.
   The available PKI could be used to generate digital signatures for each message. The sender computes a message digest by using a hash function and signs the digest with his private key. The signed digest is sent along with the message. The receiver will be able to verify the signature by using the public key of the sender.

2. Message authentication codes.
   Instead of using the PKI, symmetrical keys could be used. Each pair of NCPs should have their own symmetrical key. When encrypting each message with the key that was shared with the recipient, this ensures some kind of non-repudiation of origin. A message authentication code can be considered as a keyed hash function, which computes a message digest by using the unique shared key. The message authentication code is sent along with the message and can be verified by the receiver.

3. epSOS notary.
   Instead of using a direct communication between the NCPs, a central epSOS organisation could be used that registers all communicated messages and acts as a notary between NCPs in case of later dispute. Each message from a NCP that is intended for another NCP~~,~~ should be sent to the epSOS notary, who registers the message and forwards it to the intended receiving NCP. To ensure mutual authentication when communicating with the notary, either symmetric or asymmetric keys could be used.

A summary of the options it is presented in the following table:

Option	1	2	3
Pros	o No extra key management o Direct communication	o Light computational effort for each message o Direct communication	o No additional computations for non-repudiation
Cons	o Reasonable computational effort for each message	o Receiver is able to create fake messages	o Small amount of extra key management o A trusted notary should be permanently available o Indirect communication

Table 7-1: Non Repudiation Option Summary

Since option 2 enables the receiver to pretend a message was sent that was never actually sent, its notion of non-repudiation of origin is very weak. For option 3 it would be very difficult to find an organisation that is trusted by all NCPs. Furthermore, being a central point of failure, the availability requirement of such a notary would be very high. And finally, a delay could be introduced because of the indirect communication which would be undesirable for many practical situations. Therefore, option 1 is chosen to implement non-repudiation of origin. The implementation mechanism will rely on XML signature standard.

7.8.1 Pilot phase simplification

To reduce the cost and the complexity of the epSOS NCP technical implementation, the requirement for a signed Acknowledge of every received message could be relaxed.
The waiver of this requirement must be compensate by specific organizational and procedural security measures, and must be confirmed by a risk analysis.

8 PKI Security Service

8.1 Introduction

This paper provides guidance to the epSOS PKI Security Services description. The Public Key Infrastructure (PKI) is a set of hardware, software, people, policies, and procedures needed to create, manage, store, distribute, verify and revoke digital certificates.

In cryptography, a PKI is an arrangement that binds public keys with respective user identities by means of a *certificate authority* (CA). The user identity must be unique for each CA. The binding is established through the registration and issuance process, which, depending on the level of assurance the binding has, may be carried out by software at a CA, or under human supervision. The PKI role that assures this binding is called the *Registration Authority* (RA). For each user, the user identity, the public key, their binding, validity conditions and other attributes are made unforgettable in public key certificates issued by the CA.

Another important service associated with a PKI is Time Stamp. This service is implemented by the TSA (Timestamping Authority). It can be used for logging, auditing, accounting, etc. However, this service will not be necessary for the epSOS project due to the fact that there is no need for strict accuracy in date and time, since each epSOS-NCP in the project can simply trust in the date and time synchronization service, that is already a requirement.

Therefore, the most important things a PKI can offer to the epSOS project are the certificates on their own and the validation service for them. It is assumed that each epSOS-NCP will choose the PKI that will be used based on the requirements of this document.

8.2 PKI Security Service Objective

The objective of PKI security service is to supply certificates and validation services that will be used to ensure the confidentiality and the integrity of the services of epSOS Patient Summary and ePrescription. Having taken this into account, the PKI security service should ensure:

- A secure exchange of information among the epSOS-NCPs involved
- A way to assure that the epSOS-NCPs are correctly identified and authenticated
- The confidentiality of data
- Integrity of data, and
- Non-repudiation of processes.

These criteria have already been translated into Technical requirements in the deliverable of the WP 3.7. Those requirements are listed in the “WP3.7 D3.7.2 Master document” and are:

- **epSOS-NCP_Req#3.7.01 (epSOS-NCP identification);**
- **epSOS-NCP-Req#3.7.02 (Authenticating Network Access);**
- **epSOS-NCP-Req#3.7.03 (Digital Signatures);**
- **epSOS-NCP-Req#3.7.05 (Confidentiality);**
- **epSOS-NCP-Req#3.7.08 (System and data integrity);**
- **epSOS-NCP-Req#3.7.10 (Non Repudiation).**

Accuracy on date and time (timestamp) is not the goal of the PKI service since this is being covered in another service profile-(time synchronization profile).

8.3 PKI Security Service Scope

The scope of the PKI security service is the European epSOS domain, that is, the epSOS-NCPs and their processes and data needed for their relationships. The processes between epSOS-NCPs and their corresponding PoCs are out of scope. Those processes have to be dealt with internally by the MSs.

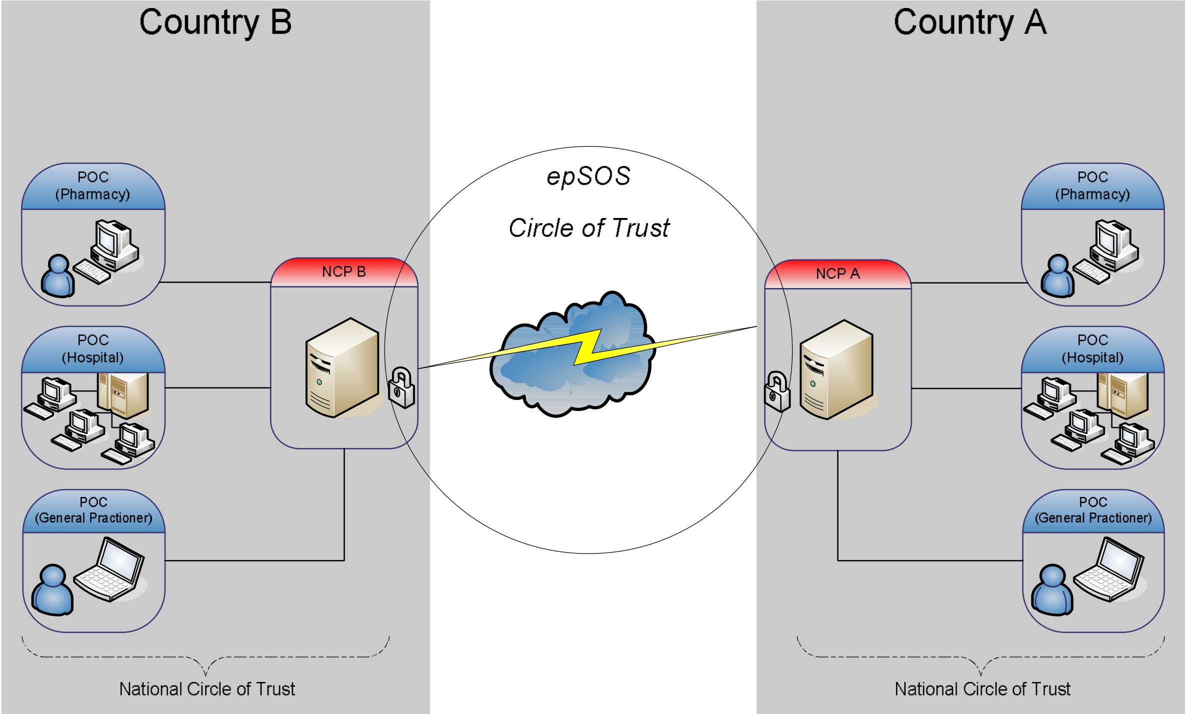


Figure 8-1: European and National epSOS domains.

8.4 Vulnerability Management

<u>Vulnerability Assessment:</u>

1. Vulnerabilities related to CA.
2. Vulnerabilities related to private key.
3. Vulnerabilities related to the verifying computer (the one that uses the certificate).
4. Inappropriate content of digital certificates.
5. Vulnerabilities related to final users.
6. Malpractices associated with the implementation of PKI.
7. Vulnerabilities associated with CRL.
8. Unavailability of PKI's validation service.
9. To keep up-to-date the epSOS-NCPs' URL with its corresponding certificates.
10. If Timestamp is used, it would be possible to time stamp with incorrect date/time.
11. Lack of compliance with EU Electronic Signatures Directive and EU Data protection.

Attack scenarios analysis:

1. Vulnerabilities related to CA.
    a. A CA might not be as trusted as desirable.
    b. Phishing the CA.
2. Vulnerabilities related to private key.
    a. There is always the risk of using a weak algorithm to generate the public key from the private key. In this case, the value of the private key could be found out. A weakness in key generation would expose the system to, e.g., brute-force attacks.
3. Vulnerabilities related to the verifying system (the one that uses the certificate).
    a. Poor security measures could make an attack by viruses and other intrusion tools easy. An attacker could add his own public key to the final user computer.
4. Inappropriate content of digital certificates.
    a. Certificates' content varies with CA system. A certificate's security has both physical and logical vulnerabilities that are outgrowths of the software used to generate a digital signature.
5. Vulnerabilities related to final users.
    a. Lack of education on final users would fall into compromising keys.
6. Malpractices associated with the implementation of PKI.
    a. Exploitation of the weaknesses of the strategy for deployment of PKI (technical measurements, policies, procedures…).
7. Vulnerabilities associated with CRL.
    a. Inaccuracies in the CRL would be a source of transaction risk for the CA system.
    b. Use of revoked certificates due to unavailability of CRL.
    c. It can be used by a false user (attacker).
8. Unavailability of PKI's validation service
    a. It wouldn't be able to be used by any actor because the connection will be refused, therefore this would be a deprecation of the service.
9. To keep up-to-date the epSOS-NCPs' URL with its corresponding certificates
    a. It wouldn't be able to be used by any actor because the connection will be refused, therefore this would be a deprecation of the service.
10. If Timestamp is used, it would be possible to time stamp with incorrect date/time.
    a. If the date-time cannot be correctly checked this can be used by some actor to repudiate its actions.

11. Lack of compliance with EU Electronic Signatures Directive and EU Data protection.
    a. Possibility of legal actions against the organization.

Acceptable risk level definition:

The risk will be low because the possible attacks are not very likely for all the vulnerabilities and there are many measures and standards to keep the PKI services under control. The vulnerabilities risk level definitions are:

1. Vulnerabilities related to CA.

   Not everyone who possesses a digital certificate is actually trustable. The selected CA must have sufficient international recognition. Certification Authority(ies) must be appropriately accredited by any of the following: the national authorities, the ETSI 101 456 Policy Requirements for Certification Authorities issuing Qualified Certificates, the “Webtrust programme for Certification Authorities”, the standard framework of AICPA/CISA, tScheme, or equivalent.

2. Vulnerabilities related to private key.

   Each issuing authority must use well-known algorithms and a large bit length for the generated keys to prevent an attacker from predicting the keys and causing problems. In this way, the software of the PKI that issues the certificates must be at least certified to Common Criteria level EAL3, although level EAL4+ would be desirable.

3. Vulnerabilities related to the verifying system (the one that uses the certificate).

   Computers, storage devices or whatever they use to keep the private keys, will be protected by a combination of security solutions to achieve a high-level protection such as strong passwords, anti-virus, firewalls, intrusion detection tools, etc.

4. Inappropriate content of digital certificates.

   Validation can be implemented via OCSP protocol RFC 2560 over HTTP/HTTPS or using CRLs (Certificate Revocation List) RFC 5280.
   Timestamping, if finally considered, can be implemented via RFC 3161 Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP).

5. Vulnerabilities related to final users.

   CA will need to provide some information concerning the basic services supported and the rights and responsibilities of subscribers and relying parties. The use of the public key cryptography on a broad scale allows the users to benefit from a security infrastructure to manage public keys.

6. Malpractices associated with the implementation of PKI.

   The CA must maintain a strong and secure architecture to avoid security breaches, and a comprehensive fail-over plan that provides a secondary infrastructure to maintain availability of services in the case of a failure of the primary infrastructure.

7. Vulnerabilities associated with CRL.

   CRLs (Certificate Revocation List) must comply with IETF RFC 5280.

8. Unavailability of PKI’s validation service.

   It will be necessary to establish a strong validation service architecture addressed to assure a reasonable level of availability.

9. To keep the epSOS-NCPs’ URL up-to-date with its corresponding certificates.

   Probably, this vulnerability is not used for an attack but it could deprecate the service. There should be synchronization between the URL modification and the certificate.

10. If Timestamp is used, it would be possible to time stamp with incorrect date/time.

    Timestamp should comply with RFC 3161 Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP).

11. Lack of compliance with EU Electronic Signatures Directive and EU Data protection.

    The PKI security service must be developed within the existing legal framework in the EU.

8.5 Requirements and Constraints

There are a number of requirements that are a must and some of them that are desirable. Having taken into account the state of the art of the current PKI security services the following requirements have been considered:

8.5.1 Mandatory requirements

1. Certificates must comply with UIT-T x.509 v3 standard.

The use of these certificates together with the mechanisms of Digital Signature~~,~~ and secure communications (for example, SSL) assure the confidentiality, integrity, and non-repudiation of the data.

2. CRLs (Certificate Revocation List) must comply with IETF RFC 5280.

3. The PKI used also must comply with:
   - PKCS#7 (Cryptographic Message Syntax Standard),
   - PKCS#10 (Certification Request Standard), and
   - PKCS#12 (Personal Information Exchange Syntax Standard) standards.

4. The software of the PKI that issues the certificates must be at least certified to Common Criteria level EAL3.

5. Certification Authority(ies) must be appropriately accredited by any of the following: the national authorities, the ETSI 101 456 Policy Requirements for Certification Authorities issuing Qualified Certificates, the “Webtrust programme for Certification Authorities”, the standard framework of AICPA/CISA, tScheme, or equivalent.
   ETSI 101 456. It is the standard European certification therefore we should recommend the certification in this one.

8.5.2 Desirable requirements

1. Cryptographic hardware should comply with FIPS 140-2 level 3 or CEN CWA 14167 1-2.

2. The software of the PKI that issues the certificates should be certified to Common Criteria level EAL4+.

3. Each epSOS-NCP should have different certificates for each function of the PKI:

    - Mutual Authentication Certificate
    - Digital Signature Certificate

If timestamp is to be considered, Timestamp should comply with RFC 3161 Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP).

8.6 **Methods**

The PKI provides certificates and a service to validate them. In this profile the validation service is the only one being used and described.

8.6.1 Measures

- **Measures identification:**

Validation can be implemented via OCSP protocol RFC 2560 over HTTP/HTTPS or using CRLs (Certificate Revocation List) RFC 5280.
Timestamping, if finally considered, can be implemented via RFC 3161 Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP).

- **Measures option definition & variant analysis**

The most common measure to validate that certificates are not revoked is CRLs, however this method has some deficiencies that have been solved with OCSP. Particularly, the mechanism is safer since OCSP checks the validity of the certificates on-line. However, CRLs may suffer a delay in its update and therefore, revoked certificates might be validated.

8.6.2 Implementation

The PKI service could be implemented in four different ways.

1. Using a different PKI for each epSOS-NCP with a repository of root certificates in each epSOS-NCP.
   - Pros:
     - To facilitate the deployment of the Project
     - Probably, the PKIs in use have already been proven so they should be reliable.
     - Using already implemented PKI services would allow the maintenance of the architecture of the epSOS-NCPs and the applications already in use.
   - Cons:
     - Difficult updates.
     - Multiple resources to check for the validity of the certificates.
2. Using a~~n~~ unique PKI for the epSOS Project with a proprietary one (like Verisign).
   - Pros:
     - Easy to use.
     - Easy to validate.
     - Common configuration of the PKI services among the epSOS-NCPs.
   - Cons:
     - External party (no under epSOS control).
     - Probably, this will imply modifications in the epSOS-NCPs (against the premises of ANNEX I).
3. Using a different PKI for each epSOS-NCP with a common repository of root certificates.
   - Pros:
     - It would allow the use of pre-existent certificates.
     - Easy updates.
   - Cons:
     - The maintenance of the root certificates should be assured by one epSOS-NCP or European node.
     - epSOS-NCPs would lose independence.
4. Implementation of dedicated epSOS-PKI.
   - Pros:
     - Common configuration of the PKI services among the epSOS-NCPs.
     - Easy to update
   - Cons:
     - Expensive
     - One epSOS-NCP or European Node must take it over.
     - This will imply modifications in the epSOS-NCPs (against the premises of ANNEX I).

The different options have been listed in order of feasibility having taken into account the pros and cons of each option. The specification for the most recommended option, option 1, will be provided in the WP3.4, PKI profile.

Based on the objectives of the PKI service profile the validation service (which must be defined in WP3.4) is proposed to be used in the following way having taken into account that the protocols should be finally described in the deliverable of WP3.4.

A summary of this is the following table:

Option	#1	#2	#3	#4
Description	Using a different PKI for each NCP with a repository of root certificates in each NCP.	Using a unique PKI for the epSOS Project with a proprietary one (like Verisign).	Using a different PKI for each NCP with a common repository of root certificates.	Implementation of dedicated epsos-PKI.
Pros	-To facilitate the deployment of the Project - Probably, the PKIs in use have already been proven so they should be reliable. - Using already implemented PKI services would allow maintaining the architecture of the NCPs and the applications already in use.	- Easy to use. - Easy to validate. - Common configuration of the PKI services among the NCPs.	- It would allow the use of pre-existent certificates. - Easy updates.	- Common configuration of the PKI services among the NCPs. - Easy to update
Cons	- Difficult updates. - Multiple resources to check for the validity of the certificates.	- External party (not under epSOS control). - Probably, this will imply modifications in the NCPs (against the premises of ANNEX I).	- The maintenance of the root certificates should be assured by one NCP or European node. - NCPs would lose independence.	- Expensive - One NCP or European Node must take it over. - This will imply modifications in the NCPs (against the premises of ANNEX I).
Complexity/costs	Complexity: Low in implementation, high in update Cost: No new costs	Complexity: Low Cost: 1500 € per year and certificate, approach	Complexity: Medium Cost: New infrastructure is needed	Complexity: Medium Cost: Expensive
Diffusion in MS	High[7] implementation of PKI services			None
Constraints				
Adoption in pilot/target phase	Yes			

Table 8-1: PKI Implementation Option Summary.

[7] Based on the answers given gathered in WP 3.7 questionnaire [ref WP3.7 D3.7.2 Master doc

- **Identification, authentication, and confidentiality objective**

The certificates used must be domain web server certificates, obviously. The epSOS-NCPs must be identified by their own certificate and must validate among themselves. Usually, the mutual authentication among the epSOS-NCPs can be offered by the protocol IPSec at the network layer, by TSL at the transport layer and by SSL v3 at the transport to application layer. Besides, these protocols also ensure the confidentiality of the data transfer. These protocols create an encrypted tunnel for secure exchange of data.

epSOS-NCPs would work as client or as server according to the address of the message to be transferred and the authentication certificate should be used.

However, due to the secure point-to-point nature of the socket connection, it doesn’t work for multi-hop connections, e.g. in the presence of firewalls, proxies, etc. Then the easier, but not the only way to implement a secure connection between NCP-A and NCP-B is by using SOAP WS – SecureConversation (WS-SC), a protocol on top of SOA WS – Security protocol. WS-SC defines how secure contexts are established and how derived keys are computed and passed.

- **Integrity and non-repudiation objective**

Regarding the integrity and non-repudiation each epSOS-NCP-sender must digitally sign every message before the exchange of information using its own private key associated with its certificate, which should be specific for this purpose as mentioned above (digital signature certificate). After that, the epSOS-NCP-receiver will receive the digitally signed message and will use the PKI validation service to check the certificate to see if it could be accepted or not. Then, the epSOS-NCP-receiver will check the integrity of the message to either accept it or not. That means, the addressee verifies the signature; it computes the hash value of the received message, and using the public key from the certificate, it decrypts the encrypted hash value (digital/electronic signature of the message), and compares both hashes. If they are equal, that means it has been correctly signed, the epSOS-NCP-sender cannot repudiate the sending of the message.

To send the message the recommended protocol according to the agreements in epSOS about the use of SOA standards is the WS-Security: SOAP Security Message. This protocol provides support for multiple security token formats, multiple trust domains, multiple signature formats, and multiple encryption technologies. This specification provides three main mechanisms: the ability to send security tokens as part of a message, message integrity, and message confidentiality.

9 Bibliography

9.1 Access Control

1. Abadi, M., Burrows, M., Lampson, B., Plotkin, G.: A Calculus for Access Control in Distributed Systems. Transactions on Programming Language and Systems 15(4), 706-734 (September 1993)
2. Fournet, C., Gordon, A., Maffeis, S.: A Type Discipline for Authorization in Distributed Systems. In : 20th Computer Security Foundation Symposium (2007)
3. IHE IT-Infrastructure: White Paper: Access Control (Version for Public Comment).
4. NIST Available at: **ttp://csrc.nist.gov/news_events/privilege-management-workshop/PvM-Model-Survey-Aug26-2009.pdf**
5. O'Neill, M.: Web Services Security. McGraw-Hill, Berkeley, US (2003)
6. OASIS: The XACML standard document. In: OASIS Home. (Accessed 2004) Available at: **http://www.oasis-open.org/specs/index.php#xacml**
7. OASIS: SAML 2.0 Profile of XACML v2.0. In: OASIS OPEN. (Accessed February 1, 2005) Available at: **http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-saml-profile-spec-os.pdf**
8. Abadi, M., Andrew, G.: A Calculus for Cryptographic Protocols: The Spi Calculus., 36-47 (1997)
9. Dolev, D., Yao, A.: On the security of public key protocols. Technical CS-TR-81-854, Stanford University, Stanford, CA (1981)
10. Broadfoot, P., Lowe, G.: On Distributed Security Transactions that use Secure Transport Protocols. Technical Report, Oxford University Computing Laboratory, Oxford (2003)
11. Kleiner, E., Roscoe, A. W.: Web Service Security: a preliminary study using Casper and FDR. Technical, Oxford University Computing Laboratory, Oxford (2004)

12 IHE ITI Technical Committee: IHE IT Infrastructure Technical Framework.

10 Appendix

10.1 Access Control Examples (Non Normative)

In this section a set of example messages and policies are proposed. However they are just for clarification of the overall approach of the epSOS Access Control Security Service, and not to be used for other purposes.
The following example is a request from Dr. Marley sitting in Vienna AKH in Austria who wants to have read access to the Patient Summary for John Doe, an Italian patient. This SAML assertion MUST be inserted in the Header of the SOAP message according to WS-Security SAML token profile and the epSOS HCP identity assertion specification. Permissions are obtained by the HL7 RBAC catalogue and XSPA profile for XACML.

```
<soap12:Envelope … >
 <soap12:Header … >
  <ws:Security … >
<saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
 ID="urn:uuid:7102AC72154DCFD1F51253534608780"
 IssueInstant="2009-09-21T12:03:28.788Z" Version="2.0">
 <saml:Issuer
   Format="urn:todo:format" NameQualifier="urn:TODO:nq"
    SPNameQualifier="urn:TODO:SP">
    urn:idp:countryB:Austria
  </saml:Issuer>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod
       Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
      <ds:Reference URI="#urn:uuid:7102AC72154DCFD1F51253534608780">
       <ds:Transforms>
         <ds:Transform
           Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
         <ds:Transform
          Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
           <ec:InclusiveNamespaces
            xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#"
            PrefixList="ds saml xs" />
         </ds:Transform>
       </ds:Transforms>
       <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
       <ds:DigestValue>A1LyLvFHRrYaOJ28YVFd3MfKGSI=</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>
    cH+lCY …
    </ds:SignatureValue>
    <ds:KeyInfo>
      <ds:X509Data>
        <ds:X509Certificate>MIIADS …   </ds:X509Certificate>
      </ds:X509Data>
    </ds:KeyInfo>
  </ds:Signature>
```

```
<saml:Subject>
  <saml:NameID
   Format="urn:oasis:names:tc:SAML:1.1:nameidformat:emailAddress">
   bobmarley@viennaakh.at
  </saml:NameID>
  <saml:SubjectConfirmation
    Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
  <saml:SubjectConfirmationData>
  <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:X509Data>
      <ds:X509Certificate>MIIDczH</ds:X509Certificate>
      <ds:X509SubjectName>EMAILADDRESS=bobmarley@viennaakh.at,
         CN=bobmarley@viennaakh.at,
         OU=IT Services, O=Vienna AKH IT Services,
         L=Vienna,
         ST=Austria, C=AT
      </ds:X509SubjectName>
      <ds:X509IssuerSerial>
        <ds:X509IssuerName>EMAILADDRESS=idps@viennaakh.at,
         CN=idps.viennaakh.at, OU=Massimiliano Masi - IT Services,
         O=Vienna AKH IT Services,
         L=Vienna, ST=Austria, C=AT
        </ds:X509IssuerName>
        <ds:X509SerialNumber>
          12840552752097057716
        </ds:X509SerialNumber>
      </ds:X509IssuerSerial>
        </ds:X509Data>
      </ds:KeyInfo>
    </saml:SubjectConfirmationData>
  </saml:SubjectConfirmation>
</saml:Subject>
<saml:Conditions
 NotBefore="2009-09-21T12:03:28.788Z"
 NotOnOrAfter="2009-09-21T22:03:28.788Z">
  <saml:AudienceRestriction>
    <saml:Audience>http://epSOS-NCP-country-a/italy</saml:Audience>
  </saml:AudienceRestriction>
</saml:Conditions>
<saml:AuthnStatement
 AuthnInstant="2009-09-21T12:03:28.788Z"
 SessionNotOnOrAfter="2009-09-21T22:03:28.788Z">
 <saml:AuthnContext>
   <saml:AuthnContextClassRef>
    “urn:eu:epsos:trustlevel:4”.
     urn:oasis:names:tc:SAML:2.0:ac:classes:kerberos
   </saml:AuthnContextClassRef>
 </saml:AuthnContext>
</saml:AuthnStatement>
<saml:AttributeStatement>
  <saml:Attribute
    FriendlyName="XSPA subject"
    Name="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
    <saml:AttributeValue
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="xs:string">Dr. Marley
   </saml:AttributeValue>
  </saml:Attribute>
  <saml:Attribute
   FriendlyName="XSPA organization"
   Name="urn:oasis:names:tc:xspa:1.0:subject:organization"
   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
   <saml:AttributeValue
     xmlns:xs="http://www.w3.org/2001/XMLSchema"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:type="xs:string">Vienna AKH
   </saml:AttributeValue>
  </saml:Attribute>
  <saml:Attribute
   FriendlyName="XSPA organization id"
   Name="urn:oasis:names:tc:xspa:1.0:subject:organization-id"
   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
   <saml:AttributeValue
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="xs:anyURI">urn:oid:1.2.3.4.5.6.7
   </saml:AttributeValue>
  </saml:Attribute>
  <saml:Attribute
   FriendlyName="XSPA permissions according with Hl7"
   Name="urn:oasis:names:tc:xspa:1.0:subject:hl7:permission"
   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
   <saml:AttributeValue
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="xs:string">urn:oasis:names:tc:xspa:1.0:hl7:PRD-006[8]
   </saml:AttributeValue>
   <saml:AttributeValue
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="xs:string">urn:oasis:names:tc:xspa:1.0:hl7:PRD-003
   </saml:AttributeValue>
   <saml:AttributeValue
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="xs:string">urn:oasis:names:tc:xspa:1.0:hl7:PRD-012
   </saml:AttributeValue>
   <saml:AttributeValue
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="xs:string">urn:oasis:names:tc:xspa:1.0:hl7:PRD-010
   </saml:AttributeValue>
  </saml:Attribute>
  <saml:Attribute
   FriendlyName="XSPA role"
   Name="urn:oasis:names:tc:xspa:1.0:subject:role"
```

[8] Patient identification and Lookup, according to Hl7 Role Based Access Control (RBAC) Healthcare Permission Catalog.

```
   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
   <saml:AttributeValue
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="xs:string">Physician
   </saml:AttributeValue>
  </saml:Attribute>
  <saml:Attribute
   FriendlyName="XSPA Purpose Of Use"
   Name="urn:oasis:names:tc:xspa:1.0:subject:purposeofuse"
   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
   <saml:AttributeValue
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="xs:string">TREATMENT
   </saml:AttributeValue>
  </saml:Attribute>
  <saml:Attribute
   FriendlyName="Medical Record"
   Name="urn:oasis:names:tc:xspa:1.0:environment:locality"
   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
   <saml:AttributeValue
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="xs:string">vienna-akh
   </saml:AttributeValue>
  </saml:Attribute>
 </saml:AttributeStatement>
</saml:Assertion>
```