Common Criteria Protection Profile

for an

ArchiSafe Compliant Middleware

for Enabling the Long-Term Preservation

of Electronic Documents

(ACM_PP)

Version	Date	Author	Remarks
1.0.0	31/10/08	Dr. Wolf Zimmer	Final version

0 Table of Contents

Figures

Tables

1 PP Introduction

This document represents a Protection Profile (PP) for products enabling the 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]

1.1 PP Reference

PP Name:	Protection Profile for an ArchiSafe Compliant Middleware for Enabling the Long-Term Preservation of Electronic Documents (ACM_PP)
Certification ID:	BSI-CC-PP-0049
PP Version:	1.0.0
Date:	31/10/2008
Applicant:	Bundesamt für Sicherheit in der Informationstechnik (German Federal Office for Information Security)
Authors:	Dr. Wolf Zimmer, Computer Science Corporation, Wiesbaden Tobias Schäfer, Physikalisch-Technische Bundesanstalt, Braunschweig
Keywords:	ArchiSafe, Protection Profile, ACM_PP
CC Version:	3.1

1.2 TOE Overview

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 regulated retention times. The ArchiSafe approach [5] to long-term preservation of electronic documents claims:

- To use 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] Please consider the requirements of section 1.2.3 of this PP.

- To reduce the dependencies from obsolescent IT infrastructure and storage technology by a straight service-oriented, multi-tier and client capable architecture.

The TOE specified in this PP 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.

1.2.1 Usage and major security features of the TOE

The target of evaluation (TOE) is a product or part of a product providing 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.

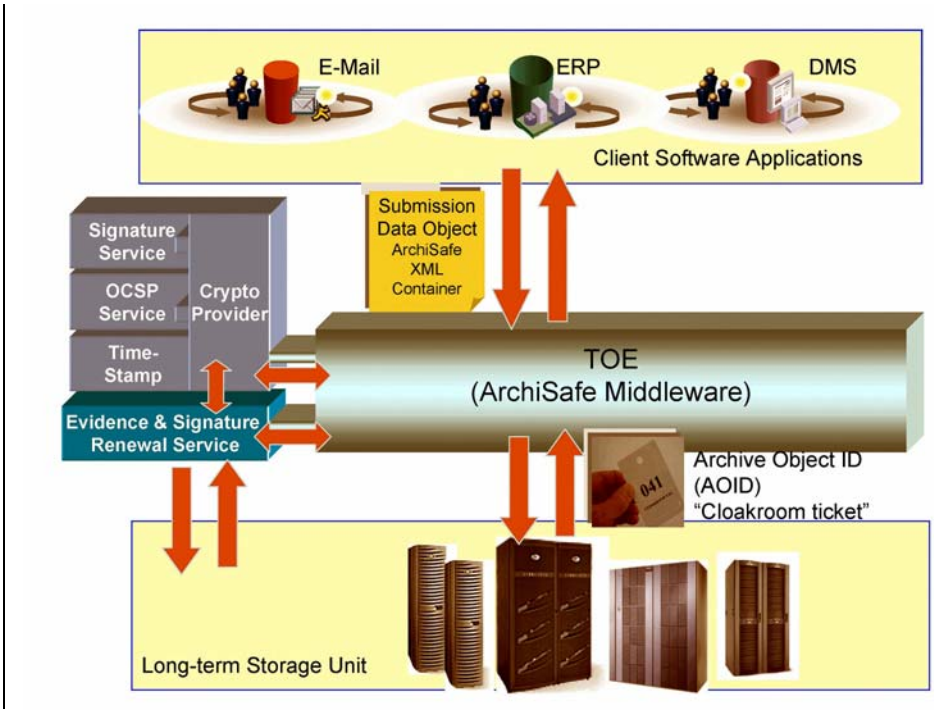

Figure 1: Architectural Overview

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 interfaces an external crypto provider as shown in Figure 1.
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 CS and the SU, the TOE provides the following general security functionalities:

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

(SS 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,

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

(SS 4) preventing the erasure of ADO's by any other CS than the CS which has also submitted this ADO and preventing the erasure of ADO's 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 also not protect the confidentiality of the documents.

[2] The denomination follows the OAIS framework for sharing archival notions. 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), s. also: http://www.personal.leeds.ac.uk/~ecldh/cedars/ieee00.html

1.2.2 TOE Type

The TOE is an IT middleware component or part of an IT middleware component that trustworthy and reliable mediates and controls the access to a SU for submission or retrieval of SDO's and ADO's.

1.2.3 Required non-TOE hardware/software/firmware

The TOE runs as an application on an IT system and needs the protection by the underlying system platform, e.g. the operating system.

The CS and the SU (or another trustworthy applications interfacing with the TOE and the SU) are not part of the TOE. The TOE depends on some features of these parties, e.g. the generation of the unique archive object ID by the SU or another non-TOE part of the archive middleware.

The TOE itself does not execute any cryptographic mechanisms for protecting or evaluating the integrity and authenticity of the data to be archived. For this purpose the TOE uses trustworthy crypto providers which are explicitly not part of the TOE. Crypto providers may be implemented in hardware, software or firmware.

The TOE itself also does not provide any functionality and/or mechanisms to preserve and to renew the non-repudiation of the archived data. For this purpose the TOE uses trustworthy components which are explicitly not part of the TOE. These components may be implemented in hardware, software or firmware and interfaced by the crypto providers, the long-term storage unit(s) and/or the archive middleware.

2 Conformance Claim

2.1 CC Conformance Claim

The Protection Profile is based upon

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; Version 3.1, Revision 1, CCMB-2006-09-001, [1]
- Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components; Version 3.1, Revision 2, CCMB-2007-09-002, [2]
- Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components; Version 3.1, Revision 2, CCMB-2007-09-003, [3]

referenced hereafter as [CC].

This Protection Profile claims the following CC conformance:

- part 2 conformant
- part 3 conformant
- evaluation assurance level (EAL) 3

2.2 Conformance Statement

Security targets or other PPs wishing to claim conformance to this PP can do so as *Strict PP conformance*. *Demonstrable PP conformance* is not allowed for this PP.

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/company.

Administrator (Admin)

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.

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[3] or TIFF[4], which has to be converted into a native text format (MIME Base64[5] coded) for embedding it in the XML based data object.

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

[4] TIFF Revision 6.0, Final – June 3, 1992, at http://partners.adobe.com/asn/developer/pdfs/tn/TIFF6.pdf

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.

Submission Data Objects (SDO's)

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 signature and/or time-stamp blocks are also included (see Figure 2).

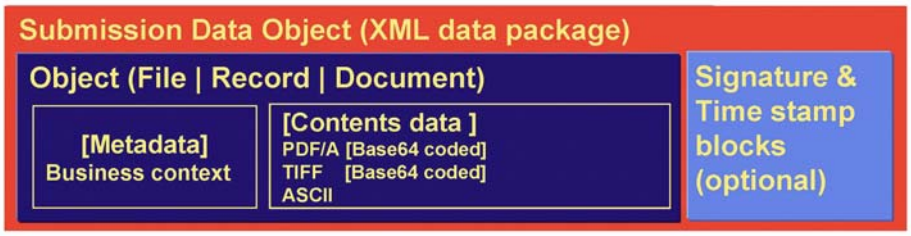

Figure 2: Submission Data Object

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 object ID, an XML based description of the documents business context or the documents retention time, is contained in the metadata block of the XML structure.

[5] Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies, section 6.8, Base64 Content-Transfer-Encoding, IETF RFC 2045, N. Freed & N. Borenstein, November 1996, at http://www.ietf.org/rfc/rfc2045.txt?number=2045

Archive Data Objects (ADO's)

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

XML Schema

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

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.

Application Note: The rules may specify that the TOE

- *Must digital timestamp any submission data object. For this purpose, the TOE shall use the external crypto provider.*
- *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 may interface to an external crypto provider or to another special application.*

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, an evidence record includes archive timestamps, and additional verification data, like certificates, revocation information, trust anchors, policy details, role information, etc.[6] Evidence data will be generated,

[6] T. Gondrom, R. et al. , Evidence Record Syntax (ERS), IETF Network Working Group, Aug. 2007 at http:// www.ietf.org/rfc/rfc4998.txt

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, the Client Software Application wants to store a (new) submission data object 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 / modified.
- **Retrieval** means, the Client Software Application wants to read out a particular archive data object from the storage. Modification/Update of this archive data object in the storage is not possible.
- **Erasure** means, 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.
- **Request for evidence** means, 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, the Client Software Application wants to read out some meta information of one, some or all archive data objects stored

[7] T. Gondrom, R. et al. , Evidence Record Syntax (ERS), IETF Network Working Group, Aug. 2007 at http:// www.ietf.org/rfc/rfc4998.txt

	in the storage. These meta information may contain search indices, ownership of archive data objects, retention times, digital signatures, etc.
Authentication of a 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/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	*Application Note: This PP does not want to specify how to identify/authenticate archive requests in detail. Product developers shall be free to use their own procedures.* *The following definition is just one possible solution.* An archive request can be authenticated by verification of the (optional) digital signature of this request. The authentication fails if the signature is wrong/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 client software application or if the signature does not exist.
Checks / Verification of submission data objects	Technically spoken, the submission data object is a 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.[8]

[8] see http://www.w3.org/TR/xmlschema-0/

Verification of an archive request	*Application Note: This PP does not want to specify how to identify/authenticate archive requests in detail. Product developers shall be free to use their own procedures.* *The following definition is just one possible solution.* Technically spoken, the archive request is a 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.[9]
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”
Read 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.

[9] see http://www.w3.org/TR/xmlschema-0/

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 by the TOE.
Submitter of a submission data object	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	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	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 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 secure configured and all data required for the configuration of the TOE are secure and reliable transported to and installed on the machine which runs the TOE.

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 runs. All underlying systems are secure installed and protected against unauthorized physical and logical access and modification. The machine on which the TOE runs is free from malware and viruses.

A.STORAGE	The dedicated SU provides a reliable, secure and available storage of data, even for long-terms. *Application Note: Logically or physically separated parts or components of the dedicated SU provides 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.*[10] *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.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. *Application Note: These special applications may be realized as part or aside the cryptographic components, the SU or the non-TOE parts of the middleware.*
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 are secure, and provide reliable measures regarding the authentication and access authorization of users.

[10] T. Gondrom, R. et al. , Evidence Record Syntax (ERS), IETF Network Working Group, Aug. 2007 at http:// www.ietf.org/rfc/rfc4998.txt

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. *Application Note: 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 protection profile*
A.XMLSCHEMA	For any CS using the TOE for submitting SDO's into the SU must exist a valid data schema (XML schema). Schema instructions and rules defined for using the schema do not introduce any security risk.

3.3 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 AOID's.
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 the retention time of the archive object 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 (see SS 3 on page 7) 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.

3.4 Organizational Security Policies

P.ACCESS The TOE only allows the following operations:

- Submit submission data objects to the storage,
- Retrieve archive data objects from the storage,
- 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 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 to the storage,
- Retrieve archive data objects from the storage,
- Delete archive data objects from the storage,
- Request for evidence of an archive data object and
- Reading of 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.

O.CRYPTOPROV The TOE assures that the selected (defined) trustworthy cryptographic component can not 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 archived 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 a 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 archive data objects.
O.TOE_ACCESS	The TOE does not grant any access to TSF (TOE Security Functions) data. *Application Note: Configuration data may be 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 secure and reliable transported to and installed on the machine which runs the TOE.

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 runs 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 has to provide a reliable, secure and available storage of data, even for long-terms.

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[11].
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's must exist a valid data schema (XML schema). Schema instructions and rules defined for using the schema must not introduce any security risk.

[11] T. Gondrom, R. et al. , Evidence Record Syntax (ERS), IETF Network Working Group, Aug. 2007 at http:// www.ietf.org/rfc/rfc4998.txt

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.

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 client applications 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 secure configured and all data required for the configuration of the TOE are secure and reliable transported and installed on the machine on which the TOE runs. The security objective OE.CONFIGURATION 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 secure 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.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[12]. The security objective OE.EVIDENCEDATA for the operational environment covers this assumption.

A.TIMESTAMP: A.TIMESTAMP assumes that the TOE is provided reliable time-stamps. 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.XMLSCHENMA: 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.

[12] T. Gondrom, R. et al. , Evidence Record Syntax (ERS), IETF Network Working Group, Aug. 2007 at http:// www.ietf.org/rfc/rfc4998.txt

T.DATA_ACCESS1: This threat focuses on any attempts to gain unauthorized access to the archive, e. g. by sending manipulated AOID's. 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 focuses on attempts to modify in a specific manner a data package during the transmission between 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 the TOE allows only the following operations:

- Submit data objects to the storage,
- Retrieve data objects from the storage,
- Delete data objects within the storage,
- Request for evidence and

- Reading of 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 assigns to any XML data package to be archived 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 archive 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.

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 that shall be fulfilled by a product that is conformant to this protection profile.

- 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.
- Operations, which are not executed, are reproduced from the [CC] without any changes.
- Uncompleted Operations are still written in brackets containing at first the executed part of the operation and subsequently the specification of the operation to be performed.

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

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 [selection, choose one of: *minimum, basic, detailed, not specified*] 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**
- [assignment: *other specifically defined auditable events*].

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 PP/ST, **for successful erasure archive requests for archive objects whose retention time is not yet expired, the justification**, [assignment: *other audit relevant information*].

Application Note: The uncompleted operations enable a product developer to specify its own level of auditing and some more audit events and audit 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** 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

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** to objects based on the following: **a) list of subjects: Client Software Applications** o **Security Attribute: Client Software Application Identity** **b) objects: Archive Object** o **Security Attribute: Owner**
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**
FDP_ACF.1.3	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*].
FDP_ACF.1.4	The TSF shall explicitly deny access of subjects to objects based on the [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*].

Application Note: The operations of the last two elements of this component are not finally detailed for enabling a product developer to implement some more access control rules. Examples for such additional rules may be "Explicit access authorization for the company's data protection officer". Generally, the rules already specified should not be bypassed.

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**.
FDP_DAU.1.2	The TSF shall provide **the Client Software Applications** 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** when exporting data objects, controlled under the SFP(s), to the long-term storage unit or another trustworthy application.
FDP_ETC.2.2	The TSF shall export the data object with the data object's 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, are unambiguously associated with the exported data object.
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: **The data object shall be augmented with the ID of the submitting Client Software Application**.

FDP_IFC.1	**Subset information flow control**
Hierarchical to:	No other components.
Dependencies:	FDP_IFF.1 Simple security attributes
FDP_IFC.1.1	The TSF shall enforce the **TSP_IFC** 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** [assignment: *list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP*].

Application Note: The uncompleted operation gives a product developer the ability to control some more information flows.

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

FDP_IFF.1.1 The TSF shall enforce the **TSP_IFC** based on the following types of subject and information security attributes:

- **Subject: Client Software Applications,**
  - **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**
  - **Security Attributes: Long-term storage unit identity or application identity**
- **Information: Data objects**
  - **Security Attributes: Submitter of the data object**
- **Information: Archive Objects**
  - **Security Attributes: Owner of the archive object**

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

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**

[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**

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

FDP_IFF.1.4 The TSF shall explicitly authorise an information flow based on the following rules: [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**

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

Application Note: The uncompleted operations give a product developer the ability to specify some more information flow rules. These additional rules should not bypass the rules already specified.

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** when importing data objects, controlled under the SFP, from submitting Client Software Applications.
FDP_ITC.1.2	The TSF shall ignore any security attributes associated with the data objects when imported from submitting Client Software Applications.
FDP_ITC.1.3	The TSF shall enforce the following rules when importing data objects controlled under the SFP from submitting Client Software Applications: • **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.**

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** when importing archive requests, controlled under the SFP, from submitting Client Software Applications.
FDP_ITC.2.2	The TSF shall use the security attributes associated with the imported archive requests.
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 received.
FDP_ITC.2.4	The TSF shall ensure that interpretation of the security attributes of the imported archive requests is as intended by the source of the archive requests.
FDP_ITC.2.5	The TSF shall enforce the following rules when importing archive requests controlled under the SFP from submitting Client Software Applications: • **The imported security attributes of an archive request shall proof the integrity and authenticity of the archive request**

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
FDP_ITC.2.1	The TSF shall enforce the **TSP_IFC** when importing archive objects, controlled under the SFP, from the long-term storage unit or another trusted application which in turn interfaces with the long-term storage unit.
FDP_ITC.2.2	The TSF shall use the security attributes associated with the imported archive objects
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 received.
FDP_ITC.2.4	The TSF shall ensure that interpretation of the security attributes of the imported archive objects is as intended by the source of the archive objects.
FDP_ITC.2.5	The TSF shall enforce the following rules when importing archive objects controlled under the SFP from the long-term storage unit or another trusted application which in turn interfaces with the long-term storage unit: • **The imported security attributes shall identify the owner of the archive object.**

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** when importing XML schemas, 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.
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 received.
FDP_ITC.2.4	The TSF shall ensure that interpretation of the security attributes of the imported XML schemas is as intended by the source of the XML schemas.
FDP_ITC.2.5	The TSF shall enforce the following rules when importing XML schemas controlled under the SFP from outside the TOE: • **The imported security attributes of a XML schema shall proof the integrity and authenticity of the XML schema** • **The imported security attributes of a XML schema shall identify the issuing organization**

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 to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that Client Software Application.

FIA_UID.2	User identification before any action
Hierarchical to:	FIA_UID.1 Timing of identification
Dependencies:	No dependencies.

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

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** to restrict the ability to *modify* or *delete* 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** to **nobody**.

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

FMT_MSA.3.1	The TSF shall enforce the **TSP_ACC** to provide *restrictive* default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2	The TSF shall allow **nobody** 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** to provide *restrictive* default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2	The TSF shall allow **nobody** 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.

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,** **[**assignment: *the authorised identified roles*].
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,** [assignment: *list of TSF data types*] when shared between the TSF and the underlying system.
FPT_TDC.1.2	The TSF shall use [assignment: *list of interpretation rules to be applied by the TSF*] when interpreting the TSF data from the underlying system.

Application Note: This SFR ensures that the TOE can read and interpret the configuration data which contains the XML schemas and the organization specific rules.

The operation for the interpretation rules was not detailed because the interpretation of these configuration data may follow different rules in different products.

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 a trusted crypto provider 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* 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**.

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 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 *Client Software Application* to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for **nothing**.

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 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* 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**

5.3 Security Assurance Requirements

The following Table 1 gives an overview on the security assurance requirements that have to be fulfilled by the TOE. They correspond to the assurance level EAL3 of part 3 of the Common Criteria.

Table 1: TOE security assurance requirements

Assurance class	Assurance components
ADV: Development	ADV_ARC.1 Security architecture description
	ADV_FSP.3 Functional specification with complete summary
	ADV_TDS.2 Architectural design
AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
ALC: Live-cycle support	ALC_CMC.3 Authorisation controls
	ALC_CMS.3 Implementation representation CM coverage
	ALC_DEL.1 Delivery procedures
	ALC_DVS.1 Identification of security measures
	ALC_LCD.1 Developer defined life-cycle model
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

Assurance class	Assurance components
	ATE_DPT.1 Testing: basic design
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.2 Vulnerability analysis

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 Protection Profile.

Table 2: Coverage of the security objectives by security functional requirements

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)

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”) can not 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 a dedicated crypto provider. Thus, the selected (defined) trustworthy cryptographic component can not 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 a 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), FTP_ITC.1 (STORAGE) guarantees that the TOE is capable to authenticate itself reliably against external entities.

5.5 Rationale For Assurance Requirements

EAL3 as minimum level for PP compliant products was chosen because the intention of these systems is to provide a trustworthy access point to a digital and long-term archive.

The definitions of the EALs 1 and 2 state that they are only applicable when a low to medium level of independently assured security is required. Here, a trustworthy long-term archive access point requires a higher level of security.

5.6 Rationale for all not-satisfying Dependencies

With the exception of the security functional components FAU_GEN.1, FMT_MSA.1 (FLOW) and FMT_MSA.3 (ACCESS) all dependencies are contained in this Protection Profile.

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
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. *Application note: It may be possible that a specific product contains management functions. Then, the respective ST shall resolve this dependency.*
	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. *Application note: It may be possible that a specific product contains management functions. Then, the respective ST shall resolve this dependency.*
	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	---

6 Acronyms

CC	Common Criteria for IT Security Evaluation
EAL	Evaluation Assurance Level
IT	Information Technology
OSP	Organisational Security Policies
PP	Protection Profile
SFP	Security Function Policy
ST	Security Target
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSP	TOE Security Policy

7 References

[1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; Version 3.1, Revision 1, CCMB-2006-09-001,

[2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components; Version 3.1, Revision 2, CCMB-2007-09-002,

[3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components; Version 3.1, Revision 2, CCMB-2007-09-003,

[4] VLA – Vertrauenswürdige elektronische Langzeitarchivierung, BSI Technische Richtlinie, BSI-TR 03125

[5] ArchiSafe: http://www.archisafe.de

[1-3] are referenced together as [CC]