Security Target

for

SecDocs Security Komponenten Version 1.0

Version	Date	Author	Comments
0.1	19.10.2010	Stefan Dörpinghaus	Initial version based on PP v.1.0.0 (Certification ID BSI-CC-PP-0049)
0.2	05.11.2010	Stefan Dörpinghaus	Integration of Crypto Provider and OverSign components into TOE
0.3	17.11.2010	Stefan Dörpinghaus	Initial draft version
0.4	25.11.2010	Stefan Dörpinghaus	2nd draft version
0.4.1	29.11.2010	Stefan Dörpinghaus	Inclusion of comments from several colleagues
0.5	03.12.2010	Stefan Dörpinghaus	Overall Revision
0.5.1	07.12.2010	Stefan Dörpinghaus	Smaller Rework
0.6	10.12.2010	Stefan Dörpinghaus	Enhancements according to internal discussions with A. Lunkeit and A. Menke
0.6.1	15.12.2010	Stefan Dörpinghaus	Minor corrections
0.6.2	05.01.2011	Stefan Dörpinghaus	Changing acronym for security functions to SF; TOE supports only 64-bit OS
0.7	11.02.2011	Stefan Dörpinghaus	Revised version separating middleware and SCA component
0.7.1	16.02.2011	Stefan Dörpinghaus	Minor corrections concerning TOE summary specification
0.8	28.02.2011	Stefan Dörpinghaus	Revised version based on observations made by the certification facility
0.9	16.03.2011	Stefan Dörpinghaus	Further corrections based on observations made by the certification facility; update of cited literature
0.9.1	06.04.2011	Stefan Dörpinghaus	Further corrections based on observations made by the certification facility
0.9.5	11.05.2011	Stefan Dörpinghaus	Final version: Last adoptions mostly in TSS; changed audit level in FAU_GEN.1
0.9.6	22.06.2011	Stefan Dörpinghaus	Final version: Adoption to diminished evaluation assurance level 3; updated version of literature reference #13 (Algorithm catalogue)
0.9.7	06.07.2011	Stefan Dörpinghaus	Final version: Change of the description of the EAL 3 goal in chapter 5.5
0.9.8	08.07.2011	Stefan Dörpinghaus	Final version: Changes according to comments from the certifier
0.9.9	21.07.2011	Stefan Dörpinghaus	Final version: Description of obsolete integrity tool omitted
1.0.0	29.07.2011	Stefan Dörpinghaus	Final version: More detailed specification of the supported OS; deletion of incorrect term at end of 4th paragraph in chap. 6.1; changed TOE description

1.0.1	15.09.2011	Stefan Dörpinghaus	Final version: Corrected specification of the supported OS (see p. 13)
1.0.2	23.09.2011	Stefan Dörpinghaus	Corrected typo concerning necessary TOE disk space
1.0.3	05.10.2011	Stefan Dörpinghaus	Introduction of a footnote concerning the planned TR-03125 certification process
1.1	31.10.2011	Stefan Dörpinghaus	Final version: Adoption to increased evaluation assurance level 4
1.2	05.06.2012	Stefan Dörpinghaus	Final version: Inclusion of topics resulting from a discussion with the certification facility
1.3	06.06.2012	Stefan Dörpinghaus	Final version: Inclusion of further comments from the evaluation facility
1.4	15.06.2012	Stefan Dörpinghaus	Final version: Inclusion of last comments from the evaluation facility
1.5	16.06.2012	Stefan Dörpinghaus	Final version: Inclusion of new hash value of user documentation archive
1.6	04.07.2012	Stefan Dörpinghaus	Final version: Changes due to further comments from the certification facility
1.7	10.07.2012	Stefan Dörpinghaus	Final version: Inclusion of new hash value of user documentation archive
1.8	10.07.2012	Stefan Dörpinghaus	Final version: Inclusion of new hash value of user documentation archive

Contents

List of Figures

List of Tables

1 ST Introduction

This document represents a Security Target (ST) for the software *SecDocs Security Komponenten Version 1.0* enabling the legally compliant long-term preservation of electronic documents by implementing the ArchiSafe concept developed by the Physikalisch-Technische Bundesanstalt (PTB) - the German National Metrology Institute providing scientific and technical services.

1.1 ST Reference

ST Name:	Security Target for *SecDocs Security Komponenten Version 1.0*
TOE:	SecDocs Security Komponenten Version 1.0
Certification ID:	BSI-DSZ-CC-0685
ST Version:	1.8
Date:	10.07.2012
Sponsor:	OpenLimit SignCubes AG
Editors:	OpenLimit SignCubes AG
CC Version:	3.1 (Revision 3)

This document contains the Security Target of the software *SecDocs Security Komponenten Version 1.0* which is from now on called TOE ("*Target of Evaluation*").

This Security Target is compliant to the Common Criteria Protection Profile for an "ArchiSafe Compliant Middleware for Enabling the Legally compliant Long-Term Preservation of Electronic Documents" (BSI-CC-PP-0049) [5].

1.2 TOE Reference

The TOE described in this ST is the software named "SecDocs Security Komponenten Version 1.0" manufactured by the OpenLimit SignCubes AG. The build version of the TOE is 1.0.308_6236. The

TOE is part of the software product “SecDocs Version 1.0”, but can also be utilized in other software products. The TOE’s components and services are described in the following chapter. The software product “SecDocs Version 1.0” comprises beside the TOE a Client Software Application (CS), a Crypto Provider component and Storage Plugins offering the possibility to attach Long-Term Storage Systems to the product. Herein the CS allows the web-based connection of customers delivering data to the CS, and the Crypto Provider component offers all the necessary cryptographic services including the access the external timestamp providers.

1.3 TOE Overview

Legally compliant electronic business based on electronic documents is not possible without serious precautions to ensure the authenticity and integrity of the digitally information, at least for the time schedule of legally specified and regulated retention times. The ArchiSafe approach (cf. http://www.archisafe.de) to legally compliant long-term preservation of electronic documents claims:

- To use only permanent and standardized document formats for the contents data only, which guarantees the long-term readability of the stored information,
- To encapsulate the contents data together with all the business information, required for a complete reconstruction of the business operation in the future, in a self-contained archive object, based on a valid and authorized XML schema,
- To protect the integrity and authenticity of the actual content (“primary information”) by strong cryptographic operations, like digital signatures and digital time-stamps,
- To sustain the non-repudiation of digitally signed and archived information objects by due and evidential renewal of electronic signatures[1],
- To reduce the dependencies from obsolescent IT infrastructure and storage technology by a straight service-oriented, multi-tier and client capable architecture.

1 Signature renewal is accomplished through timestamp renewal or hash-tree renewal according to [7].

The TOE specified in this ST enforces an access control to the archive and the archived objects and ensures that only authorized applications have read and write access to the archive. The archived objects can only be deleted by those applications which have generated and submitted these particular archive objects. The TOE also enforces the provisioning of a justification, if an archive object shall be deleted before its retention time.

1.3.1 Usage and major security features of the TOE

The TOE is a software product providing amongst others the core of an ArchiSafe compliant archive middleware which acts as secure archive gateway. The TOE mainly decouples the data flow (i.e. the flow of archive objects) between third party applications, such as document management systems, and the long-term storage solutions. The architecture of the complete system is shown in Figure 1.

Any archive request from a **client software application (CS)**, e.g. a document management system or any other host-like entity, to the **long-term storage unit (SU)** must be routed through the TOE.

The CS packages the information to be archived into a valid and self-contained XML document and submits the **submission data object (SDO)**[2], represented by the XML document, to the long-term storage unit via the TOE. The TOE identifies and authenticates the requesting CS and checks the integrity and validity of the submitted XML document. Furthermore, the TOE is able to check the submission data objects for compliance to rules defined by the administrator. This may include checks about existence, quality and validity of the digital signature of the submission data object. For cryptographic operations the TOE uses an external Crypto Provider as shown in Figure 1.

2 The denomination follows the OAIS framework for sharing archival notions (cf. http://www.personal.leeds.ac.uk/~ecldh/cedars/ieee00.html). OAIS distinguishes between:

- what is preserved, an *Archival Information Package* (OAIS AIP),
- what is submitted to the archive, a *Submission Information Package* (OAIS SIP), and
- what is delivered to the archive clients, a *Dissemination Information Package* (OAIS DIP).

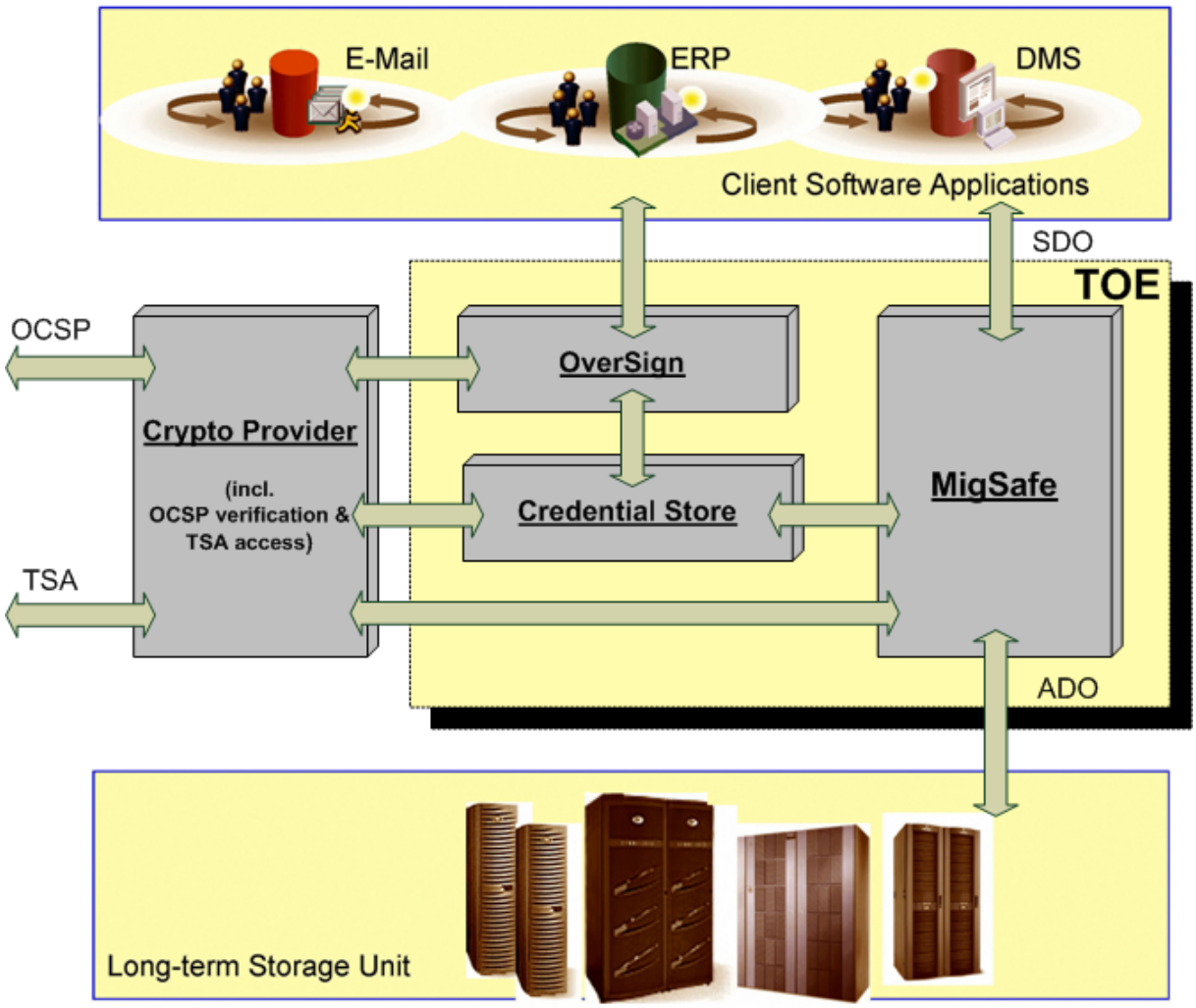

Figure 1: Architectural Overview

The real long-term storage unit in the back-end receives the submitted submission data object from the TOE for saving. The archived data object is now called **archive data object (ADO)**. The SU quits the successful storage of the ADO by sending back a unique **archive object identifier (AOID)** to the requesting CS via the TOE. This AOID will be generated outside the TOE, e.g. by the long-term storage unit or by a non-TOE part of the middleware and is required for searching and retrieving the archive object in the future by the CS.

Based on the functionality to decouple the data flow between the CS and the SU, the TOE provides the following general security functionalities:

(SF 1) preventing the access to the archive from unknown CS by reliable identification and authentication of these external entities,

(SF 2) preventing the storage of invalid submission data objects by reliable verification of the SDO before forwarding them to the SU or another trusted application which in turn forwards the SDO to the SU,

(SF 3) forwarding of successfully checked SDOs to the dedicated SU only or another trusted application which in turn forwards the SDO to the dedicated SU only,

(SF 4) preventing the erasure of ADOs by any other CS than the CS which has also submitted this ADO and preventing the erasure of ADOs before expiry of their retention time without a justification.

The TOE itself does not provide any mechanisms for long-term preservation of the non-repudiation of digitally signed archive objects by due and evidential creation or renewal of electronic signatures. The TOE does not protect the confidentiality of the documents.

1.3.2 TOE Type

The TOE is a software library being part of a TR-03125 compliant[3] IT middleware component that trustworthy and reliable mediates and controls the access to a SU for submission of SDOs or retrieval of ADOs.

The TOE consists of a set of jar archives each representing one of the following three TOE components (*MigSafe*, *OverSign*, *Credential Store*). The TOE component *MigSafe* constitutes the base of a secure archive gateway controlling the access of business applications to the long-term storage unit. The TOE component *OverSign* offers methods for the generation and renewal of evidence

[3] The TOE manufacturer is planning a TR-03125 compliance certification.

records proving the unmodified existence of archive data objects at a certain time. The TOE component *Credential Store* is used for the management and storage of user accounts and their profiles during runtime of the TOE.

For being used the TOE has to been integrated in a software component by the TOE integrator. In other terms the TOE can only been run in its integrated form. The software product “SecDocs v.1.0” offers such an integrated form of the TOE.

1.3.3 Required non-TOE hardware / software

Operational Environment

The TOE is intended to be integrated in an IT product running on servers supported by the TOE. The machine running this server must at least have a 2 GHz processor, 4 GB RAM, and 100 MB hard disk space.

The TOE is intended to be run at least in a protected environment specified in [22] as “geschützter Einsatzbereich”.

The IT environment of the TOE is protected by virus and malware protection components and the platform is protected against network based attacks. It protects the TOE and the resources used by the TOE against unauthorized modifications by suitable access protection mechanisms.

For the communication between the TOE and the *Crypto Provider Component* “OpenLimit Middleware Version 3 Server” a socket-based communication according to [25] is used.

User and administrators of the product are trustworthy and follow the instructions of the user guidance delivered with the TOE.

Operating System

The TOE runs as part of an application on an IT system and needs the protection by the underlying system platform, e.g. the operating system plus a Java Virtual Machine.

For being used the TOE needs to be integrated in an ArchiSafe compliant archive middleware. Therefore the TOE needs further parts of the ArchiSafe architecture being supplied by the TOE integrator: the TOE needs an implementation of a *Client Software Application* (CS), of the *Crypto Provider Component* and of a so-called *Storage Plugin* as a trustworthy application interfacing with the long-term storage system (SU). The TOE is designed for the usage of the *Crypto Provider Component* "OpenLimit Middleware Version 3 Server" available from OpenLimit SignCubes AG.

The CS, the *Crypto Provider Component*, the *Storage Plugin* and the SU (or another trustworthy applications interfacing with the SU) - as further parts of the ArchiSafe architecture - are not part of the TOE although the TOE depends on some features of these parties, e.g. the generation of the unique AOID by the SU (or another non-TOE part of the archive middleware).

For the integration of the TOE in an ArchiSafe compliant middleware the Java SDK 1.6.0_24 in its 64 bit version is needed.

For the execution of the TOE being integrated in an ArchiSafe compliant archive middleware acting as a secure archive gateway, the so-called "integrated form" of the TOE, the Java Virtual Machine 1.6.0_24 in its 64 bit version is needed.

The following operating systems (in their 64-bit version) are supported by the TOE:

- Red Hat Enterprise Linux Server (RHEL) 5.6 and
- Red Hat Enterprise Linux Server (RHEL) 6.0

1.3.4 Scope of the TOE

The TOE comprises the following parts:

TOE part	Name of the TOE part	SHA-256 hash value
TOE library *MigSafe*	MigSafeLibrary.jar	29d3d2489039159090324 31f58b7f98e4af9c73e674fa7be2ec3067879725c7c [4]
TOE library *OverSign*	OverSignLibrary.jar	
TOE library *CredentialStore*	CredentialStore.jar	
TOE documentation archive	MigSafeOverSign-V1.0_Documentation.tgz	55cb8691150ae77b3171ab7d7c7149211da29678a4cdc9613b19d202fe871d68

Table 1: Physical parts of the TOE

The listed TOE libraries are specific for each TOE integrator. For checking the TOE integrity, at first the integrity of the TOE documentation archive has to be verified as following: The calculated SHA-256 hash value of the file "MigSafeOverSign-V1.0_Documentation.tgz" has to be equal to the value listed in table 1.

As second step, the so-called *static* SHA-256 hash value of the TOE's libraries (aka. JAR archives) has to be verified according to the procedure described in the user documentation ([26] resp. [27]) as part of the verified TOE documentation archive: The calculated SHA-256 hash value of the TOE's libraries has to be equal to the value listed in table 1.

The logical scope of the TOE is defined through the following services the TOE provides:

- The TOE accepts archive requests from authenticated Client Software Applications (CS). Thus a successfully authenticated CS is allowed to
  - submit a submission data object to the storage,
  - retrieve an archive object from the storage,
  - delete an archive object within the storage,
  - request for evidence of a particular archive object and,
  - read some meta-information.

4 This is the *static* hash value over all existing and forthcoming integrator specific TOE libraries.

- The TOE provides an interface for so-called “Storage Plugins” as a trusted application which in turn submits the data objects to the dedicated SU. Implementations of this interface supplied by the TOE integrator reflect the characteristics of the underlying long-term storage system (SU).

Chapter 1.3 “TOE Overview” and especially chapter 1.3.1 “Usage and major security features of the TOE” offer a description of the TOE’s logical security features.

2 Conformance Claims

2.1 CC Conformance Claim

This Security Target is based upon the following:

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; Version 3.1, Revision 3, CCMB-2009-07-001 [1],
- Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components; Version 3.1, Revision 3, CCMB-2009-07-002 [2], and
- Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components; Version 3.1, Revision 3, CCMB-2009-07-003 [3].

This Security Target claims the following CC conformance:

- Part 2 conformant
- Part 3 conformant
- Evaluation Assurance Level (EAL) 4 + augmented with ALC_FLR.1

2.2 PP Claim / Conformance Statement

This security target claims strict conformance to the Protection Profile “ArchiSafe Compliant Middleware for Enabling the Legally compliant Long-Term Preservation of Electronic Documents” (BSI-CC-PP-0049) [5].

2.3 Package Claim

This security target is conforming to assurance package EAL4 augmented with ALC_FLR.1 defined in CC part 3 [3].

3 Security Problem Definition

The security problem definition describes assumptions about the operational environment in which the TOE is intended to be used and represents the conditions for the secure operation of the TOE.

3.1 Definitions

3.1.1 Subjects

Organization using the TOE

The agency or company who operates the TOE. It may be possible that the clients and their applications and/or the storage system(s) are owned by another agency or company.

Administrator

The **Administrator** installs the TOE and is in charge of the correct configuration of the TOE. In particular the Administrator is responsible for the correct implementation of the XML schemas announced and authorized by the organization using the TOE.

Integrator

The **Integrator** is in charge of the correct integration of the TOE as a part of an IT middleware component that trustworthy and reliable mediates and controls the access to a SU for submission of SDOs or retrieval of ADOs.

Client

An agency or company who operates at least one CS.

Client Software Application (CS)

An external IT entity which is capable and authorized to use the TOE for submitting archive requests to the SU.

Submitter

An external IT entity which submits a submission data object. A submitter shall be a CS.

Owner

The owner of an archive object is the CS which has submitted this particular archive object for archiving.

3.1.2 Objects

Primary Information

The contents data ("primary information") are recommended to be archived as a standard format like ASCII, PDF/A [8] or TIFF [9][5], which has to be converted into a native text format (MIME Base64 coded according to [10], section 6.8) for embedding it in the XML based data object.

Meta Information

Textual data embedded in the metadata tag of the XML based data object serving for the identification and reconstruction of the business context of the primary information.

Archive Request

An XML based data structure transferred from the CS to the TOE representing a request (operation) from this CS to the TOE. Valid requests are

- submit a submission data object to the storage,
- retrieve an archive object from the storage,
- delete an archive object within the storage,
- request for evidence of a particular archive object and,
- read some meta-information.

5 Note that the TOE will not restrict the format of the documents to be submitted to the mentioned formats nor their usage.

Submission Data Objects (SDOs)

All primary information and metadata required for an evidentiary reconstruction of business transactions in the future stored in the specified format.

A valid **submission data object (SDO)** is a self-contained XML data package, structured according to a valid and authorized XML schema. Besides the version information and the statement of the assigned XML schema, such a submission data object comprises in the simplest case two self-describing data blocks which include the contents data ("primary information") and the accompanying business context. Optionally, one or several signatures and/or time-stamp blocks are included also (see Figure 2).

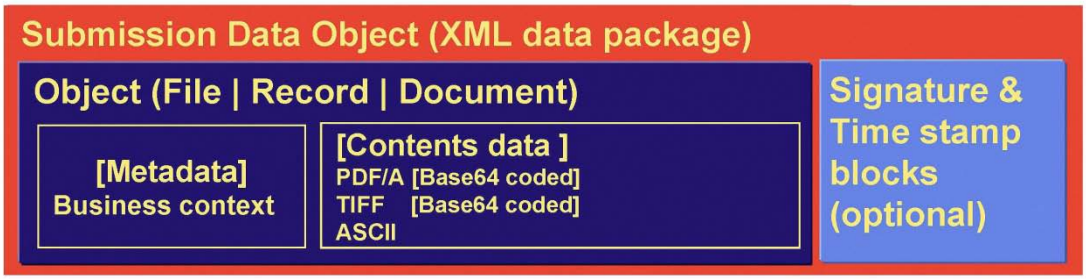

Figure 2: Structure of a Submission Data Object

The contents data block as part of the XML structure contains one or more electronic documents or the primary information in plain text, either directly or referenced by a unique resource identifier (URI). The accompanying XML metadata, like an OID, an XML based description of the document's business context or the document's retention time, is contained in the metadata block of the XML structure.

Archive Data Objects (ADOs)

Once a submission data object was successfully checked by the TOE, it will be augmented with a reference to the submitting CS (stored in the metadata block) by the TOE and stored within the archive. Now, it is called **archive data object (ADO)**. Archive Data Objects must not be modified by any party.

TOE configuration data

TOE internal data required for the correct execution of the security functionalities, especially for the correct and reliable CS identification and authentication and the verification and processing of the archive request by the TOE.

The configuration data can be specific for a CS and contain at least a set of XML schemas and a set of rules for the verification and processing of submission data objects.

XML Schema

The XML schemas define the syntax and semantic of SDOs. Authorized by the organization using the TOE, the XML schemas are the basis for the correct evaluation and processing of submitted SDOs.

Rules

The rules specify operations the TOE must perform on submission data objects, archive data objects and archive requests. Rules must be specified by the organisation using the TOE.

The rules may specify that the TOE must digital timestamp any submission data object. For this purpose, the TOE uses the external Crypto Provider.

The rules may specify that the TOE has to start the generation of an evidence record for any or a particular request for retrieval of archive Data Objects. For this purpose, the TOE interfaces to the external Crypto Provider.

Protocol Data

Log information which will be produced by the TOE.

Evidence Data

Evidence data serve for proving the unmodified existence of archive data objects at a certain time. In accordance with the specification of the IETF [7], an evidence record includes archive timestamps, and additional verification data, like certificates, revocation information, trust anchors, policy details, role information, etc. Evidence data will be generated, managed and renewed by a special application in a secure environment outside the TOE. The TOE allows for a CS to request an evidence record for a particular archive data object.

3.1.3 Operations

Archive Requests

An archive request is a call from the Client Software Application to the TOE to perform a certain operation on the storage.

- **Submission** means that the Client Software Application wants to store a (new) submission data object[6] into the archive. The submission data object is included in this archive request. An already existing archive data object in the storage cannot be overwritten, updated or modified.
- **Retrieval** means that the Client Software Application wants to read out a particular archive data object[7] from the storage. Modification or update of this archive data object in the storage is not possible.
- **Erasure** means that the Client Software Application wants to delete a particular archive data object from the storage. An erasure request may happen before or after the retention time of the archive data object. The TOE enforces the submission of a justification if the archive data object shall be deleted before expiration of the retention time.

6 A submission data object can also be an evidence record object to be stored in the storage.

7 An archive data object can also be an evidence record object to be retrieved from the storage.

- **Request for evidence** means that the Client Software Application requests evidence to the fact that an archive data object or any collection of archive data objects does exist unmodified within the storage at a certain point of time. The returned expression must comply with the Evidence Record Syntax specified by the IETF [7].
- **Read metadata information** means that the Client Software Application wants to read out some meta information of one, some or all archive data objects stored in the storage. These meta information may contain search indices, ownership of archive data objects, retention times, digital signatures, etc.

Authentication of an XML schema

An XML schema can be authenticated by verification of the (optional) digital signature of this XML schema. The authentication fails

- if the signature is wrong or invalid, or
- if the certificate used for the signature could not be verified, or
- if the certificate used for the signature is not owned by an authorized organisation or
- if the signature does not exist.

Authentication of an archive request

An archive request can be authenticated by verification of the (optional) digital signature of this request or an archive request can be authenticated by the successful identification and authentication of the requesting CS. The authentication fails,

- if the CS cannot be identified, or
- if the password supplied by the successfully identified CS cannot be successfully checked.

The data needed for the identification and authentication of the requesting CS must be submitted by the TOE administrator to the Credential Store component of the TOE.

Check or verification of SDOs

Technically spoken, the submission data object is an XML package which contains all required information.

Verification of a submission data object means that the TOE verifies the XML structure of the submission data object against a defined XML schema [6].

Verification of an archive request

Technically spoken, the archive request is an XML package which contains all information about the request and all data relevant for this request.

Verification of such a request means that the TOE verifies the XML structure of the request against a defined XML schema [6].

Submission of an archive data object

See “Archive Request”

Retrieval of an archive data object

See “Archive Request”

Erasure of an archive data object

See “Archive Request”

Request for evidence

See “Archive Request”

Request meta information

See “Archive Request”

3.1.4 Security Attributes

Client Software Application Identity

All Client Software Applications shall have a unique identity, e.g. a numeric value or a unique name.

Owner

The Owner for a submission data object or an archive data object is the Client Software Application which initially submits this object to the archive.

The security attribute “Owner” stores the Client Software Application Identity of the respective application.

Long-term storage unit identity

Each long-term storage unit connected to the TOE or another trustworthy application which in turn connects to the long-term storage unit must have a unique identifier, e.g. a numeric value or a unique name. The TOE shall only connect to storage units/trustworthy applications whose identity is known to the TOE.

Submitter of a SDO

A submitter of a submission data object is a Client Software Application. The values of this security attribute are the unique identifiers of the Client Software Application (see “**Client Software Application Identity**”).

Retention Time

The retention time of a submission data object/archive data object is an attribute storing the date and time when this archive data object can be deleted without justification. The value will be specified for each submission data objects to be archived by the submitting client software application and will not be modified by the TOE or the Long-term storage unit.

Usually, this value lies 10 years or more in the future since submission.

Object ID (OID)

The object ID is a unique identifier of the submission data object a client software application has submitted for archiving. “Unique” means here “at least unique for the submitting client software application”.

Archive Object ID (AOID)

The archive object ID is a unique identifier of any archive data object stored in the Long-term storage unit. This ID will be generated outside the TOE, e.g. by the long-term storage unit or by a non-TOE part of the middleware, when a submission data object will be sent to the TOE and stored in the SU. This ID will be returned to the submitting client software application by the TOE, so that this application is able to retrieve or erase its archive data object sometimes in the future.

3.2 Assets

Protocol Data

Log information generated by the TOE, provided by the TOE for usage through the TOE's Administrator, and usually stored in the Long-term storage unit [8].

3.3 TSF Data

CS assigned configuration data

On start-up or during operation the Administrator must load CS assigned configuration data. The protection of the *integrity* of these CS assigned configuration data ensures the correct functionality resp. behaviour of the TOE to process the archive objects according to the rules defined by the organization using the TOE.

Credential data stored in the Credential Store component

On start-up or during operation the Administrator must load Client Software Application assigned credential data into the Credential Store component of the TOE. The protection of

8 It's in the responsibility of the integrator of the TOE to provide a mechanism to store the protocol data if needed.

the *integrity* of these credential data ensures the correct functionality resp. behaviour of the TOE to identify and authenticate the Client Software Applications.

3.4 Assumptions

The description of assumptions illustrates the security aspects of the environment in which the TOE is intended to be used.

A.ADMIN

The administrators of the TOE, of the underlying systems, of the communication connections (e.g. the LAN) and the long-term storage system are not careless, wilfully negligent, or hostile, and will follow and abide the instructions provided by the administrator's guidance. They are well trained to securely and trustworthy administer all aspects of TOE operation in accordance with the guidance. The administrators will protect their credentials used for authentication. Credentials must not be disclosed to other individual.

A.AUTHENT

All CS, which are authorized by the IT environment for archive requests, identify and authenticate the TOE before data transfer.

A.COMMUNICATION

The communication interconnections between the TOE and all external components are protected by the environment – by physical or logical security measures – against disclosure.

A.CONFIGURATION

The TOE is securely configured and all data required for the configuration of the TOE are securely and reliably transported to and installed on the machine which runs the TOE.

A.EVIDENCEDATA

The generation, management and renewal of evidence data for proving the unmodified existence of archive data objects at a certain time will be provided by trustworthy special applications in a secure non-TOE environment.

A.NO_BYPASS

The TOE is integrated in the IT environment in such a way that all storage access by the CS must pass the TOE.

A.PHYSPROT

The machine on which the TOE runs is protected against unauthorized physical access and modification.

A.SERVER

No other software application except the TOE is installed on the machine on which the TOE is running. All underlying systems are securely installed and protected against unauthorized physical and logical access and modification. The machine on which the TOE is running is free from malware and viruses.

A.STORAGE

The dedicated SU provides a reliable, secure and available storage of data, even for long-terms.

Logically or physically separated parts or components of the dedicated SU provide a reliable, secure and available storage of evidence data which may prove the existence and integrity of particular archive data objects at a certain time. The evidence data must comply with the requirements of the Evidence Record Syntax specified by the IETF [7].

The generation and management of the evidence data may be provided by the SU, or components of the SU itself or additional and trustworthy non-TOE parts of the environment interacting.

A.TIMESTAMP

The environment provides reliable time-stamps to the TOE.

A.TOKEN

The environment, e. g. the SU or a non-TOE part of the middleware provides a reliably generated unique archive object identifier (AOID) for any successfully archived data object.

A.TRUSTAPP

The archive requesting CS is secure, and provides reliable measures regarding the authentication and access authorization of users.

A.TRUSTCRYPTO

Only trustworthy cryptographic components are used. The cryptographic components do not send any security relevant and confidential data to any external entity and will reliably protect all security relevant and confidential data from disclosure by an external entity.

A special application or component in the non-TOE environment, which generates, manages and renews the evidence data and uses the trustworthy cryptographic components for executing required cryptographic operations through secure communication channels only is not regarded as an external entity within the frame of this security target.

A.XMLSCHEMA

For any CS using the TOE for submitting SDOs into the SU a valid data schema (XML schema) must exist. Schema instructions and rules defined for using the schema do not introduce any security risk.

3.5 Threats

The threat agents can be categorized as either

- Unidentified individuals or client software applications, i.e. entities not known by the TOE but having access to the communication interfaces exposed by the TOE or to the client software applications, or
- Identified users of the TOE, i.e. individuals or entities, which may access resources controlled by the TOE.

The threat agents are assumed to originate from a well-known user community in a non-hostile environment. The TOE therefore protects against threats of inadvertent or casual attempts to breach the system security. The TOE is not intended to be used in environments where protection is required against determined and hostile attacks to breach the system security at all. Resuming, the following threats need to be countered by the TOE:

T.CRYPTO

An attacker attempts to substitute the cryptographic component or to intercept and manipulate the communication between the TOE and the cryptographic component.

T.DATA_ACCESS1

An attacker attempts to gain unauthorized access to the archive, e.g. by sending manipulated AOIDs.

T.DATA_ACCESS2

An attacker attempts to gain unauthorized access to the archive, e.g. by simulating an authorized client software application.

T.ERASURE

A CS attempts to delete an archive object before expiry of its retention time without any justification.

T.INVALID_XML

The SDO submitted by a CS cannot be reliable interpreted by the TOE or does not correspond to an XML schema deposited within the TOE.

T.MODIFY

An attacker attempts to modify a submission data object in a specific manner during transmission between CS and the TOE.

T.SCHEMA

An XML schema assigned to a CS is not or invalid authorized.

T.STORAGE

An attacker attempts to substitute the SU or another trustworthy application which in turn is dedicated to forward the SDO to the SU or to manipulate the communication between the TOE and the SU or the other trusted application.

T.TOE_ACCESS

An attacker attempts to gain access to the internal data of the TOE and the resources it protects.

T.TOE_SPOOF

An attacker attempts to feign TOE functionalities to the CS.

3.6 Organizational Security Policies

P.ACCESS

The TOE only allows the following archive operations[9]:

- Submit submission data objects to the storage,
- Retrieve archive data objects from the storage,

9 An “archive operation” is the request of the Client Software Application (CS) to the TOE to execute defined operations in the archive system.

- Delete archive data objects from the storage
- Request for evidence of archive data objects and
- Reading metadata information of archive data objects.

P.ARCHIVE

The TOE submits successfully verified submission data objects to the SU only or to another trustworthy application which in turn is dedicated to forward the SDO to the SU. The verification assures that the XML document corresponds to the assigned XML schema and contains at least an object ID and a retention time.

P.OBJECT

The requesting CS assigns to any XML data package to be archived a unique object identifier (OID).

P.RETURN

After successful storage of a submission data object the TOE returns to the requesting CS the assigned object identifier and the archive object ID (AOID) generated by the environment of the TOE, e. g. the SU or any non-TOE part of the middleware.

P.SCHEMA

The TOE must select the right configuration data assigned to the requesting CS, must interpret it in a correct manner and execute the instructions / rules defined within in the configuration data in the right order.

P.STORAGE

The TOE must not interpret or change the archive object ID.

4 Security Objectives

This section defines the security objectives for the TOE and its supporting environment. The security objectives are categorized as security objectives for the TOE or for the environment.

4.1 Security Objectives for the TOE

O.ACCESS

The TOE allows the following operations only:

- Submit submission data objects [10] to the storage,
- Retrieve archive data objects [11] from the storage,
- Delete archive data objects from the storage,
- Request for evidence of archive data object, and
- Reading metadata information.

O.AO_EXAM

The TOE assures that only successfully verified submission data object will be submitted to the SU or another trustworthy application which in turn must forward the SDO to the SU. The verification assures at least the conformity of the data object with an assigned XML schema and in addition that the metadata of the data object contains an object ID and a retention time.

O.APPL_COMM

The TOE assures the authenticity and integrity of the archive requests by means of examining the authenticity and integrity of the client requests. Vice versa, the TOE adds to the request responses reliable authentication and integrity attributes.

10 A submission data object can also be an evidence record object to be stored in the storage.

11 An archive data object can also be an evidence record object to be retrieved from the storage.

O.CRYPTOPROV

The TOE assures that the selected (defined) trustworthy cryptographic component cannot be substituted unnoticed and will be exclusively used for all required cryptographic operation.

O.DATA_ACCESS

The TOE allows only authorized CS the submission of submission data objects (XML documents) to the SU; the access to archived data objects is restricted by the TOE for a requesting CS to only these archive data objects which have been submitted by this application.

O.ERASURE

The TOE assures that archive data objects can only be deleted by client requests before expiry of the retention time, when the delete request will be submitted together with a justification.

O.ERASURE_LOG

The TOE must log any delete requests and the accompanying justification for archive data objects, if the retention time of these archive objects is not yet expired.

O.RETURN

After successful storage of submission data object/archive data object the response of the TOE to the requesting CS must contain at least the archive object ID (AOID) and the object identifier of the CS. The TOE does not interpret, change or modify the AOID.

O.SCHEMA

The TOE assures the selection and application of the appropriate configuration assigned to the requesting client application, interprets the configuration data in a correct manner and executes the instructions / rules defined within in the configuration data in the right order.

O.SCHEMA_AUTH

The TOE checks the valid authorization of the XML schemas.

O.SCHEMA_EXAM

The TOE checks the conformity of the submitted submission data objects with the assigned XML schemas and assures the correct execution of additional instructions and/or rules defined in the configuration data.

O.STORAGE

The TOE assures that the selected and dedicated SU (or another trustworthy application which in turn forwards the SDO to the SU) will be used for saving the ADOs.

O.TOE_ACCESS

The TOE does not grant any access to TSF (TOE Security Functions) data. Configuration data are accessible for the administrator of the TOE only.

O.TOE_AUTHENT

The TOE is capable to authenticate itself reliably against external entities.

4.2 Security Objectives for the Operational Environment

OE.ADMIN

The administrators of the TOE, of the underlying systems, of the communication connections (e.g. the LAN) and the long-term storage system must not be careless, wilfully negligent, or hostile, and shall follow and abide the instructions provided by the administrator's guidance. They shall be well trained to securely and trustworthy administer all aspects of TOE operation in accordance with the guidance. The administrators shall protect their credentials used for authentication. Credentials must not be disclosed to other individual.

OE.AUTH_ATTR

The CS's identify and authenticate the TOE before any data transfer and protect the archive requests by means of reliable authentication and integrity attributes.

OE.COMMUNICATION

The communication interconnections between the TOE and all external components must be protected by the environment – by physical or logical security measures – against disclosure.

OE.CONFIGURATION

The TOE has to be securely configured and all data required for the configuration of the TOE must be securely and reliably transported to and installed on the machine on which the TOE is running.

OE.EVIDENCEDATA

The generation, storage, management and renewal of evidence data for proving the unmodified existence of archive data objects at a certain time is provided by trustworthy special applications in a secure non-TOE environment in accordance with the requirements of the Evidence Record Syntax specified by the IETF [7].

OE.NO_BYPASS

The TOE must be integrated in the IT environment in such a way that all storage access by the CS must pass the TOE.

OE.OBJECT

The requesting CS must provide and assign a unique object identifier (OID) to any submission data object to be archived.

OE.PHYSPROT

The machine on which the TOE is running must be protected against unauthorized physical access and modification.

OE.SERVER

No other software application except the TOE must be installed on the machine on which the TOE runs. All underlying systems must be securely installed and protected against unauthorized physical and logical access and modification. The machine on which the TOE runs must be free from malware and viruses.

OE.STORAGE

The dedicated SU must provide a reliable, secure and available storage of data, even for long-terms.

OE.TIMESTAMP

The environment shall provide reliable time-stamps to the TOE.

OE.TOKEN

The environment, e. g. the SU or a non-TOE part of the middleware must be able to generate reliably unique archive object identifier (AOID) for any successfully archived submission data object.

OE.TRUSTAPP

The archive requesting CS must be secure, and have to provide reliable measures regarding the authentication and access authorization of users.

OE.TRUSTCRYPTO

Only trustworthy cryptographic components shall be used. The cryptographic components may not send out any security relevant and confidential data to any external entity and shall reliable protect all security relevant and confidential data from disclosure by an external entity.

OE.XMLSCHEMA

For all CS must exist a valid data schema (XML schema). Schema instructions and rules defined for using the schema must not introduce any security risk.

4.3 Rationale For Security Objectives

This chapter explains how each aspect of the security environment of the TOE will be covered by the security objectives. In addition the security environment is explained.

The following table provides an overview for security objectives coverage.

	O.ACCESS	O.AO_EXAM	O.APPL_COMM	O.CRYPTOPROV	O.DATA_ACCESS	O.ERASURE	O.ERASURE_LOG	O.RETURN	O.SCHEMA	O.SCHEMA_AUTH	O.SCHEMA_EXAM	O.STORAGE	O.TOE_ACCESS	O.TOE_AUTHENT	OE.ADMIN	OE.AUTH_ATTR	OE.COMMUNICATION	OE.CONFIGURATION	OE.EVIDENCEDATA	OE.NO_BYPASS	OE.OBJECT	OE.PHYSPROT	OE.SERVER	OE.STORAGE	OE.TIMESTAMP	OE.TOKEN	OE.TRUSTAPP	OE.TRUSTCRYPTO	OE.XMLSCHEMA
T.CRYPTO				X																									
T.DATA_ACCESS1			X		X			X																					
T.DATA_ACCESS2			X		X			X								X													
T.ERASURE						X	X																						
T.INVALID_XML									X	X	X																		
T.MODIFY			X																										
T.SCHEMA										X																			
T.STORAGE												X																	
T.TOE_ACCESS	X												X																
T.TOE_SPOOF														X															
P.ACCESS	X																												
P.ARCHIVE		X																											
P.OBJECT																					X								
P.RETURN								X																					
P.SCHEMA									X																				
P.STORAGE								X																					
A.ADMIN															X														
A.AUTHENT																X													
A.COMMUNICATION																	X												
A.CONFIGURATION																		X											
A.EVIDENCEDATA																			X										
A.NO_BYPASS																				X									
A.PHYSPROT																						X							
A.SERVER																							X						
A.STORAGE																								X					
A.TIMESTAMP																									X				
A.TOKEN																										X			
A.TRUSTAPP																											X		
A.TRUSTCRYPTO																												X	
A.XMLSCHEMA																													X

Table 2: Security Objective Rationale

4.3.1 Coverage of the Assumptions

A.ADMIN: A.ADMIN assumes that the administrators for the TOE, of the underlying systems, of the communication connections (e. g. the LAN) and the storage system are not careless, wilfully negligent, or hostile, and will follow and abide the instructions provided by the administrator's guidance. They are well trained to securely and trustworthy administer all aspects of TOE operation in accordance with the TOE's security objectives. They will protect their credentials used for authentication against the TOE. Credentials must not be disclosed to other individual. The security objective OE.ADMIN for the operational environment covers this assumption.

A.AUTHENT: A.AUTHENT assumes that the authorized archive requesting CS will reliably identify and authenticate the TOE before any data transfer. This supports OE.AUTH_ATTR.

A.COMMUNICATION: A.COMMUNICATION assumes that communication interconnections between the TOE and all external components are protected by the environment – by physical or logical security measures – against disclosure. The security objective OE.COMMUNICATION for the operational environment covers this assumption.

A.CONFIGURATION: A.CONFIGURATION assumes that TOE is securely configured and all data required for the configuration of the TOE are securely and reliably transported and installed on the machine on which the TOE runs. The security objective OE.CONFIGURATION for the operational environment covers this assumption.

A.EVIDENCEDATA: A.EVIDENCEDATA assumes that evidence data for proving the unmodified existence of archive data objects at a certain time will be generated, stored, managed and renewed by trustworthy special applications in a secure non-TOE environment in accordance with the requirements of the Evidence Record Syntax specified by the IETF [7]. The security objective OE.EVIDENCEDATA for the operational environment covers this assumption.

A.NO_BYPASS: A.NO_BYPASS assumes that the TOE is integrated in the IT environment in such a way that all storage access by the clients must pass the TOE. The security objective OE.NO_BYPASS for the operational environment covers this assumption.

A.PHYSPROT: A.PHYSPROT assumes that the machine on which the TOE runs is protected against unauthorized physical access and modification. The security objective OE.PHYSPROT for the operational environment covers this assumption.

A.SERVER: A.SERVER assumes that no other software except the TOE is installed on the machine on which the TOE runs, that all underlying systems are securely installed and protected against unauthorized physical and logical access and modification, and that the machine on which the TOE runs is free from malware and viruses. The security objective OE.SERVER of the operational environment covers this assumption.

A.STORAGE: A.STORAGE assumes that the dedicated storage system provides a reliable, secure and available storage of the data even for long-terms. The security objective OE.STORAGE for the operational environment covers this assumption.

A.TIMESTAMP: A.TIMESTAMP assumes that the TOE is provided with reliable time-stamps by the environment of the TOE. The security objective OE.TIMESTAMP for the operational environment covers this assumption.

A.TOKEN: A.TOKEN assumes that the environment of the TOE, e. g. the SU or any non-TOE part of the middleware, generates reliably a unique archive object identifier (AOID) for any successfully archived data object. The security objective OE.TOKEN for the operational environment covers this assumption.

A.TRUSTAPP: A.TRUSTAPP assumes that the archive requesting client applications are secure, and provide reliable measures regarding the authentication and access authorization of users. The security objective OE.TRUSTAPP for the operational environment covers this assumption.

A.TRUSTCRYPTO: A.TRUSTCRYPTO assumes that only trustworthy cryptographic components are used, and the cryptographic components do not send any security relevant and confidential data to any external entity and will reliably protect all security relevant and confidential data from disclosure by an external entity. The security objective OE.TRUSTCRYPTO for the operational environment covers this assumption.

A.XMLSCHEMA: A.XMLSCHEMA assumes that for all client applications exists a valid data schema (XML schema), and the schema instructions and rules defined for using it will not introduce any risk. The security objective OE.XMLSCHEMA for the operational environment covers this assumption.

4.3.2 Encounter the Threats

T.CRYPTO: This threat covers attempts to substitute the cryptographic component or to intercept and manipulate the communication between the TOE and the cryptographic component. The security objective O.CRYPTOPROV encounters this threat.

T.DATA_ACCESS1: This threat focuses on any attempts to gain unauthorized access to the archive, e. g. by sending manipulated AOIDs. The security objectives O.DATA_ACCESS, O.APPL_COMM and O.RETURN encounter this threat.

T.DATA_ACCESS2: This threat focuses on attempts to gain unauthorized access to the archive, e. g. by simulating an authorized client application. The security objectives O.DATA_ACCESS, O.APPL_COMM, O.RETURN and OE.AUTH_ATTR encounter this threat.

T.ERASURE: This threat covers attempts to delete an archive object before expiry of the retention time of the archive object without any justification. The security objectives O.ERASURE and O.ERASURE_LOG encounter this threat.

T.INVALID_XML: This threat focuses on situations where a data object submitted by the client software application cannot be reliably interpreted by the TOE or doesn't correspond to the authorized XML schema deposited within the TOE. The security objectives O.SCHEMA_AUTH, O.SCHEMA_EXAM and O.SCHEMA encounter this threat.

T.MODIFY: This threat focuses on attempts to modify in a specific manner a data package during the transmission between the client applications and the TOE. The security objective O.APPL_COMM encounters this threat.

T.SCHEMA: This threat covers the situation that an XML schema assigned to a client software application is not or invalid authorized. The security objective O.SCHEMA_AUTH encounters this threat.

T.STORAGE: This threat covers attempts to substitute the storage system or the other trustworthy application interfacing with the SU or to manipulate the communication between the TOE and the storage system / the other trustworthy application. The security objective O.STORAGE encounters this threat.

T.TOE_ACCESS: This threat focuses on attempts to gain access to the internal data of the TOE and resources it protects. The security objective O.TOE_ACCESS and O.ACCESS encounter this threat.

T.TOE_SPOOF: This threat focuses on attempts to feign TOE functionalities to the client software applications. The security objective O.TOE_AUTHENT encounters this threat.

4.3.3 Implementation of Organizational Security Policies

P.ACCESS: This OSP focuses on the demand that the TOE allows only the following operations:

- Submit data objects [12] to the storage,
- Retrieve data objects [13] from the storage,
- Delete data objects within the storage,
- Request for evidence, and
- Read metadata information of archive objects.

The security objective O.ACCESS covers the OSP.

P.ARCHIVE: This OSP focuses on the demand that the TOE submits successful verified archive objects to the storage system. The verification assures that the XML document corresponds to the assigned XML schema and contains an object ID and a retention time at least. The security objective O.AO_EXAM covers the OSP.

P.OBJECT: This OSP focuses on the demand that the requesting client application to any XML data package to be archived assigns a unique object identifier (OID). The security objective OE.OBJECT for the operational environment covers the OSP.

P.RETURN: This OSP focuses on the demand that after successful storage of an data object the TOE returns to the requesting client application the archive object ID (AOID) generated outside the TOE, e.g. by the storage system or any non-TOE part of the middleware, and the assigned object identifier. The security objective O.RETURN covers the OSP.

P.SCHEMA: This OSP focuses on the selection of the right configuration data assigned to the requesting client application, the correct interpretation and execution of the instructions / rules within in the configuration data in a right order. The security objective O.SCHEMA covers the OSP.

12 A submission data object can also be an evidence record object to be stored in the storage.

13 An archive data object can also be an evidence record object to be retrieved from the storage.

P.STORAGE: This OSP focuses the demand that the TOE must not interpret or change the archive object ID generated by the storage system. The security objective O.RETURN covers the OSP.

5 Security Requirements

This section comprises security functional and security assurance requirements and the rationale.

All of the security functional requirements in this section have been drawn from Common Criteria Part 2 [2].

In this section the following typographic conventions have been used:

- Selections performed have been marked in *italics*.
- Assignments performed have been marked in **bold**.
- Refinements have been marked as underlined.
- Iterations of security requirements have been marked by applying an additional identifier to the appropriate component names.

5.1 Security Policies

Within this section the security policies the TOE shall implement will be defined.

5.1.1 Access Control Policy (TSP_ACC)

The TOE shall control the access to the archive according to the following rules:

- Only authorized Client Software Applications (CS) will get permission for accessing the archive by using valid archive requests.
- Access to Archive Objects will only be granted to this particular Client Software Application which has submitted the data object for archiving.

5.1.2 Information Flow Control Policy (TSP_IFC)

The TOE shall implement an information flow control policy which follows the following rules:

- The TOE accepts and performs only the following types of archive requests:
  - Submission of a data object to be archived
  - Request for retrieval of an archive object
  - Request for erasure of an archive object
  - Request for evidence
  - Request for reading meta information
- The TOE does not disclose any TOE data as a result of an archive request.
- All rules specified by the organization using the TOE shall be performed by the TOE in accordance with the specification and in the context of the respective archive request.
- All successfully checked and verified data objects will immediately be transferred to the archive.
- Erasure of an archive object before expiration of its retention time requires a justification submitted together with the archive request for erasure.
- The TOE shall return the object ID and the archive object ID as result of a submission archive request.
- The TOE must not perform an archive request, if the XML schema for the requesting CS cannot be authenticated.
- The TOE must not perform an archive request, if the archive request cannot be authenticated.
- The TOE must not perform an archive request, if the archive request cannot be successfully verified against the XML schema for the requesting CS.

5.2 Security Functional Requirements (SFRs)

5.2.1 Class FAU: Security Audit

FAU_GEN.1 Audit data generation

Hierarchical to: No other components.

Dependencies: FPT_STM.1 Reliable time stamps

FAU_GEN.1.1	The TSF shall be able to generate an audit record of the following auditable events: a) Start-up and shutdown of the audit functions; b) All auditable events for the *not specified* [14] level of audit; and c) • **Unsuccessful authentications of Client Software Applications, Crypto Providers, the long-term storage unit and other trustworthy applications connected to the TOE,** • **Unsuccessful authentication of an XML schema** • **Unsuccessful authentication or verification of an archive request** • **Unsuccessful access attempts to archive objects** • **Successful and unsuccessful erasure archive requests for archive objects whose retention time is not yet expired** • **other specifically defined auditable events: none** [15].
FAU_GEN.1.2	The TSF shall record within each audit record at least the following information a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and b) For each audit event type, based on the auditable event definitions of the functional components included in the ST [16], **for successful erasure archive requests for archive objects whose retention time is not yet expired, the justification, other audit relevant information: none** [17].

14 [selection, choose one of: *minimum, basic, detailed, not specified*]

15 [assignment: *other specifically defined auditable events*]

16 [refinement: *PP/ST*]

17 [assignment: *other audit relevant information*]

5.2.2 Class FDP: User Data Protection

FDP_ACC.1 Subset access control

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1 The TSF shall enforce the **TSP_ACC** [18] on

a) list of subjects: Client Software Applications
b) objects: Archive Object
c) operations: Submission, retrieval and erasure of archive objects, requests for evidence and reading of metadata information by Client Software Applications [19].

FDP_ACF.1 Security attribute based access control

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1 The TSF shall enforce the **TSP_ACC** [20] to objects based on the following:

a) list of subjects: Client Software Applications
- **Security Attribute: Client Software Application Identity**

b) objects: Archive Object
- **Security Attribute: Owner** [21].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **Only the owner of an archive object is authorized to access this archive object** [22].

18 [assignment: *access control SFP*]

19 [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

20 [assignment: *access control SFP*]

21 [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

22 [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

FDP_ACF.1.3	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **none** [23].
FDP_ACF.1.4	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **none** [24].

FDP_DAU.1 Basic Data Authentication

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_DAU.1.1	The TSF shall provide a capability to generate evidence that can be used as a guarantee of the validity of **Return Values the TOE sent to the Client Software Applications** [25].
FDP_DAU.1.2	The TSF shall provide **the Client Software Applications** [26] with the ability to verify evidence of the validity of the indicated information.

FDP_ETC.2 Export of user data with security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]

FDP_ETC.2.1	The TSF shall enforce the **TSP_IFC** [27] when exporting data objects [28], controlled under the SFP(s), to the long-term storage unit or another trustworthy application [29].

23 [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]

24 [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

25 [assignment: *list of objects or information types*]

26 [assignment: *list of subjects*]

27 [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

FDP_ETC.2.2	The TSF shall export the data object [30] with the data object's [31] associated security attributes.
FDP_ETC.2.3	The TSF shall ensure that the security attributes, when exported to the long-term storage unit or another trustworthy application [32], are unambiguously associated with the exported data object [33].
FDP_ETC.2.4	The TSF shall enforce the following rules when a data object is exported from the TOE to the long-term storage unit or to another trustworthy application [34]: **The data object shall be augmented with the ID of the submitting Client Software Application** [35].

FDP_IFC.1 Subset information flow control

Hierarchical to: No other components.

Dependencies: FDP_IFF.1 Simple security attributes

28 [refinement: *user data*]

29 [refinement: *outside of the TOE*]

30 [refinement: *user data*]

31 [refinement: *user data*]

32 [refinement: *outside of the TOE*]

33 [refinement: *user data*]

34 [refinement]

35 [assignment: *additional exportation control rules*]

FDP_IFC.1.1	The TSF shall enforce the **TSP_IFC**[36] on **Subjects: Client Software Applications, Long-term storage unit or another trustworthy application which in turn connects to the Long-term storage unit** **Information: Data objects, Archive Objects** **Operations: Archive Requests**[37].

FDP_IFF.1 Simple security attributes

Hierarchical to:	No other components.	
Dependencies:	FDP_IFC.1	Subset information flow control
	FMT_MSA.3	Static attribute initialisation

36 [assignment: *information flow control SFP*]

37 [assignment: *list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP*]

FDP_IFF.1.1	The TSF shall enforce the **TSP_IFC** [38] based on the following types of subject and information security attributes: • **Subject: Client Software Applications,** o **Security Attributes: Client Software Application identity** • **Subject: Long-term storage unit or another trustworthy application which in turn connects to the Long-term storage unit** o **Security Attributes: Long-term storage unit identity or application identity** • **Information: Data objects** o **Security Attributes: Submitter of the data object** • **Information: Archive Objects** o **Security Attributes: Owner of the archive object** [39].
FDP_IFF.1.2	The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: • **The submitter identity of a data object is identical to the submitting Client Software Application** • **The identity of the requesting Client Software Application is identical to the owner of the archive object** [40].

38 [assignment: *information flow control SFP*]

39 [assignment: *list of subjects and information controlled under the indicated SFP, and for each, the security attributes*]

40 [assignment: *for each operation, the security attribute-based relationship that must hold between subject and information security attributes*]

FDP_IFF.1.3	The TSF shall enforce the **following rules**: • **The TOE only accepts and performs archive requests of the type "Submission", "Retrieval", "Erasure", "Request for evidence" or "Read metadata information"** • **All successfully checked and verified data objects shall immediately be transferred to the long-term storage unit or another trustworthy application which in turn forwards the SDO to the long-term storage unit** • **The TOE shall return the object ID and the archive object ID to the submitting Client Software Application without interpretation as result of a successful submission archive request** • **Erasure of an archive object before expiration of its retention time requires a justification submitted together with the erasure archive request** • **Data objects shall only be transferred to the long-term storage unit or another trustworthy application which in turn forwards the SDO to the long-term storage unit, if all checks are successfully executed** • **Archive objects shall only be transferred to the requesting CS, if the CS is the owner of this archive object** [41].
FDP_IFF.1.4	The TSF shall explicitly authorise an information flow based on the following rules: • **none** [42].

41 [assignment: *additional information flow control SFP rules*]

42 [assignment: *rules, based on security attributes, that explicitly authorise information flows*]

FDP_IFF.1.5	The TSF shall explicitly deny an information flow based on the following rules: • **The TOE must not perform an archive request, if the XML schema for the requesting CS cannot be authenticated or the issuing organization is not an authorized organization** • **The TOE must not perform an archive request if the archive request cannot be authenticated** • **The TOE must not perform an archive request if the archive request cannot be successfully verified against the XML schema for the requesting CS** [43].

FDP_ITC.1 Import of user data without security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FMT_MSA.3 Static attribute initialisation

FDP_ITC.1.1	The TSF shall enforce the **TSP_IFC** [44] when importing data objects [45], controlled under the SFP, from submitting Client Software Applications [46].
FDP_ITC.1.2	The TSF shall ignore any security attributes associated with the data objects [47] when imported from submitting Client Software Applications [48].

43 [assignment: *rules, based on security attributes, that explicitly deny information flows*]

44 [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

45 [refinement: *user data*]

46 [refinement: *outside of the TOE*]

47 [refinement: *user data*]

48 [refinement: *outside the TOE*]

FDP_ITC.1.3	The TSF shall enforce the following rules when importing data objects [49] controlled under the SFP from submitting Client Software Applications [50]: • **The data object shall conform to the XML schema assigned to the submitting Client Software Application** • **The meta information of the data object shall contain at least an object ID and a retention time** • **The TOE shall execute the rules specified by the organization using the TOE** [51].

Application note: This SFR ensures the correct import (which is actually a verification) of a data object submitted by a Client Software Application.

FDP_ITC.2 (AREQ) Import of user data with security attributes

Hierarchical to:	No other components.	
Dependencies:	[FDP_ACC.1	Subset access control, or
	FDP_IFC.1	Subset information flow control]
	[FTP_ITC.1	Inter-TSF trusted channel, or
	FTP_TRP.1	Trusted path]
	FPT_TDC.1	Inter-TSF basic TSF data consistency

FDP_ITC.2.1	The TSF shall enforce the **TSP_IFC** [52] when importing archive requests [53], controlled under the SFP, from submitting Client Software Applications [54].

49 [refinement: *user data*]

50 [refinement: *outside the TOE*]

51 [assignment: *additional importation control rules*]

52 [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

53 [refinement: *user data*]

54 [refinement: *outside of the TOE*]

FDP_ITC.2.2	The TSF shall use the security attributes associated with the imported archive requests[55].
FDP_ITC.2.3	The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the archive requests[56] received.
FDP_ITC.2.4	The TSF shall ensure that interpretation of the security attributes of the imported archive requests[57] is as intended by the source of the archive requests[58].
FDP_ITC.2.5	The TSF shall enforce the following rules when importing archive requests[59] controlled under the SFP from submitting Client Software Applications[60]: • **The imported security attributes of an archive request shall proof the integrity and authenticity of the archive request**[61].

Application Note: This SFR ensures the integrity and authenticity of archive requests.

FDP_ITC.2 (CSID) Import of user data with security attributes

Hierarchical to:	No other components.	
Dependencies:	[FDP_ACC.1	Subset access control, or
	FDP_IFC.1	Subset information flow control]
	[FTP_ITC.1	Inter-TSF trusted channel, or
	FTP_TRP.1	Trusted path]
	FPT_TDC.1	Inter-TSF basic TSF data consistency

55 [refinement: *user data*]

56 [refinement: *user data*]

57 [refinement: *user data*]

58 [refinement: *user data*]

59 [refinement: *user data*]

60 [refinement: *outside the TOE*]

61 [assignment: *additional importation control rules*]

FDP_ITC.2.1	The TSF shall enforce the **TSP_IFC**[62] when importing archive objects[63], controlled under the SFP, from the long-term storage unit or another trusted application which in turn interfaces with the long-term storage unit[64].
FDP_ITC.2.2	The TSF shall use the security attributes associated with the imported archive objects[65].
FDP_ITC.2.3	The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the archive objects[66] received.
FDP_ITC.2.4	The TSF shall ensure that interpretation of the security attributes of the imported archive objects[67] is as intended by the source of the archive objects[68].
FDP_ITC.2.5	The TSF shall enforce the following rules when importing archive objects[69] controlled under the SFP from the long-term storage unit or another trusted application which in turn interfaces with the long-term storage unit[70]: • **The imported security attributes shall identify the owner of the archive object**[71].

62 [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

63 [refinement: *user data*]

64 [refinement: *outside of the TOE*]

65 [refinement: *user data*]

66 [refinement: *user data*]

67 [refinement: *user data*]

68 [refinement: *user data*]

69 [refinement: *user data*]

70 [refinement: *outside the TOE*]

71 [assignment: *additional importation control rules*]

Application Note: This SFR ensures that the ownership of an archive object will be imported from the long-term storage unit.

FDP_ITC.2 (SCHEMA) Import of user data with security attributes

Hierarchical to:	No other components.	
Dependencies:	[FDP_ACC.1	Subset access control, or
	FDP_IFC.1	Subset information flow control]
	[FTP_ITC.1	Inter-TSF trusted channel, or
	FTP_TRP.1	Trusted path]
	FPT_TDC.1	Inter-TSF basic TSF data consistency

FDP_ITC.2.1	The TSF shall enforce the **TSP_IFC** [72] when importing XML schemas [73], controlled under the SFP, from outside of the TOE.
FDP_ITC.2.2	The TSF shall use the security attributes associated with the imported XML schemas [74].
FDP_ITC.2.3	The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the XML schemas [75] received.
FDP_ITC.2.4	The TSF shall ensure that interpretation of the security attributes of the imported XML schemas [76] is as intended by the source of the XML schemas [77].

72 [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

73 [refinement: *user data*]

74 [refinement: *user data*]

75 [refinement: *user data*]

76 [refinement: *user data*]

77 [refinement: *user data*]

FDP_ITC.2.5	The TSF shall enforce the following rules when importing XML schemas [78] controlled under the SFP from outside the TOE: • **The imported security attributes of an XML schema shall proof the integrity and authenticity of the XML schema** • **The imported security attributes of an XML schema shall identify the issuing organization** [79].

Application Note: This SFR ensures that all XML schemas must be upright and authorized by the issuing organization.

5.2.3 Class FIA: Identification and Authentication

FIA_UAU.2 User authentication before any action

Hierarchical to:	FIA_UAU.1	Timing of authentication
Dependencies:	FIA_UID.1	Timing of identification

FIA_UAU.2.1	The TSF shall require each Client Software Application [80] to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that Client Software Application [81].

FIA_UID.2 User identification before any action

Hierarchical to:	FIA_UID.1	Timing of identification
Dependencies:	No dependencies.	

78 [refinement: *user data*]

79 [assignment: *additional importation control rules*]

80 [refinement: *user*]

81 [refinement: *user*]

FIA_UID.2.1	The TSF shall require each Client Software Application [82] to be successfully identified before allowing any other TSF-mediated actions on behalf of that Client Software Application [83].

5.2.4 Class FMT: Security management

FMT_MSA.1 (FLOW) Management of security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1	The TSF shall enforce the **TSP_IFC** [84] to restrict the ability to *modify or delete* [85] the security attributes **Client Software Application identity, Long-term storage unit identity or trustworthy application identity, Submitter of the data object, Owner of the archive object** [86] to **nobody** [87].

FMT_MSA.3 (ACCESS) Static attribute initialisation

Hierarchical to: No other components.

Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

82 [refinement: *user*]

83 [refinement: *user*]

84 [assignment: *access control SFP(s), information flow control SFP(s)*]

85 [selection: *change_default, query, modify, delete, [assignment: other operations]*]

86 [assignment: *list of security attributes*]

87 [assignment: *the authorised identified roles*]

FMT_MSA.3.1	The TSF shall enforce the **TSP_ACC** [88] to provide *restrictive* [89] default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2	The TSF shall allow **nobody** [90] to specify alternative initial values to override the default values when an object or information is created.

Application Note: This SFR shall ensure that the ownership of a data object submitted by a client application to be archived is per default "nobody" or any other neutral identity else, to prevent an unauthorized access. Of course, the value of this security attribute of a particular archive object can be changed / defined before it is stored in the long-term storage unit, e.g. by declaration within the meta data. This functionality and the configuration of the authorized Client Software Applications are here per definition out of the TOE scope.

FMT_MSA.3 (FLOW) Static attribute initialisation

Hierarchical to:	No other components.	
Dependencies:	FMT_MSA.1	Management of security attributes
	FMT_SMR.1	Security roles

FMT_MSA.3.1	The TSF shall enforce the **TSP_IFC** [91] to provide *restrictive* [92] default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2	The TSF shall allow **nobody** [93] to specify alternative initial values to override the default values when an object or information is created.

Application Note: This SFR ensures that all security attributes relevant for the information flow control (e.g. the possible types of archive requests) will be initialized with secure default values.

88 [assignment: *access control SFP(s), information flow control SFP(s)*]

89 [selection, choose one of: *restrictive, permissive, [assignment: other property]*]

90 [assignment: *the authorised identified roles*]

91 [assignment: *access control SFP(s), information flow control SFP(s)*]

92 [selection, choose one of: *restrictive, permissive, [assignment: other property]*]

93 [assignment: *the authorised identified roles*]

FMT_SMR.1 Security roles

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.1.1	The TSF shall maintain the roles **authorized Client Software Application** [94].
FMT_SMR.1.2	The TSF shall be able to associate users with roles.

Application Note: The roles "Administrator" and "Organization using the TOE" will be defined by the operational environment and are not maintained by the TSF. Here, "Users" are the different client software applications accessing the archive.

5.2.5 Class FPT: Protection of the TSF

FPT_TDC.1 Inter-TSF basic TSF data consistency

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_TDC.1.1	The TSF shall provide the capability to consistently interpret **TOE configuration data** [95] when shared between the TSF and the underlying system [96].
FPT_TDC.1.2	The TSF shall use **none** [97] when interpreting the TSF data from the underlying system [98].

94 [assignment: *the authorised identified roles*]

95 [assignment: *list of TSF data types*]

96 [refinement: *another trusted IT product*]

97 [assignment: *list of interpretation rules to be applied by the TSF*]

98 [refinement: *another trusted IT product*]

5.2.6 Class FTP: Trusted path/channels

FTP_ITC.1 (CRYPTO) Inter-TSF trusted channel

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_ITC.1.1	The TSF shall provide a communication channel between itself and the trusted crypto provider [99] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.
FTP_ITC.1.2	The TSF shall permit *the TSF* [100] to initiate communication via the trusted channel.
FTP_ITC.1.3	The TSF shall initiate communication via the trusted channel for **performing all types of cryptographic operations** [101].

FTP_ITC.1 (CS) Inter-TSF trusted channel

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_ITC.1.1	The TSF shall provide a communication channel between itself and a Client Software Application [102] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

99 [refinement: *itself and another trusted IT product*]

100 [selection: *the TSF, another trusted IT product*]

101 [assignment: *list of functions for which a trusted channel is required*]

102 [refinement: *another trusted IT product*]

FTP_ITC.1.2	The TSF shall permit the *Client Software Application* [103] to initiate communication via the trusted channel.
FTP_ITC.1.3	The TSF shall initiate communication via the trusted channel for **nothing** [104].

FTP_ITC.1 (STORAGE) Inter-TSF trusted channel

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_ITC.1.1	The TSF shall provide a communication channel between itself and a long-term storage unit or another trustworthy application which in turn connects to the long-term storage unit [105] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.
FTP_ITC.1.2	The TSF shall permit *the TSF* [106] to initiate communication via the trusted channel.
FTP_ITC.1.3	The TSF shall initiate communication via the trusted channel for • **storing archive objects in the long-term storage unit** • **retrieving archive objects from the long-term storage unit** • **erasing archive objects from the long-term storage unit** • **retrieving evidence records** • **reading out meta information from the long-term storage unit** [107].

103 [selection: *the TSF, another trusted IT product*]

104 [assignment: *list of functions for which a trusted channel is required*]

105 [refinement: *another trusted IT product*]

106 [selection: *the TSF, another trusted IT product*]

107 [assignment: *list of functions for which a trusted channel is required*]

5.3 Security Assurance Requirements (SARs)

The security assurance requirements for the evaluation of the TOE and its development and operating environment are those taken from the

Evaluation Assurance Level 4 (EAL 4).

augmented by the following component:

ALC_FLR.1.

The following “Table 3” gives an overview on the security assurance requirements that have to be fulfilled by the TOE.

Assurance class	**Assurance components**
ADV: Development	ADV_ARC.1 Security architecture description
	ADV_FSP.4 Complete functional specification
	ADV_IMP.1 Implementation representation of the TSF
	ADV_TDS.3 Basic modular design
AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
ALC: Live-cycle support	ALC_CMC.4 Production support, acceptance procedures and automation
	ALC_CMS.4 Problem tracking CM coverage
	ALC_DEL.1 Delivery procedures
	ALC_DVS.1 Identification of security measures
	ALC_FLR.1 Basic flaw remediation

Assurance class	Assurance components
	ALC_LCD.1 Developer defined life-cycle model
	ALC_TAT.1 Well-defined development tools
ASE: Security target evaluation	ASE_CCL.1 Conformance claims
	ASE_ECD.1 Extended components definition
	ASE_INT.1 ST introduction
	ASE_OBJ.2 Security objectives
	ASE_REQ.2 Derived security requirements
	ASE_SPD.1 Security problem definition
	ASE_TSS.1 TOE summary specification
ATE: Tests	ATE_COV.2 Analysis of coverage
	ATE_DPT.1 Testing: basic design
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing – sample
AVA: Vulnerability assessment	AVA_VAN.3 Focused vulnerability analysis

Table 3: TOE Security Assurance Requirements

5.4 Rationale for the Security Functional Requirements

The following table indicates that the security objectives pointed out in section 4.1 will be covered by the security functional requirements represented in section 5.2 of this Security Target.

Security objective	SFR
O.ACCESS	FDP_IFC.1, FDP_IFF.1, FMT_MSA.1 (FLOW), FMT_MSA.3 (FLOW)
O.AO_EXAM	FDP_IFC.1, FDP_IFF.1, FDP_ITC.1, FMT_MSA.1 (FLOW), FMT_MSA.3 (FLOW)
O.APPL_COMM	FAU_GEN.1, FDP_DAU.1, FDP_ITC.2 (AREQ), FIA_UID.2, FTP_ITC.1 (CS), FMT_SMR.1
O.CRYPTOPROV	FTP_ITC.1 (CRYPTO)
O.DATA_ACCESS	FAU_GEN.1, FDP_ACC.1, FDP_ACF.1, FDP_ETC.2, FDP_ITC.2 (CSID), FIA_UAU.2, FIA_UID.2, FTP_ITC.1 (CS), FMT_SMR.1, FMT_MSA.3 (ACCESS)
O.ERASURE	FDP_IFC.1, FDP_IFF.1
O.ERASURE_LOG	FAU_GEN.1
O.RETURN	FDP_IFC.1, FDP_IFF.1, FMT_MSA.1 (FLOW), FMT_MSA.3 (FLOW)
O.SCHEMA	FDP_IFC.1, FDP_IFF.1, FDP_ITC.1, FIA_UID.2
O.SCHEMA_AUTH	FAU_GEN.1, FDP_ITC.2 (SCHEMA), FPT_TDC.1
O.SCHEMA_EXAM	FDP_ITC.1
O.STORAGE	FTP_ITC.1 (STORAGE)
O.TOE_ACCESS	FDP_IFC.1, FDP_IFF.1, FMT_MSA.1 (FLOW), FMT_MSA.3 (FLOW)
O.TOE_AUTHENT	FTP_ITC.1 (CS), FTP_ITC.1 (CRYPTO), FTP_ITC.1 (STORAGE)

Table 4: Coverage of the Security Objectives by Security Functional Requirements

In the following it is pointed out how each of the security objectives is covered by the security functional requirements:

O.ACCESS:

FMT_MSA.1 (FLOW) and FMT_MSA.3 (FLOW) enforce that nobody will be able to modify or delete internal TOE data, which includes the types of archive requests. FDP_IFC.1 and FDP_IFF.1 guarantee that the TOE will only allow these types of archive requests.

O.AO_EXAM:

FDP_IFC.1, FDP_IFF.1 and FDP_ITC.1 enforce that only data objects, which has been successfully verified for being conform with an assigned XML schema and for containing an object ID and a retention time, will be submitted (even indirectly) to the long-term storage unit. FMT_MSA.1 (FLOW) and FMT_MSA.3 (FLOW) assure the possible results values of the verification (the reference values inside the TOE for “successful” and “not successful”) cannot be tampered.

O.APPL_COMM:

FDP_ITC.2 (AREQ) enforces that the authenticity and integrity of any archive request will be checked. FIA_UID.2 and FTP_ITC.1 (CS) support the authenticity and integrity checks of the archive requests by establishing a trustworthy channel between CS and TOE and the identification of the CS. FDP_DAU.1 assures that the TOE adds to the request responses reliable authentication and integrity attributes. FMT_SMR.1 assures the assignment of allowed archive requesting roles. FAU_GEN.1 records all illegal or invalid archive requests.

O.CRYPTOPROV:

FTP_ITC.1 (CRYPTO) enforces a reliable identification of the dedicated crypto provider. Thus, the defined trustworthy cryptographic component cannot be substituted unnoticed.

O.DATA_ACCESS:

FIA_UAU.2 and FIA_UID.2 enforce the identification and authentication of all requesting CS. FTP_ITC.1 (CS) supports that only identified and authenticated client software applications are allowed to communicate with the TOE. FDP_ETC.2 guarantees that any archived data object has been augmented with the ID of the submitting client software application. Thus, FDP_ACC.1 and FDP_ACF.1 can enforce that only the real submitter / owner of an archive object will have access to this archive object. FDP_ITC.2 (CSID) supports this by analysing the owner ID stored in the metadata of the archive object. FMT_MSA.3 (ACCESS) enforces the reliable assignment of the client software application ID to the data objects to be archived. FMT_SMR.1 assures the assignment of allowed archive requesting roles. FAU_GEN.1, in addition, will record any unsuccessful, i.e. unidentified or unauthenticated, archive requests.

O.ERASURE:

FDP_IFC.1 and FDP_IFF.1 enforce that nobody will be able to delete an archive object before the expiry of its retention time without any justification.

O.ERASURE_LOG:

FAU_GEN.1 guarantees that any erasure request to archive objects before the expiry of their retention time will be recorded.

O.RETURN:

FDP_IFC.1 and FDP_IFF.1 enforce that the TOE after successful storage of a data object returns the archive object ID (AOID) and the object identifier of the client software application to the submitting client software application. FMT_MSA.1 (FLOW) and FMT_MSA.3 (FLOW) assure the correct assignment between data object ID and archive object ID.

O.SCHEMA:

FDP_IFC.1, FDP_IFF.1 and FDP_ITC.1 assure that the TOE interprets the configuration data in a correct manner and executes the instructions / rules defined within the configuration data in the right order. FIA_UID.2 supports this by identification of the requesting CS because the ID of this CS may be used to identify the appropriate schema and rules.

O.SCHEMA_AUTH:

FPT_TDC.1 assures a reliable communication with the underlying system when importing configuration data, like XML schema. FDP_ITC.2 (SCHEMA) enforces the validity check of authorization of the XML schemas. FAU_GEN.1 guarantees that any attempt to deposit an XML schema without an authorization or with an invalid authorization doesn't remain unnoticed.

O.SCHEMA_EXAM:

FDP_ITC.1 enforces that the TOE checks the conformity of the submitted archive requests with the assigned XML schemas.

O.STORAGE:

FTP_ITC.1 (STORAGE) enforces that the selected and dedicated long-term storage unit or another trusted application which in turn connects to the long-term storage will be identified and authenticated before it will be used for saving the archive objects by the TOE.

O.TOE_ACCESS:

FDP_IFC.1 and FDP_IFF.1 assure that the TOE does not grant any access to TOE data. FMT_MSA.1 (FLOW) ensures that nobody can modify or delete TOE data. FMT_MSA.3 (FLOW) does not allow the change of this default.

O.TOE_AUTHENT:

FTP_ITC.1 (CS), FTP_ITC.1 (CRYPTO) and FTP_ITC.1 (STORAGE) guarantee that the TOE is capable to authenticate itself reliably against external entities.

5.5 Rationale for the Security Assurance Requirements

The evaluation assurance level EAL4 with augmentations chosen in this security target permits a developer to gain maximum assurance from positive security engineering based on good commercial development practices which, though rigorous, do not require substantial specialist knowledge, skills, and other resources.

The selection of the component **ALC_FLR.1** requiring the TOE developer to track and correct flaws in the TOE subsequently provides assurance that the TOE will be maintained and supported in the future.

All dependencies resulting directly or indirectly from the augmentation ALC_FLR.1 are discussed in the following:

The component **ALC_FLR.1** has no dependencies.

5.6 Rationale for the Security Functional Requirements and their dependencies

The following table shows the security functional requirements of the TOE, their dependencies and how these dependencies are resolved.

SFR	Dependencies	Resolved
FAU_GEN.1	FPT_STM.1	Resolved by the TOE environment
FDP_ACC.1	FDP_ACF.1	Resolved
FDP_ACF.1	FDP_ACC.1	Resolved
	FMT_MSA.3	Resolved by FMT_MSA.3 (ACCESS)
FDP_DAU.1	No dependency	---
FDP_ETC.2	FDP_ACC.1 or FDP_IFC.1	Resolved by FDP_IFC.1

SFR	Dependencies	Resolved
FDP_IFC.1	FDP_IFF.1	Resolved
FDP_IFF.1	FDP_IFC.1	Resolved
	FMT_MSA.3	Resolved by FMT_MSA.3 (FLOW)
FDP_ITC.1	FDP_ACC.1 or FDP_IFC.1	Resolved by FDP_IFC.1
	FMT_MSA.3	Resolved by FMT_MSA.3 (FLOW)
FDP_ITC.2 (AREQ)	FDP_ACC.1 or FDP_IFC.1	Resolved by FDP_IFC.1
	FPT_TDC.1	Resolved
	FTP_ITC.1 or FTP_TRP.1	Resolved by FTP_ITC.1 (CS)
FDP_ITC.2 (CSID)	FDP_ACC.1 or FDP_IFC.1	Resolved by FDP_IFC.1
	FPT_TDC.1	Resolved
	FTP_ITC.1 or FTP_TRP.1	Resolved by FTP_ITC.1 (STORAGE)
FDP_ITC.2 (SCHEMA)	FDP_ACC.1 or FDP_IFC.1	Resolved by FDP_IFC.1
	FPT_TDC.1	Resolved
	FTP_ITC.1 or FTP_TRP.1	Resolved by FTP_ITC.1 (CRYPTO)
FIA_UAU.2	FIA_UID.1	Resolved by hierarchical FIA_UID.2
FIA_UID.2	No dependency	---
FMT_MSA.1 (FLOW)	FDP_ACC.1 or FDP_IFC.1	Resolved by FDP_IFC.1
	FMT_SMF.1	Not resolved because the TOE does not have management functions.
	FMT_SMR.1	Resolved
FMT_MSA.3 (ACCESS)	FMT_MSA.1	Not resolved because the management of these security attributes is out of TOE scope.
	FMT_SMR.1	Resolved
FMT_MSA.3 (FLOW)	FMT_MSA.1	Resolved by FMT_MSA.1 (FLOW)
	FMT_SMR.1	Resolved
FMT_SMR.1	FIA_UID.1	Resolved by hierarchical FIA_UID.2
FPT_TDC.1	No dependency	---
FTP_ITC.1 (CRYPTO)	No dependency	---
FTP_ITC.1 (CS)	No dependency	---
FTP_ITC.1 (STORAGE)	No dependency	---

Table 5: Rationale for the Security Functional Requirements and their dependencies

6 TOE Summary Specification

The following paragraph provides a TOE summary specification describing how the TOE meets each SFR.

6.1 SF 1: Secure Client TOE Access

The TOE controls the access to the archive permitting archive requests only from successfully authenticated CS. Therefore the TOE owns a reliable identification and authentication process using a single-instance of the TOE component *Credential Store* conducting the identification and authentication. The identification of the CS is done via its name. The authentication is done via signed-challenge authentication (FDP_ACC.1, FDP_ACF.1, FIA_UAU.2, FIA_UID.2, FMT_MSA.3 (ACCESS), and FMT_SMR.1).

Before using the TOE the *Credential Store* has to be filled by the TOE's administrator with the identification and authentication data of the CS that intends to use the TOE.

If the client isn't successfully authenticated by the TOE, the client's request is rejected. If the client is successfully identified and authenticated, the TOE generates a so-called session ID for the CS returning this session ID to the CS which must permanently store this session ID. In every further archive request, the client has to use his session ID ensuring that the submitter identity is identical to the submitting CS (FDP_IFC.1, FDP_IFF.1). A client can request more than one session ID. All unsuccessful client requests are audited by the TOE (FAU_GEN.1).

If the TOE receives a submission archive request of a successfully authenticated CS, the CS identity will be closely connected to the submitted data objects. Thus the TOE augments the data objects to be archived with the ID of the submitting client software application (FDP_ETC.2).

Access to archive objects will only be granted to this particular CS which has submitted the data object for archiving (FDP_ACC.1, FDP_ACF.1, FMT_MSA.1 (FLOW), and FMT_MSA.3 (ACCESS)). Therefore the successfully identified and authenticated CS has to use one of his session IDs.

Before being successfully authenticated and identified the TOE doesn't allow any archive requests to a CS (FIA_UAU.2, FIA_UID.2, FDP_IFC.1, FDP_IFF.1, and FMT_MSA.1 (FLOW)).

For a successfully identified and authenticated CS the TOE allows the following request types (FDP_IFF.1, FMT_MSA.1 (FLOW), and FMT_MSA.3 (FLOW)):

- Request for storing data objects in the storage
- Request for retrieving data objects from the storage
- Request for erasing data objects from the storage
- Request for retrieving evidence records
- Request for reading meta information

As the result of an archive request the TOE returns at least the OID and the AOID having received from the SU to the submitting CS without interpretations (FDP_IFC.1, FDP_IFF.1) providing an unambiguous association between the archive request and the meta information of the ADO as proof of the integrity and authenticity of the archive request (FDP_ITC.2 (AREQ)) and as proof of the identity of the owner of the ADO (FDP_ITC.2 (CSID), FMT_MSA.1 (FLOW), and FMT_MSA.3 (FLOW)).

The authentication of the TOE by the CS has to been done by the TOE integrator through a programmatical validation of the SHA-256 value of each delivered Integrator-specific JAR archive. The authentication of the CS by the TOE is done through a signed-challenge authentication mechanism between the TOE and the CS. The description how the TOE technically meets the requirement FTP_ITC.1 (CS) is done in [23, chapter 3.2.2]. The communication via this trusted channel is always initiated by the CS (FDP_IFC.1).

Because of the TOE's architectural structure realizing the external user interfaces in form of Java functional calls, the need for the TOE to generate a guarantee of authenticity of the information content is not needed here. There is no channel in a physical sense used for information flow so the requirement FDP_DAU.1 is inherently fulfilled.

6.2 SF 2: Data Object Verification

The TOE prevents the storage of invalid data objects by reliable verification of the submitted data objects before forwarding them to the SU (or another trusted application which in turn forwards the SDOs to the SU) through the following measures:

If a CS requests the TOE for submission of a data object to a dedicated archive, the TOE validates this data object taking in account the referenced XML schema used for creation of the data object which the CS has submitted along with this submission archive request (FDP_ITC.1).

XML schemas can only be loaded into the TOE through the use of the function *addProfile*. This function can only be used by a successfully authenticated user. For the authorization of the import of the XML schema the user wants to use, this user has to calculate a so-called *ProfileIdentToken* ("*ProfileID*") as a concatenation of the SHA-256 hash values of the XML filter, the XML schema and its dependencies he wants to import into the TOE. This *ProfileID* has to be sent by the user in the function *addProfile* along with the XML schema and XML filter. The TOE only imports these XML schemas if the verification of the submitted data and the according *ProfileID* succeeds (FDP_ITC.2 (SCHEMA)). For any further archive request made by a TOE user, this *ProfileID* has to be part of the request so that the TOE can ensure the interrelation between the archive request and the corresponding XML schema (FDP_ITC.1).

While importing these XML schemas the TOE checks if the XML schema data are syntactically correct according to the XML Schema Definition [6]. The TOE refuses to import any XML schemas not being syntactically upright.

In case the XML schema is changed through an authorized or unauthorized action, the CS gets an appropriate message unambiguously saying that this *ProfileID* cannot be used by the CS. For the submission of data objects the CS must submit this *ProfileID*. The TOE is capable to consistently interpret these XML schemas as part of the TOE configuration data being shared with the underlying system (FPT_TDC.1). If the TOE could not locate the XML schema named in this submission archive request in the TOE's client-dependant configuration, an error message is returned to the CS and the

submission archive request is cancelled (FDP_IFC.1, FDP_IFF.1) and an audit record is generated by the TOE (FAU_GEN.1). If the data object to be archived doesn't contain at least an OID and retention time as meta information of the data object, the request is rejected and an error message is sent to the CS (FDP_ITC.1). [108]

If the TOE has successfully located the XML schema named in the submission archive request of the CS, the TOE uses this XML schema for syntactical validation of the SDO. If the validation of the SDO against the XML schema fails, an error message is returned to the CS, the submission archive request is cancelled and the unsuccessful verification of the XML schema is audited by the TOE (FAU_GEN.1).

If the validation of the SDO against the XML schema succeeds, the TOE proceeds with the process of the cryptographic verification of the digital signatures contained in the SDO if applicable using the defined external Crypto Provider component.

Therefore the SDO as an XML container is parsed with a set of XPath expressions [19]. If the XML container contains any digitally signed objects, the signatures and their corresponding documents are extracted from the SDO as the TOE user has assigned the TOE through the submission of the XML filter to be used for the archive request. Each extracted digital signature and its corresponding document will be cryptographically verified by the defined external Crypto Provider component. The TOE receives the overall verification result as well as an according XML verification report from the defined Crypto Provider. If any digitally signed object cannot be successfully be verified, the TOE cancels the submission request, an error message is returned to the CS and the unsuccessful verification of the digitally signed object is audited by the TOE (FAU_GEN.1).

---

108 As required by FDP_ITC.1.3, the meta information of each data object to be archived by the TOE will contain at least an object ID and a retention time. Note, that it is possible to define a default value for the retention time in the used XML filter, so that the CS has not to define a value for the retention time in each submission archive request.

So, if the CS does not declare a value for the retention time of the data object to be archived and there is a default value for the retention time defined in the XML filter associated with the submission archive request, the default value for the retention time in the XML filter will be used. If the CS declares a value for the retention time of a data object in his submission archive request, the value of the request will be used. If there is neither an individual value for the retention time defined in the submission archive request, nor a value defined in the XML filter associated with the request, the request will be declined.

For the communication with the defined external Crypto Provider, the TOE uses a socket-based communication channel according to SOAP v.1.1 [25] being logically distinct from other communication channels and providing assured identification of its end points and protection of the channel data from modification or disclosure. This is accomplished on the one hand through a programmatical validation of the SHA-256 value of each delivered Integrator-specific JAR archive and on the other hand through a signed-challenge authentication mechanism. Hereby, the TOE sends a generated unique token to the external Crypto Provider which returns this token signed with a secrete key whose public key is known to the TOE. The description of the used mechanisms and how the TOE meets the requirement FTP_ITC.1 (CRYPTO) is done in [23, chapter 3.2.2]. After being successfully authenticated for performing all further types of cryptographic operations, the TOE initiates the communication via the mentioned trusted channel to the defined Crypto Provider.

After having successfully checked the archive request and having successfully verified all digitally signed objects being part of the SDO, the TOE applies the rules being specified by the organization using the TOE according to the context of the respective archive request in the defined order (FDP_IFC.1, FDP_ITC.1, and FDP_IFF.1).

6.3 SF 3: Secure Storage Unit Access

If the request for submission of an SDO to the archive is successfully authenticated by the TOE and the data objects are successfully checked and verified by the TOE, the TOE immediately passes the data objects to be archived to the SU (or a trusted application which in turn passes the data objects to the dedicated SU) (FDP_IFC.1, FDP_IFF.1).

To ensure that the data objects are correctly submitted to the SU (or a trusted application which in turn submits the data objects to the dedicated SU) the TOE uses a communication channel to a trusted application which in turn submits the data objects to the dedicated SU being logically distinct from other communication channels and providing assured identification of its end points and protection of the channel data from modification or disclosure. The mutual authentication

mechanisms used for providing assured end point identification are analogue to the mechanisms used for the mutual authentication of the CS: the authentication of the TOE by the storage plugin has to been done by the TOE integrator through a programmatical validation of the SHA-256 value of each delivered Integrator-specific JAR archive. The authentication of the storage plugin by the TOE is done through a signed-challenge authentication mechanism between the TOE and the storage plugin. The detailed description of the used mechanisms and how the TOE technically meets the requirement FTP_ITC.1 (STORAGE) is done in [23, chapter 3.2.2]. For the submission of an SDO to the archive the TOE will initiate the communication via the mentioned trusted channel.
The dedicated long-term storage unit receives the submitted SDO from the TOE for saving and must send back a unique archive object identifier (AOID) to the TOE in case of the successful storage of the ADO. The TOE returns the OID and the AOID without interpretations to the submitting CS (FDP_IFC.1, FDP_IFF.1). Nobody is allowed to modify or delete those security attributes (FMT_MSA.1 (FLOW), FMT_MSA.3 (FLOW)). The SDO is now called archive data object (ADO). If the storage fails, the TOE cancels the submission request and returns an error message to the CS.

If a request for retrieval of an ADO or for retrieval of a the meta data of an ADO is sent to the TOE by an CS and the CS is successfully authenticated according to its session ID, the retrieval archive request must include the OID of the CS and the AOID of the ADO requested proofing the ownership of the requesting CS (FDP_IFC.1, FDP_IFF.1, FDP_ITC.2 (CSID), FMT_MSA.3 (ACCESS), and FMT_SMR.1).

6.4 SF 4: Invalid Archive Data Object Erasure Prevention

The TOE prevents the erasure of ADOs by any other CS than the CS which has submitted this ADO and the erasure of ADOs before expiry of their retention time without a justification (FDP_IFC.1, FDP_IFF.1) through the following measures.

For the erasure of an archive object from the long-term storage unit the CS has to submit an archive request for deletion of an ADO along with the authentication data which should prove that the CS

submitting the request is the CS which has initially submitted the ADO. The TOE ensures that these authentication data - together with the AOID of the archive object requested for deletion - are submitted to the long-term storage unit (or a trusted application which in turn submits the request to the dedicated SU) without modification (FDP_ACC.1, FDP_ACF.1, FMT_MSA.3 (ACCESS), and FMT_SMR.1).

If a CS submits an archive request for deletion of an ADO to the TOE, this erasure archive request must include the AOID of the ADO the CS wants to delete, the CS authentication data and a justification if the ADO shall be deleted before expiration of the ADO's retention time.

The TOE requests the retention time for the erasure archive request by passing the AOID to the SU (or a trusted application which in turn returns the retention time to the TOE). The ADO's expiration time is the time when the ADO's retention time is reached beginning with the time of the ADO's successful submission to the SU.

The TOE enforces the following rules:

- If the expiration time is behind the current time, the TOE sends the request for deletion to the SU (or a trusted application which in turn sends the request to the SU).
- If the expiration time is before the current time, the TOE checks, if the erasure archive request includes a justification. If the erasure archive request includes a justification, the TOE initiates the deletion of the ADO through the SU (or a trusted application which in turn forwards the initiation for deletion to the dedicated SU).
- If the expiration time is before the current time and the deletion archive request doesn't include a justification, the TOE cancels the deletion archive request and an error message is returned to the CS.

For all erasure archive requests for ADOs whose expiration time is before the current time the TOE generates an audit record, regardless of the outcome of the archive request (FAU_GEN.1).

6.5 TSS Rationale

The following table shows the correspondence analysis for the described TOE security functionalities and the security functional requirements.

SFR	**SF 1**: Secure Client TOE Access	**SF 2**: Data Object Verification	**SF 3**: Secure Storage Unit Access	**SF 4**: Invalid ADO Erasure Prevention
FAU_GEN.1	X	X		X
FDP_ACC.1	X			X
FDP_ACF.1	X			X
FDP_DAU.1	X			
FDP_ETC.2	X			
FDP_IFC.1	X	X	X	X
FDP_IFF.1	X	X	X	X
FDP_ITC.1		X		
FDP_ITC.2 (AREQ)	X			
FDP_ITC.2 (CSID)	X		X	
FDP_ITC.2 (SCHEMA)		X		
FIA_UAU.2	X			
FIA_UID.2	X			
FMT_MSA.1 (FLOW)	X		X	
FMT_MSA.3 (ACCESS)	X		X	X
FMT_MSA.3 (FLOW)	X		X	
FMT_SMR.1	X		X	X
FPT_TDC.1		X		
FTP_ITC.1 (CRYPTO)		X		

SFR	**SF 1**: Secure Client TOE Access	**SF 2**: Data Object Verification	**SF 3**: Secure Storage Unit Access	**SF 4**: Invalid ADO Erasure Prevention
FTP_ITC.1 (CS)	X			
FTP_ITC.1 (STORAGE)			X	

Table 6: Rationale for the SFR and the TOE Security Functionalities

7 Acronyms

ADO	Archive Data Object
AOID	Archive Object Identifier
CC	Common Criteria for IT Security Evaluation
CS	Client Software Application
EAL	Evaluation Assurance Level
IT	Information Technology
OID	Object Identifier
OSP	Organisational Security Policies
PP	Protection Profile
SDO	Submission Data Object
SFP	Security Function Policy
ST	Security Target
SU	(Long-Term) Storage Unit
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSP	TOE Security Policy

8 Literature

[1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; CCMB-2009-07-001, Version 3.1, Revision 3, July 2009

[2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components; CCMB-2009-07-002, Version 3.1, Revision 3, July 2009

[3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; CCMB-2009-07-003, Version 3.1, Revision 3, July 2009

[4] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology; CCMB-2009-07-004, Version 3.1, Revision 3, July 2009

[5] Common Criteria Protection Profile for an ArchiSafe Compliant Middleware for Enabling the Long-Term Preservation of Electronic Documents (ACM_PP), BSI-CC-PP-0049, 31.10.2008, Dr. Wolf Zimmer, Bundesamt für Sicherheit in der Informationstechnik

[6] W3C, XML Schema Part 0: Primer Second Edition, W3C Recommendation, 28 October 2004, http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/

[7] Evidence Record Syntax (ERS), RFC 4998, T. Gondrom, R. Brandner, U. Pordesch: http://www.ietf.org/rfc/rfc4998.txt

[8] ISO 19005-1, Document management – Electronic document file format for long-term preservation – Part 1: Use of PDF 1.4 (PDF/A-1), 2005

[9] TIFF Revision 6.0 Final, June 1993, Adobe Systems Incorporated, http://partners.adobe.com/asn/developer/pdfs/tn/TIFF6.pdf

[10] Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies, RFC 2045, N. Freed & N. Borenstein, November 1996, http://www.ietf.org/rfc/rfc2045.txt

[11] Gesetz über Rahmenbedingungen für elektronische Signaturen (Signaturgesetz - SigG) vom 16. Mai 2001 (BGBl. I S. 876), zuletzt geändert durch Art. 4 G v. 17.7.2009 I 2091

[12] Verordnung zur elektronischen Signatur (Signaturverordnung - SigV) vom 16. November 2001 (BGBl. I S. 3074), zuletzt geändert durch Verordnung vom 15.11.2010 (BGBl I S. 1542)

[13] Bekanntmachung zur elektronischen Signatur nach dem Signaturgesetz und der Signaturverordnung (Übersicht über geeignete Algorithmen), 30. Dezember 2011, Veröffentlicht am 18. Januar 2012 im Bundesanzeiger Nr. 10, Seite 243

[14] Federal Information Processing Standards Publication FIPS PUB 180-2 Secure Hash Standard (+ Change Notice to include SHA-224), U.S. Department of Commerce / National Institute of Standards and Technology, 2002 August 1

[15] PKCS #1 v2.1: RSA Cryptography Standard, RSA Laboratories, 14. June 2002

[16] RIPEMD-160: A Strengthened Version of RIPEMD, published by German Information Security Agency and Katholieke Universiteit Leuven, announced 18 April 1996

[17] A Universally Unique IDentifier (UUID) URN Namespace, RFC 4122, P. Leach, M. Mealling & R. Salz, July 2005, http://www.ietf.org/rfc/rfc4122.txt

[18] RFC 5652, Cryptographic Message Syntax (CMS), R. Housley, Vigil Security, September 2009, http://www.ietf.org/rfc/rfc5652.txt

[19] XML Path Language (XPath) 2.0, W3C Recommendation 23rd January 2007, http://www.w3.org/TR/2007/REC-xpath20-20070123/

[20] C. Adams, P. Cain, D. Pinkas, R. Zuccherato: Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP). IETF RFC 3161, http://www.ietf.org/rfc/rfc3161.txt

[21] BSI Technische Richtlinie 03125: Vertrauenswürdige elektronische Langzeitspeicherung, Version 1.0, 31.07.2009, Bundesamt für Sicherheit in der Informationstechnik, https://www.bsi.bund.de/ContentBSI/Publikationen/TechnischeRichtlinien/tr03125/index_htm.html

[22] Einheitliche Spezifizierung der Einsatzbedingungen für Signaturanwendungskomponenten, Version 1.4, announced 19.07.2005, http://www.bundesnetzagentur.de/cae/servlet/contentblob/37092/publicationFile/2443/SpezifiziergEinsatzBedinggnId2648pdf.pdf

[23] {Confidential Document:} Ergänzende Dokumentation, Dokument 1 – Authentisierung und Kryptographie, SecDocs Security Komponenten Version 1.0 (BSI-DSZ-CC-0685), Version 1.7, 03.07.2012, OpenLimit SignCubes GmbH

[24] {Confidential Document:} Ergänzende Dokumentation, Dokument 2 - Realisierung des Zufallszahlengenerators, SecDocs Security Komponenten 1.0 (BSI-DSZ-CC-0685), Dokumentversion 0.4, 03.07.2012, OpenLimit SignCubes GmbH

[25] Simple Object Access Protocol (SOAP) 1.1, W3C Note, 08.05.2000, http://www.w3.org/TR/2000/NOTE-SOAP-20000508

[26] Handbuch für Integratoren, SecDocs Security Komponenten Version 1.0, Dokumentenversion 2.1, 09.07.2012, OpenLimit SignCubes AG

[27] Handbuch für Administratoren, SecDocs Security Komponenten Version 1.0, Dokumentenversion 1.9, 09.07.2012, OpenLimit SignCubes AG