Non-official translation

DIRECTOR GENERAL OF THE
LITHUANIAN ARCHIVES DEPARTMENT
UNDER THE GOVERNMENT OF THE REPUBLIC OF LITHUANIA

ORDER
ON THE CONFIRMATION OF THE SPECIFICATION ADOC-V1.0 OF THE ELECTRONIC DOCUMENT SIGNED BY THE ELECTRONIC SIGNATURE

September 7, 2009, No. V-60
Vilnius

In accordance with items 6.1 and 10.6 of the regulations of the Lithuanian Archives Department under the Government of the Republic of Lithuania, approved by the resolution of February 8, 2002 of the Government of the Republic of Lithuania No. 197 (Official Gazette *Valstybės Žinios*, 2002, No. 15-590; 2005, No. 12-368),

I hereby approve the specification ADOC-V1.0 (attached) of the electronic document signed by the electronic signature.

Director General Vidas Grigoraitis

APPROVED
by the Director General of the Lithuanian Archives Department under the Government of the Republic of Lithuania by the order No. V-60 of September 7, 2009

SPECIFICATION ADOC-V1.0 OF THE ELECTRONIC DOCUMENT SIGNED BY THE ELECTRONIC SIGNATURE

I. GENERAL PROVISIONS

Purpose and Scope of the Specification

1. Specification of the electronic document signed by the electronic signature (hereinafter referred to as Specification) determines the requirements for the electronically signed official electronic documents created by the state and local institutions, bodies and enterprises, other entities authorized to perform public administrative functions (hereinafter referred to as Bodies), individuals authorized by the state and received official electronic documents sent by the non-state organizations, private legal entities and natural persons (of GeDOC and GGeDOC categories), and requirements for the software managing the lifecycle of official electronic documents.

Requirements determined by the Specification applicable for the non-state organizations, private legal entities and natural persons (hereinafter referred to as Entities) are of the recommendation character except for the cases when the prepared electronic documents are to be submitted to the bodies or should be prepared in accordance with the requirements of the legislation.

2. Specification is prepared on the basis of the Set of requirements for the specification of the electronic document signed by the electronic signature, approved by the order No. V-119 of October 9, 2008 issued by the Director General of the Lithuanian Archives Department under the Government of the Republic of Lithuania (Official Gazette *Valstybės Žinios*, 2008, No. 118-4488).

3. Specification determines:

3.1. requirements for the logical and physical structure of the electronic document;

3.2. requirements for the structure of the content of the electronic document and file formats of the content;

3.3. forms of the electronic signature(s);

3.4. requirements for creation and verification of the electronic signature(s) and determination validity of the electronic certificate(s);

3.5. metadata and their properties, related with the preparation, registration, codification, access, storage and deletion procedures of electronic documents;

3.6. requirements for the verification of the electronic document.

4. Specification determines requirements for the electronic documents signed by the XAdES-EPES, XAdES-T, XAdES-C, XAdES-X, XAdES-X-L or XAdES-A forms of electronic signatures.

Interoperability guidelines

5. Specification is prepared on the basis of standards and recommendations for the open formats and electronic signatures (Appendix 18). Interoperability of the electronic documents signed by the electronic signature enables the transfer of the electronic document created in one system into any other system, where it can be verified for authenticity and suitability for the long-term preservation, if the software of these two systems corresponds to the requirements for the creation and testing of the electronic documents determined by the Specification.

6. On the basis of the appropriate standards the Specification defines the structure of the electronic signature, time-stamping and certificate validity checking services of the public keys infrastructure, and the structure of content files and the package of the electronic document signed by the electronic signature.

7. Definitions are used in the Specification:

Archive time stamp token – a nested time stamp token or a time stamp token provided for the XAdES-X-L form signature in order to construct the XAdES-A form signature.

BeDOC electronic documents – electronic documents, that are not GGeDOC electronic documents, created by the non-state organizations and private legal entities, including documents (for instance, contracts) created together with natural persons.

CeDOC electronic documents – electronic documents, that are not GGeDOC electronic documents, created by the natural persons.

Subsequent verification of the electronic signature – verification of the validity of the electronic signature based on the information stored in the electronic signature and performed without changing the form of the electronic signature.

Initial verification of the electronic signature – verification of the validity of the electronic signature and its preparation for the long-term preservation, providing the possibility of the verification of the validity of the electronic signature at a later stage irrespective of the public keys infrastructure by making the XAdES-X-L form of the electronic signature.

Initial verification of the electronic signature before the end of grace period – a stage of initial verification of the electronic signature initiated immediately after the receipt of the electronic document signed by the electronic signature and consisting of the verification of the format of the electronic signature and authenticity verification of the electronically signed electronic document, creation of the time stamp token, and primary verification of the validity of the qualified certificate.

Initial verification of the electronic signature after the end of grace period – a stage of initial verification of the electronic signature, performed after the end of grace period of the qualified certificate, calculated according to the time indicated by the time stamp token of the electronic signature and consisting of the secondary verification of the validity of the qualified certificate and making the XAdES-X-L electronic form signature in accordance with the LST CWA 14171:2005 standard (Appendix 18, item 3).

Valid qualified certificate – a qualified certificate within validity period which corresponds to all the following requirements:

- validity period is within the validity period indicated in the qualified certificate;

- certificate validity period is limited by the time of publishing the revocation or suspension status of the certificate within the grace period indicated by the policy of the qualified certificate.

GeDOC electronic documents – electronic documents prepared by the state and local institutions, bodies and enterprises, other entities authorized to perform public administrative functions, individuals authorized by the state, including documents (for instance, contracts), prepared together with the non-state organizations, private legal entities and natural persons.

GGeDOC electronic documents – electronic documents, sent from non-state organizations, private legal entities and natural persons to the state and local institutions, bodies and enterprises, other entities and individuals authorized to perform administrative functions.

IRI – an internationalized resource identifier used as the reference (address) for the resource access and composition of which is in conformity with the RFC 3987 (Appendix 18, item 19).

Explicit signature policy – a policy for the creation and verification of the electronic signature, possessing the registered unique identifier indicated in the electronic signature value <*SignaturePolicyIdentifier*>.

Directory – a service file used for storing information on other files and/or directories.

Qualified certificate – a certificate created by the certification service provider, meeting the requirements for the certification service providers creating qualified certificates, approved by the resolution No. 2108 of the Government of the Republic of Lithuania issued on

December 31, 2002 (Official Gazette *Valstybės Žinios*, 2003, No. 2-47), and registered observing the procedure approved by the registration order for the certification service providers creating qualified certificates approved by the resolution No. 2108 of the Government of the Republic of Lithuania issued on December 31, 2002 (Official Gazette *Valstybės Žinios*, 2003, No. 2-47), or by the certification service provider of the European Union Member State, possessing the status of the qualified certification service provider endowed by the legislation of the European Union Member State. This certificate contains data indicated in the Law on Electronic Signature of the Republic of Lithuania (Official Gazette *Valstybės Žinios*, 2000, No. 61-1827).

Qualified electronic signature – an advanced electronic signature which is based on a qualified certificate and which is created by a secure-signature-creation device.

Time stamp token – a proof-of-existence for a datum at a particular point in time, in the form of a data structure signed by a time stamping authority.

Time stamp service – a service that provides a trusted association between a datum and a particular point in time, in order to establish reliable evidence indicating the time at which the datum existed.

Nested time stamp token – a time stamp token given for the XAdES-A form signature, extending the XAdES-A form signature, and stored in the signature element <*UnsignedSignatureProperties*>.

MIME type label – a sequence of the text characters used to identify the content type of the file or data in conformity with the requirements of the RFC 4288 (Appendix 18, item 20) (e.g. *text/xml*).

Implicit signature policy – a policy for the creation and verification of the electronic signature, identified in the electronic document signed by the electronic signature or approved by the legislation regulating the usage of the electronic documents and indicated in the XAdES-EPES form electronic signature by the element <*SignaturePolicyIdentifier*> containing the element <*SignaturePolicyImplied*> inside.

Trusted time stamp service – a time stamp service in conformity with the requirements of the LST ETSI TS 102 023 v1.2.1:2007 standard (Appendix 18, item 7) or the one that has a trust by the verifier of the electronic signature.

Trusted certification authority – a certification service provider who issues qualified certificates or the one that has a trust by the verifier of the electronic signature.

Secure signature creation system – a system in conformity with the requirements of the LST CWA 14170:2005 standard (Appendix 18, item 2), including the requirements for the secure signature-creation devices (LST CWA 14169:2009 standard (Appendix 18, item 1)).

Secondary verification of the validity of the certificate – verification of the validity of the qualified certificate confirming the signature of the electronic document and the verification of the validity of the certification path, performed after the end of grace period of the qualified certificate, according to the time indicated by the time stamp token in the signature.

Primary verification of the validity of the certificate – verification of the validity of the qualified certificate confirming the signature of the electronic document and the verification of the validity of the certification path, performed immediately after the reception of the document signed by the electronic signature and after the time stamp token embedding into the signature received.

Grace period of the certificate – a time period during which the certificate revocation information is propagated through the revocation process to relying parties; it is the minimum time period an initial verifier has to wait to allow any authorized entity to request a certificate revocation and the relevant revocation status provider to publish revocation status.

Certification path – a chain of multiple certificates, comprising a certificate of the public key owner (the end entity) signed by one certification authority, and zero or more additional certificates of certification authorities signed by other certification authorities; it ends by the root certificate of the certification authority who creates the certificate for himself and signs it.

Root certificate – a self-signed certificate that can be verified by its public key.

URI – a uniform resource identifier used as a reference (address) for the resource access composition of which is in conformity with the RFC 3986 (Appendix 18, item 18).

Public keys infrastructure – a set of the organizational and technical measures enabling certification service providers to uniquely assign public keys of asymmetric cryptography with the individuals or entities of the electronic environment so that they can be identified in the electronic environment.

XAdES – the XML format-based electronic signature standard LST ETSI TS 101 903 V1.4.1:2009 “XML Advanced Electronic Signatures (XAdES)”.

XAdES-A – an archival form of the electronic signature created in conformity with the XAdES standard.

XAdES-BES – a basic electronic signature created in conformity with the XAdES standard.

XAdES-C – an electronic signature with complete validation data references created in conformity with the XAdES standard.

XAdES-EPES – an explicit policy based electronic signature created in conformity with the XAdES standard.

XAdES-T – an electronic signature with a time stamp token created in conformity with the XAdES standard.

XAdES-X – an extended electronic signature with verification references and a time stamp token created in conformity with the XAdES standard.

XAdES-X-L - an extended long-term electronic signature created in conformity with the XAdES standard.

XML – the descriptive language for generic data structures and their content recommended by the World Wide Web Consortium (W3C).

XMLDSIG – the XML-based format of the electronic signature recommended by the World Wide Web Consortium (W3C) (Appendix 18, item 22).

ZIP archive – a ZIP format file containing one or more ZIP elements intended for storing data and/or compressing the files included.

Type of ZIP element– the MIME type of ZIP element file.

Other definitions used in this Specification should be interpreted as they are defined by the Law on Documents and Archives of the Republic of Lithuania (Official Gazette *Valstybės Žinios*, 1995, No. 107-2389; 2004, No. 57-1982), by the Law on Electronic Signature of the Republic of Lithuania (Official Gazette *Valstybės Žinios*, 2000, No. 61-1827), by the Set of Requirements for the Specification of the Electronic Document Signed by the Electronic Signature, approved by the order No. V-119 (Official Gazette *Valstybės Žinios*, 2008, No. 118-4488) of October 9, 2008 issued by the Director General of the Lithuanian Archives Department under the Government of the Republic of Lithuania (hereinafter referred to as Description of the Specification Requirements), by the Rules for the Management of the Electronic Documents, approved by the order No. V-12 (Official Gazette *Valstybės Žinios*, 2006, No. 7-268) of January 11, 2006 issued by the Director General of the Lithuanian Archives Department under the Government of the Republic of Lithuania, and other regulatory legislation.

II. PACKAGE OF THE ELECTRONIC DOCUMENT

8. The package of electronic document is a file format in conformity with the requirements indicated in the section 17 of the standard LST ISO/IEC 26300:2007 “Information Technology – Open Document Format for Office Applications (OpenDocument) v1.0 (ISO/IEC 26300:2006, identical)“, and the requirements of this Specification (Appendix 3):

8.1. The package is a ZIP archive in conformity with the ZIP file specification (Appendix 18, item 23), and consisting of one or more files grouped into directories containing files and/or other directories.

8.2. The package should not be protected by passwords.

8.3. The package should contain *META-INF/manifest.xml* file providing the list of files and directories present in the package and MIME type labels for each file and directory;

8.4. Electronic signatures should be stored in the *META-INF* directory; file names of the electronic signature should contain the word *signatures*.

9. The package can contain:

10. Thumbnail of the electronic document that is added by creating a separate directory `Thumbnails` and including the file `thumbnail.png` in conformity with the requirements. Empty MIME type label should be set for this directory in the `manifest.xml` file.

10.1. Text file `mimetype` whenever the file extension of the electronic document is not sufficient for the identification of the document specification. The content of the file *mimetype* is the MIME type label of the electronic document (package) identifying the specification of the electronic document (Appendix 9). This file should be added to the package in conformity with the requirements indicated in the section 17.4 of the standard LST ISO/IEC 26300:2007 (Appendix 18, item 12).

11. The package is a ZIP archive, where files can be stored:

11.1. compressed with *DEFLATE* algorithm;

11.2. uncompressed (*STORED*).

12. ZIP format (Appendix 18, item 23) adds the following limits for the electronic document:

12.1. The size of the electronic document package file cannot exceed 4 GB;

12.2. The size of each file before compression stored in the package cannot exceed 4 GB;

12.3. The length of the file reference cannot exceed 65,535 bytes;

12.4. The total number of files and directories of the hierarchical structure stored in the electronic document package file cannot exceed 65,535.

Model of the electronic document

13. The following structures of the electronic document are distinguished:

13.1. The *logical* structure of the document describing separate parts (components) of the electronic document;

13.2. The *physical* structure of the document describing how the parts of the electronic document are represented in *files* and *directories*;

13.3. The package where the physical document structure is stored.

14. The Specification defines the self-describing model of the electronic document where the logical document structure is separated from its presentation on the package by means of the

physical structure. This model allows the manager of the electronic document to freely determine the physical structure of the document, i.e. give names for the files corresponding to separate parts of the electronic document, group the files into directories within the requirements of the Specification, and makes an obligation to document the physical structure in conformity with the requirements of the Specification.

15. The logical structure of electronic document contains the following parts (Appendix 1):

15.1. Single main document;

15.2. Signed and unsigned metadata;

15.3. At least one electronic signature;

15.4. Description of document parts;

15.5. Description of parts relationships.

16. Electronic document may also contain the following parts:

16.1. One or more document appendices;

16.2. One or more attached independent electronic documents.

Structure of the package

17. Parts of the logical structure of the electronic document are represented in the physical structure by files grouped into directories. As the package is not a constituent part of the electronic document, physical structure of the electronic document may be stored out of the package (for instance, in a computer directory structure, in a structure of database records, etc.). The Specification defines the physical structure of the electronic document that is stored in the package.

18. The description of document parts and description of parts relationships are represented in the physical structure as XML files stored in the *META-INF* directory with the fixed file names *manifest.xml* (file for the description of files and their content types) and *relations.xml* (file for description of relationships). This directory should also contain electronic signature files.

19. The physical structure for the other parts of the electronic document is defined by the author of the electronic document and should be in conformity with the requirements of the Specification. The documentation of the physical structure of the electronic document is made by documenting the relationships of the physical and logical structures (package relationships) (Appendix 4).

20. The following requirements are applicable to the physical structure of the package (Appendix 3):

20.1. The package should be a single file with the extension *.adoc* (lowercase).

20.2. The package should be in conformity with the ZIP format specification (Appendix 18, item 23).

20.3. Names of files and directories should be UTF-8 encoded.

20.4. The root directory of the package should contain only one file – the file of the main document.

20.5. The root directory of the package may contain one or more directories with freely defined names, not coinciding with the *META-INF* directory name, that can contain the hierarchical structure of the content of the electronic document defined in the items 48-49 of the Specification, files of document appendices and attached electronic documents. The structure of files of the document content should be documented in accordance with the items 24 and 41 (Appendix 15) of the Specification.

20.6. The root directory of the package should contain the directory for metadata files of the electronic document. Metadata directory should be indicated by the MIME type label in the manifest file as indicated in Appendix 9. Metadata files should be formed and documented in accordance with items 24, 41, 53-61 of this Specification.

20.7. The root directory of the package should contain *META-INF* directory containing the following files and directories:

20.7.1. Electronic signature files documented in accordance with the requirements of items 24, 41-42 of this Specification, optionally grouped into directories. Electronic signature directory (if any) should be indicated by the MIME type label in the manifest file. Electronic signatures should correspond to the requirements set in items 8.4, 62-70 of this Specification;

20.7.2. The file *manifest.xml* providing the description of files and directories in the package and their content types;

20.7.3. The file *relations.xml* providing the description of document relationships.

Manifest file

21. The manifest file is an unsignable XML file that lists all the files and directories of the electronic document and their MIME type labels. The package of the electronic document should contain only one manifest file with the fixed name *META-INF/manifest.xml*.

22. The structure of this XML file should be in accordance with the requirements indicated in section 17.7 (Appendix 15) of the standard LST ISO/IEC 26300:2007 (Appendix 18, item 12). The requirements set in the subsections 17.7.4-17.7.6 of this standard are not applicable as this Specification does not permit an encryption of the content of the electronic document.

23. The root element of the manifest is <*manifest*> and is made up of a single definite attribute indicating the element namespace, and multiple <*file-entry*> elements listing the package files, directories and their content types (Appendix 7).

24. All the files and directories making up the electronic document, except for the files *mimetype* and *manifest.xml*, should be listed in the manifest file.

25. References to files and directories are indicated by the relative IRI references (*ipath-noscheme* according to the RFC 3987 – Appendix 18, item 19), having the root directory of the package as the reference point (base IRI). The presence of absolute IRI references, as well as the references or their fragments indicating files or directories that are not within the package, is not permitted. The package is indicated by "/"character which cannot be the first character of other references.

26. The manifest file should indicate only these MIME type labels for files and directories of the electronic document that are listed in Appendix 9.

Relationships between the parts of the electronic document

27. Parts (files) of the electronic document are related with each other: the file of the main document may contain associated files of appendices or attached documents; files of the content of the electronic document and signable metadata are signed by electronic signatures.

28. The type of relationship in the package determines the way in which one part of the electronic document is related to another part of the electronic document. These relationships can also perform the function of a reference while determining the type of the part of the document (relationships of physical and logical structures of the package). Parts of the electronic document that have no direct cross-references may be logically co-related each to other by the means of relationships, too.

29. Relationships describe how the parts of the logical structure of the electronic document are represented in the physical structure of the electronic document and ensure the interoperability of the electronic documents when they are transferred from one system to another. Therefore there are no requirements on forcing the use of some predefined set of fixed names for files or directories for different parts of the electronic document.

Relationships file

30. The relationships file is an unsignable XML file describing relationships between the parts of the package of the electronic document and the files that make up these parts. The package of the electronic document should contain only one relationships file.

31. The name of relationships file is *META-INF/relations.xml*.

32. The <*Relationships*> element is the root element of the relationships file, containing one or more elements <*SourcePart*> that describe different relationships between the package or files in the package and other files in the package(Appendix 8).

33. The relationships file of the electronic document should contain:

33.1. One element <*SourcePart*> which describes the relationships of physical and logical structures of the package;

33.2. At least one element <*SourcePart*> which describes the relationships between the files of the electronic document;

34. The value "/" of the attribute *full-path* of the element <*SourcePart*> *(full-path="/")* identifies the package. Other values of the attribute *full-path* identify the references to files of the electronic document. The reference of the element's attribute *full-path* should unambiguously indicate the package itself or the file in the package.

35. References should be indicated in accordance with the requirements of the item 25.

36. The <*Relationship*> element describes the type of the relationship between the package or the file defined by the element <*SourcePart*> and another file of the document. Element's attribute *full-path* references the file of the electronic document that is related with the package or the file, whereas the attribute *type* defines the type of the relationship. Optional element's attribute *id* is the identifier of the relationship.

37. The set of types of relationships defined by this Specification is listed in Appendix 10. The author of the document can define its own set of types of relationships for relating the parts of the electronic document or the files constituting these parts whether it is required for the operation of the author's informational system. Upon detection of the relationship type that is unknown for the software, it should behave in the same way as upon detection of unknown type of the logical relationship between the parts of the document; for example, display the relationship indicating that the type of the relationship is unknown.

38. Two parts of the electronic document may be related by different types of relationships as well as by interrelationships that should be documented by two different <*SourcePart*> elements: one should define the relationship between file *A* and file *B*, while another one should define the relationship of file *B* and file *A*.

39. If the relationship with not a whole XML file but only with some part of it is needed (for example, if only several metadata elements in the XML file are signed by e-signature), <*Element*> elements should be used to list the related XML elements of the file. The indication of relationship with the element is possible only when the element has a unique identifier provided in the XML file. *in-source-part* attribute of the <*Element*> element is used for distinguishing the partially related XML file from one of two given files (referred by the

<*SourcePart*> element or referred by the <*Relationship*> element) where the referring XML element is present, and the *ref-id* attribute indicates the reference to that element by its identifier.

40. Possible values of the *in-source-part* attribute of the element <*Element*> are:

40.1. *true* – element refers to the XML element of file indicated by the attribute *full-path* of the element <*SourcePart*>;

40.2. *false* – element refers to the XML element of the file indicated by the attribute *full-path* of the element <*Relationship*>.

41. The physical structure of the electronic document is documented in the relationships file by the means of relationships between the physical and logical structures of the electronic document (Appendix 15); the following requirements should be observed:

41.1. The relationships file should contain one element <*SourcePart*> that identifies the package;

41.2. Files of the main document, signable and unsignable metadata, electronic signatures and thumbnail (if any) should be related with the package;

41.3. Files of document appendices should be related with the file of the main document or with the file of another appendix;

41.4. Files of the attached documents should be related with the file of the main document.

42. URI references to the signed parts of the electronic document are stored in the electronic signatures files. In order to determine efficiently (without using the software for analysing the content of e-signature files) whether the package contains the appropriate file and which e-signature has been used for signing it, the relationships file should contain the description of all files signed by the electronic signatures by the means of indication of files' relationships with the files of electronic signatures. When only a part of metadata file is signed by the electronic signature, all signed metadata elements should be listed by using identifiers generated for XML elements (their groups) uniquely within the XML file.

III. CONTENT OF THE ELECTRONIC DOCUMENT

43. The content of the electronic document consist of the following parts:

43.1. *main document* – the part of content of the electronic document providing the main information of the electronic document, or the covering document where the information on the attached electronic documents is recorded;

43.2. *appendices* of the document (optional) – part of the content of the electronic document providing the information of the content of the electronic document supplementing the main document;

43.3. *attached electronic documents* (optional) – independent electronic documents that are attached to the main document as the information supplementing or explaining its content or providing the basis for the facts provided in the main document, as well as electronic documents sent together with the covering document.

44. The main document of the electronic document should be made from a single file.

45. The content of the electronic document may consist of one or more appendices stored in separate files. Files of the main document and document appendices can cover one or more appendices, but the body of the main document or one appendix should not be split into several files (Appendix 2). Appendices may have other appendices stored in separate files.

46. The content of the electronic document may contain one or more files of the attached documents, single attached document per file. The content of the attached electronic document may comprise other attached documents.

47. The total number of parts comprising the content of the electronic document should not exceed the maximum values indicated in item 12 of the Specification.

48. The file of the main document of the electronic document should be stored in the root directory of the package. Files of the document appendices and attached electronic documents may be stored in one or more directories (Appendix 3).

49. Directories of appendices and/or attached documents may form the hierarchical structure, where lower levels of the hierarchical structure may contain one or more directories with one or more files of appendices or attached documents. The maximum number of levels comprising the hierarchical structure of directories should not be greater than 3.

Formats for files comprising the content of the electronic document

50. Files of the main document and appendices of the electronic document should be based on the set of open format files listed in Appendix 5 and should satisfy the following requirements:

50.1. The content of the file should be in conformity with the requirements of the file format specification;

50.2. The file extension suitable for the file format should be present;

50.3. The file format should be identified using the MIME type label indicated in the manifest file.

51. The attached independent electronic document should be in conformity with the requirements for the specification of the attached electronic document (Appendix 6).

52. The content of files should not be damaged or otherwise corrupted – it should be possible to review the content of the file with the software designed for displaying the content of file of the particular file format.

IV. METADATA

53. Metadata of the electronic document are the data providing the information on the format, structure, content, usage and signing of the electronic document.

54. Metadata can be divided into:

54.1. *signable* metadata – metadata that must be signed by electronic signature and the content of which should not be modified after signing (or the integrity of the part of them is guaranteed by the technology of qualified or unqualified e-signature);

54.2. *unsignable* metadata – metadata that should not be signed by electronic signature and the content of which may change during the life cycle of the electronic document.

55. Electronic document, depending on its life cycle, may contain metadata of the categories indicated in Appendix 11 of this Specification, part of which are signable metadata.

56. Metadata of the electronic document are recorded in XML and may be stored in several XML files. Depending to the time metadata were created and signed, metadata of different categories may be stored in the same file, while metadata of the same category may be stored in several files.

57. Signable and unsignable metadata should be stored in separate files. Electronic document should contain at least one signable metadata file and at least one unsignable metadata file. If the electronic document during its life cycle is supplemented with a group of signable and unsignable metadata, signable and unsignable metadata should be stored in separate files. All metadata of the electronic document are stored only in metadata files.

58. Files for the signable and unsignable metadata are identified by the appropriate relationship type indicated in relationships file (Appendix 10). Every metadata file of the electronic document must contain the reference to the XML scheme (Appendix 17, section 1).

59. Refer to Appendix 12 of this Specification for the available list of metadata of the electronic document, names of XML elements used for recording metadata, requirements for their recurrence, obligatoriness and signability depending on different categories of electronic documents.

60. Metadata XML file of the electronic document should contain the root <*metadata*> element containing the reference to the namespace, which is defined by the means of the XML scheme (Appendix 17, section 1). The content of this element may contain one or more metadata elements.

61. Logically related elements of the signable metadata categories, including the root element <*metadata*> of signable metadata, and with the exception if the element <*signatures*> (as metadata of different electronic signatures should be signed by the signature that they supplement), must have the attribute *ID* containing a unique identifier within the metadata file. The groups of the unsignable metadata elements may also have an attribute *ID* containing a unique value, when it is necessary to provide the reference to the metadata element or when the value of the metadata is changed by adding new metadata element to the electronic document.

V. ELECTRONIC SIGNATURES

62. Electronic signatures with the purpose of signing and (if necessary) approval of official electronic document should be created using the secure signature creation system.

63. Electronic document should be signed by the XAdES-EPES, XAdES-T, XAdES-C, XAdES-X, XAdES-X-L or XAdES-A forms of electronic signatures that are in conformity with the XAdES standard. As the XAdES standard is universal and allows alternatives, the Specification provides the detailed information on the usage of the elements for creating the XAdES signatures for this specific use case.

64. A detached electronic signature topology is used for signing electronic document. One file can contain only one electronic signature. Electronic document and individual parts of it may be signed in the following ways:

64.1. *Parallel* signature method is applied when every electronic signature is independent and used for signing content and metadata of the electronic document, except for signing other electronic signatures;

64.2. *Multi-stage* signature method is applied when the electronic signature is used for signing other electronic signatures as well.

65. All electronic counter-signatures used for the multi-stage signing model are stored in separate files; the type of their reference element <*Reference*> indicating another signature should be *http://uri.etsi.org/01903#CountersignedSignature*.

66. In order to ensure the validity of the electronic signature during the time elapsed, time-stamp tokens are used. Time-marks are not applicable. Only the implicit mechanism is used for embedding all time-stamp tokens according to the item 7.1.4.3 *XAdESTimeStampType data type* of the XAdES standard.

67. For details on the structure of the electronic signatures refer Appendix 13 of this Specification, algorithms available for formation of electronic signatures are listed in Appendix 14 of this Specification.

68. Metadata stored in signable metadata file(s) only should be signed. If only some metadata elements, stored in the metadata file, are required to be signed only those XML elements should be signed. As some of the metadata elements are logically related to each other, and should not be signed separately (e.g., registration date and registration number), only those XML elements that have an obligatory *ID* attribute should be signed (Appendix 12). Only the entire XML element comprising all the attributes and the entire sub-tree of XML elements should be signed.

69. A separate reference (element <*ds:Reference*>) for each signable XML element should be created in e-signature file. The value of the attribute *ds:URI* of the reference element <*ds:Reference*> should refer to the entire signable metadata file. The specific signable XML element is extracted by applying transformation algorithms (element <*ds:Transforms*>). XML element after the applying transformation and canonicalization algorithms should conform the following requirements (Appendix 16):

69.1. No XML element belonging to the sub-tree of the signable XML element can be deleted during transformations;

69.2. No attribute of any XML element belonging to the sub-tree of the signable XML element (including the signable XML element) can be deleted during transformations;

69.3. No new XML element can appear in the sub-tree of the signable XML element during transformations;

69.4. No new attribute can appear in the sub-tree of the signable XML element (as well as within the signable XML element) during transformations;

69.5. No reordering of XML elements is allowed in the sub-tree of the signable XML element during transformations.

70. This Specification also permits signing of the whole metadata file as a binary file; this method of signing indicates that all metadata elements contained in the metadata file are signed.

VI. VERIFICATION OF THE ELECTRONIC DOCUMENT

71. The process of the verification of the electronic document comprises the following stages (the order of the verification stages should not necessarily be identical to the one indicated in this item and items 72-82):

71.1. Verification of the package format as defined in this Specification;

71.2. Verification of the structure of the content of the electronic document;

71.3. Verification of electronic signatures present in the package;

71.4. Verification of metadata of the electronic document.

Verification of the package format

72. During the verification of the package conformance to the requirements of the Specification the following is determined:

72.1. Whether the package file size does not exceed the 4 GB;

72.2. Whether the package file is a ZIP archive file;

72.3. Whether the ZIP archive file contains all the required parts of the electronic document:

72.3.1. The main document file;

72.3.2. At least one signable metadata file;

72.3.3. At least one unsignable metadata file;

72.3.4. At least one electronic signature file;

72.3.5. The manifest file (*manifest.xml*);

72.3.6. The relationships file (*relations.xml*).

72.4. Whether the manifest file (*manifest.xml*) satisfies the following requirements of the Specification:

72.4.1. Whether it match the XML scheme applicable for the manifest file (Appendix 17, item 4);

72.4.2. Whether the file is in the *META-INF* directory;

72.4.3. Whether the file contains all files and directories comprising the electronic document, except for the files *manifest.xml* and *mimetype*;

72.4.4. Whether all the parts of the electronic document satisfy the requirements for labelling the files (directories) indicated in the Appendix 9 of this Specification;

72.5. Whether the relationships file (*relations.xml*) satisfies the following requirements of the Specification:

72.5.1. Whether it match the XML scheme applicable for the relationships file of the electronic document (Appendix 17, item 3);

72.5.2. Whether all the relationships defining the structure of the package are present and whether they satisfy the determined requirements;

72.5.3. Whether all the parts of the electronic document found in the relationships file are indicated correctly;

72.5.4. Whether all the parts of the electronic document indicated in the file for the description of relations are signed by determined electronic signatures;

72.5.5. Whether all the signed parts of the electronic document have appropriate relationships with electronic signatures.

72.6. Whether metadata files comply with the requirements for the metadata files:

72.6.1. Whether metadata files match XML schemes (Appendix 17, section 1);

72.6.2. Whether the electronic document contains all the required metadata;

72.6.3. Whether metadata with no recurrences are recorded once, with the exception of recurrent metadata (for example, metadata of the electronic signatures) (Appendix 17, section 2);

72.6.4. Whether metadata of the electronic signature are signed by the electronic signature with the identification number that matches the value "e-signature identification number" of the corresponding metadata;

72.6.5. Whether all metadata required to be signed are signed actually (Appendix 17, section 2);

72.7. Whether the files of electronic signatures satisfy the following requirements:

72.7.1. Whether they match the XML scheme for the signatures (Appendix 17, section 3);

72.7.2. Whether the signatures file is in the *META-INF* directory (or in its' sub-directory);

72.7.3. Whether the name of the signature file contains the substring *signatures*;

72.7.4. Whether the electronic signature file contains only one electronic signature.

72.8. Whether the entire content is signed of the finished electronic document;

72.9. Whether the file of the main document of the electronic document is in the root directory of the package.

72.10. Whether the maximum number of levels comprising the hierarchical structure of directories is not greater than 3.

Verification of content of the electronic document

73. Complete verification of the content of the electronic document comprises the following stages:

73.1. Verification of relationships between the parts of the electronic document:

73.1.1. Whether the electronic document contains only the single main document file;

73.1.2. Whether parts of the document content are identified unambiguously, i.e., whether it have content parts that are not listed in the relationships file; whether the main document file is not an appendix or an attached document of any other part; whether the appendix is not an attached document of any part; whether the attached document is not an appendix of any part;

73.1.3. Whether the files of the attached documents (if any) are related only with the main document file by using the type of relationship "Attached document file";

73.1.4. Whether the main document file and all the files of document appendices (if any) form an orderly hierarchical structure according to the relationship type "Appendix(es) file" where the file of main document is on top of this structure, i.e. whether this hierarchy is the only

hierarchy (comprising all attachment files), whether there are no cycles and references to the same file.

73.2. Verification of file content types:

73.2.1. Whether the manifest file list all files comprising the content of the electronic document;

73.2.2. Whether all files of the document content comply with the requirements for the content types;

73.3. Verification of the usability of the content: whether content of all files comprising the content of the electronic document correspond to the declared file format, determined by verification methods not limited to checking extensions of files.

Verification of the electronic signatures

74. During the verification of the electronic document all its signatures are verified. Every electronic signature should be in a separate file. Initial and subsequent signature verification methods are used for the verification of electronic signature so as indicated in standard LST CWA 14171:2005 (Appendix 18, item 3). That should be determined during the verification process of the electronic signature in the electronic document:

74.1. Whether the signature is in conformance with the XMLDSIG and XAdES standards;

74.2. Whether the certificates used for signing the document had been issued by trusted certification authorities;

74.3. Whether the time-stamp tokens present in the electronic signature had been issued by trusted time-stamping authorities;

74.4. Whether the information on certificates revocation embedded in the electronic signature had been issued by the trusted certification authorities or trusted certification service providers;

74.5. Whether the certificate used in signature creation is present in the electronic signature (whether element <*KeyInfo*> is present, and it has the element <*X509Data*> containing the signer's certificate (element <*X509Certificate*>);

74.6. Whether the electronic signature match one of the following forms of electronic signatures: XAdES-EPES, XAdES-T, XAdES-C, XAdES-X, XAdES-X-L or XAdES-A;

74.7. Whether all algorithms used in creation of the electronic signature (transformation, encoding, canonicalization, hashing, signing) are listed in Appendix 14;

74.8. Whether every counter-signature contains a reference to another signed electronic signature of the special type (*type="http://uri.etsi.org/01903#CountersignedSignature"*);

74.9. Whether the electronic signature is in conformity with other requirements of this Specification;

74.10. Whether all parts comprising the content of the electronic document are signed as full binary files (only entire file is signed, not a separate part of it; there are no transformations in the references to files comprising the document content).

75. Profiles of the certificates used must be in conformance with the standard LST ETSI TS 101 862 V1.3.3:2007 “Qualified certificate profile”.

76. Information on certificate revocation should be accessible by the OCSP service accessible by HTTP or HTTPS protocols, in accordance with the RFC 2560 (Appendix 18, item 15).

77. If OCSP service is unavailable, certificate revocation lists (CRL) should be used in conformance with the RFC 5280 (Appendix 18, item 21) and accessible by HTTP or HTTPS protocols.

78. Time-stamp token should be in conformance with the standard LST ETSI TS 101 861 V1.2.1:2002 “Time-stamp profile”. Time-stamping services should be accessible by using the HTTP protocol in accordance with the RFC 3161 (Appendix 18, item 16).

Verification of metadata

79. During the verification of the electronic document all its metadata are verified. Verification of metadata for conformity with the requirements for metadata files indicated in items 72.4-72.6 of this Specification should be performed first.

80. Verification of the electronic document whether it contains all metadata required and whether all the signable metadata are signed may be performed by using XML files intended for checking the conformity of the metadata with the requirements that are provided in Appendix 17, section 2 of this Specification. The mentioned files determine the requirements for the verification of the obligatoriness, signability and recurrence of metadata elements. They are stored outside the document package together with XML schemes for metadata and may be used with the verification tools complementing the XML schemes. The Specification defines four XML files dedicated for the verification of electronic documents belonging to different categories – one for each electronic document category (GeDOC, GGeDOC, BeDOC, CeDOC).

81. The XML file dedicated for verification of the conformity with the requirements may be selected by the value of the unsignable metadata element named “Electronic document category” (*metadata/Use/technical_environment/documentCategory*). When the metadata element is not present, the metadata conformity with the requirements for the electronic documents of the GeDOC category should be verified. The verification of metadata conformity

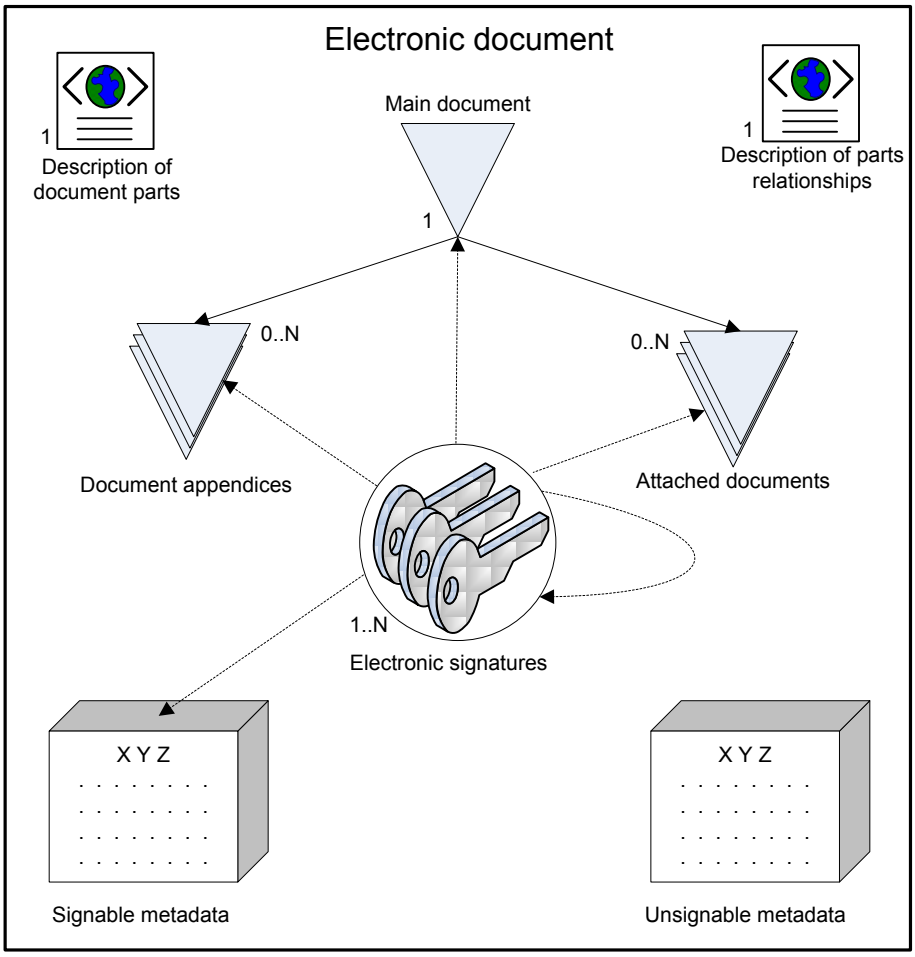
with the requirements is performed so as if all the metadata elements of the same XML namespace actually stored in different metadata files belong to the same metadata file.

82. Actually signed and not signed metadata are determined by the URI references and XPath transformations (if applicable) existing in electronic signature files. The relationships between parts of the electronic document enable more efficient determination whether particular metadata elements are signed and what signature has been used (without the usage of software for the analysing of the content of the electronic signatures file).

_____

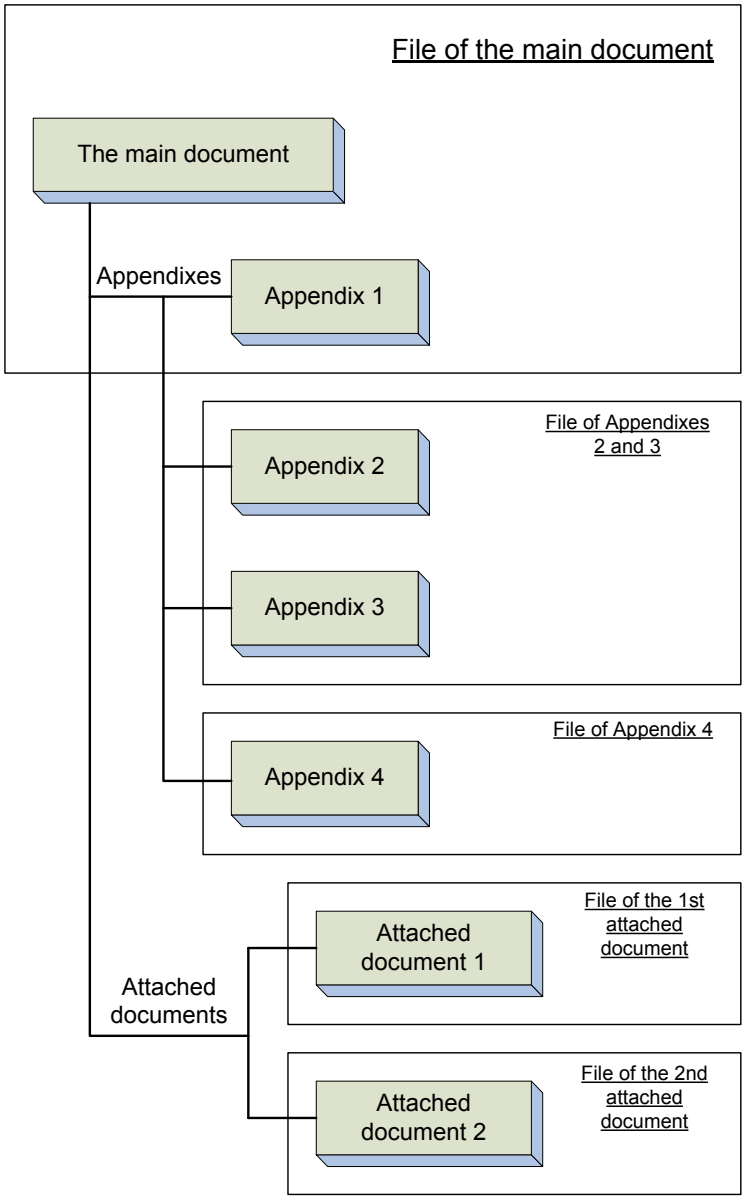
Specification ADOC-V1.0 of the electronic document signed by the electronic signature
Appendix 1

LOGICAL STRUCTURE OF THE ELECTRONIC DOCUMENT



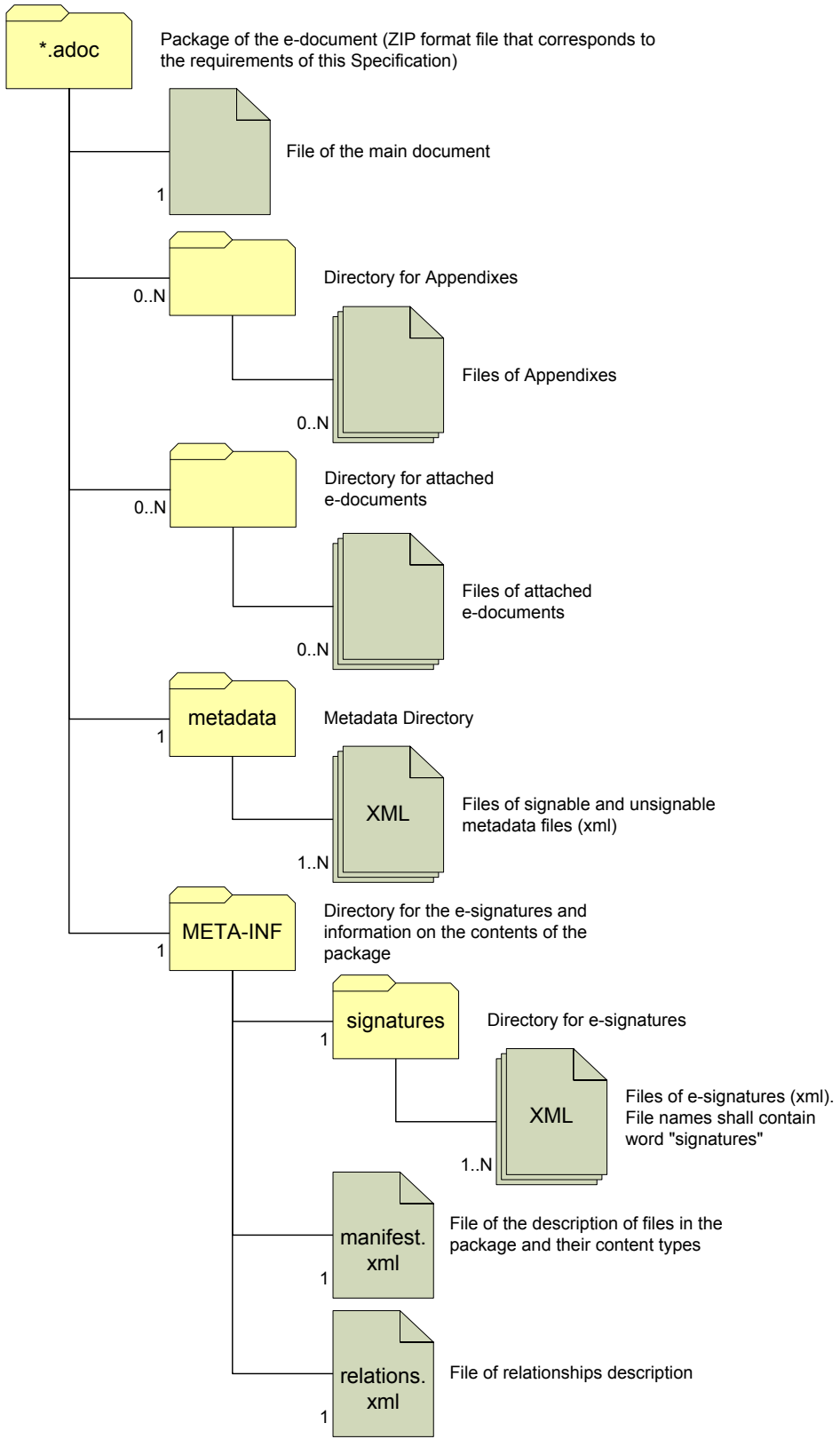
Specification ADOC-V1.0 of the electronic document signed by the electronic signature
Appendix 2

STRUCTURAL DIAGRAM OF THE CONTENT OF THE ELECTRONIC DOCUMENT



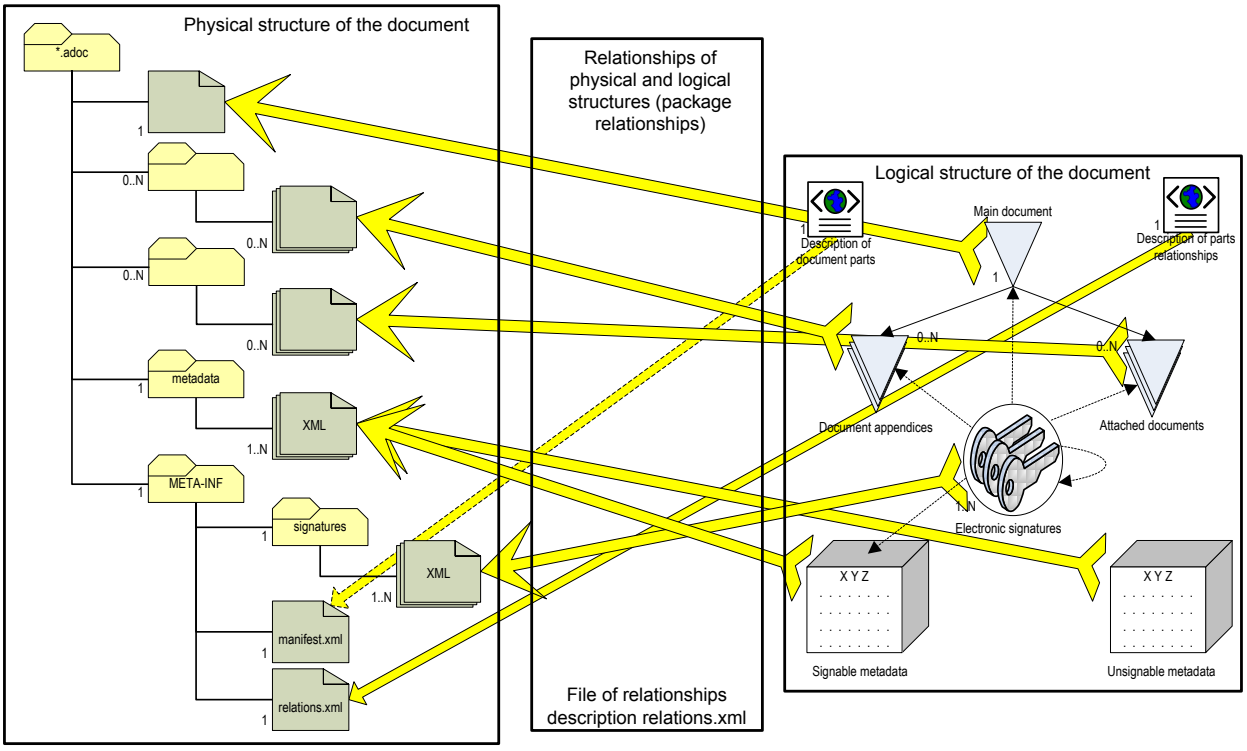
Specification ADOC-V1.0 of the electronic document signed by the electronic signature
Appendix 3

EXAMPLE OF THE PHYSICAL STRUCTURE OF THE ELECTRONIC DOCUMENT



Specification ADOC-V1.0 of the electronic document signed by the electronic signature
Appendix 4

REPRESENTATION OF LOGICAL STRUCTURE INTO PHYSICAL STRUCTURE OF THE ELECTRONIC DOCUMENT



Methods of the representation of logical structure in physical structure:

- representation of logical part by using fixed-name file (relationship is not indicated in the relationships file);

- representation of logical part (s) using file (s) and defining relationship (s) in the relationships description file.

Specification ADOC-V1.0 of the electronic document signed by the electronic signature
Appendix 5

FILE FORMATS FOR THE CONTENT OF THE ELECTRONIC DOCUMENT

No.	Criterion	Description
		1. **Document formats for word processing**
1.1.	**Format name:**	*Office Open XML word processing document format*
	File extension:	docx
	MIME type label:	application/vnd.openxmlformats-officedocument.wordprocessingml.document
	Format specification:	group of LST ISO/IEC 29500:2009 standards (Appendix 18, Item 13)
	Format realisation:	Microsoft Office Word 2007, earlier versions[1] of Microsoft Office Word and software of other manufacturers, Word Viewer (for preview)
1.2.	**Format name:**	*Open Office v 1.0 word processing document format*
	File extension:	odt
	MIME type label:	application/vnd.oasis.opendocument.text
	Format specification:	LST ISO/IEC 26300:2007 (Appendix 18, Item 12)
	Format realisation:	OpenOffice.org Writer 2 version and software of other manufacturers
		2. **Document formats for spreadsheets**
2.1.	**Format name:**	*Office Open XML spreadsheet document format*
	File extension:	xlsx
	MIME type label:	application/vnd.openxmlformats-officedocument.spreadsheetml.sheet
	Format specification:	group of LST ISO/IEC 29500:2009 standards (Appendix 18, Item 13)
	Format realisation:	Microsoft Office Excel 2007, earlier versions[1] of Microsoft Office Excel and software of other manufacturers, Excel Viewer (for preview)
2.2.	**Format name:**	*Open Office v 1.0 spreadsheet document format*
	File extension:	ods
	MIME type label:	application/vnd.oasis.opendocument.spreadsheet
	Format specification:	LST ISO/IEC 26300:2007 (Appendix 18, Item 12)
	Format realisation:	OpenOffice.org Calc 2 version and software of other manufacturers

No.	Criterion	Description
		3. **Document formats for presentations**
3.1.	**Format name:**	*Office Open XML presentation document format for development*
	File extension:	pptx
	MIME type label:	application/vnd.openxmlformats-officedocument.presentationml.presentation
	Format specification:	group of LST ISO/IEC 29500:2009 standards (Appendix 18, Item 13)
	Format realisation:	Microsoft Office PowerPoint 2007, earlier versions[1] of Microsoft Office PowerPoint and software of other manufacturers, PowerPoint Viewer 2007 (for preview)
3.2.	**Format name:**	*Office Open XML presentation document format for slide show*
	File extension:	ppsx
	MIME type label:	application/vnd.openxmlformats-officedocument.presentationml.slideshow
	Format specification:	group of LST ISO/IEC 29500:2009 standards (Appendix 18, Item 13)
	Format realisation:	Microsoft Office PowerPoint 2007, earlier versions[1] of Microsoft Office PowerPoint and software of other manufacturers, PowerPoint Viewer 2007 (for preview)
3.3.	**Format name:**	*Open Office v 1.0 XML presentation document format*
	File extension:	odp
	MIME type label:	application/vnd.oasis.opendocument.presentation
	Format specification:	LST ISO/IEC 26300:2007 (Appendix 18, Item 12)
	Format realisation:	OpenOffice.org Impress 2 version and software of other manufacturers
		4. **Vector graphics and text formats**
4.1.	**Format name:**	*PDF format for the long-term archiving (PDF/A)*
	File extension:	pdf
	MIME type label:	application/pdf *(PDF/A is distinguished from other formats by the metadata namespace: prefix – pdfaid, URI – http://www.aiim.org/pdfa/ns/id/)*
	Format specification:	LST ISO 19005-1:2008 (Appendix 18, Item 11)
	Format realisation:	Adobe Acrobat version 8, OpenOffice.org version 2, Microsoft Office 2007 and software of other manufacturers, Adobe Reader (for preview)

No.	Criterion	Description
4.2.	**Format name:**	*PDF file format*
	File extension:	pdf
	MIME type label:	application/pdf
	Format specification:	ISO 32000-1:2008 (Appendix 18, Item 14)
	Format realisation:	Adobe Acrobat version 8, OpenOffice.org version 2, Microsoft Office 2007 and software of other manufacturers, Adobe Reader (for preview)
	5. **Raster graphics formats**	
5.1.	**Format name:**	*Tagged Image File Format (TIFF)*
	File extension:	tif, tiff
	MIME type label:	image/tif, image/tiff, image/tiff-fx
	Format specification:	LST ISO 12234-2:2008 (Appendix 18, Item 9)
	Format realisation:	Scanners, software of various manufacturers
5.2.	**Format name:**	*Joint Photographic Experts Group (JPEG) format*
	File extension:	jpg, jpeg, jfif
	MIME type label:	image/jpeg
	Format specification:	LST ISO/IEC 10918-1:2009 (Appendix 18, Item 8)
	Format realisation:	Digital cameras, software of various manufacturers, internet browsers (for preview)
5.3.	**Format name:**	*Portable Network Graphics (PNG) format*
	File extension:	png
	MIME type label:	image/png
	Format specification:	LST ISO/IEC 15948:2009 (Appendix 18, Item 10)
	Format realisation:	Software of various manufacturers, internet browsers (for preview)

Note. Microsoft Corporation distributes the compatibility package for earlier versions of Microsoft Office 2003, Microsoft Office XP or Microsoft Office 2000 programs that enables to open, edit, save and create documents using open file formats (see Microsoft KB 924074 article *How to open and save Word 2007, Excel 2007, and PowerPoint 2007 files in earlier versions of Office programs*. Article ID: 924074, http://support.microsoft.com/kb/924074/en-us, 2007-04-24, 3.4).

Specification ADOC-V1.0 of the electronic document signed by the electronic signature
Appendix 6

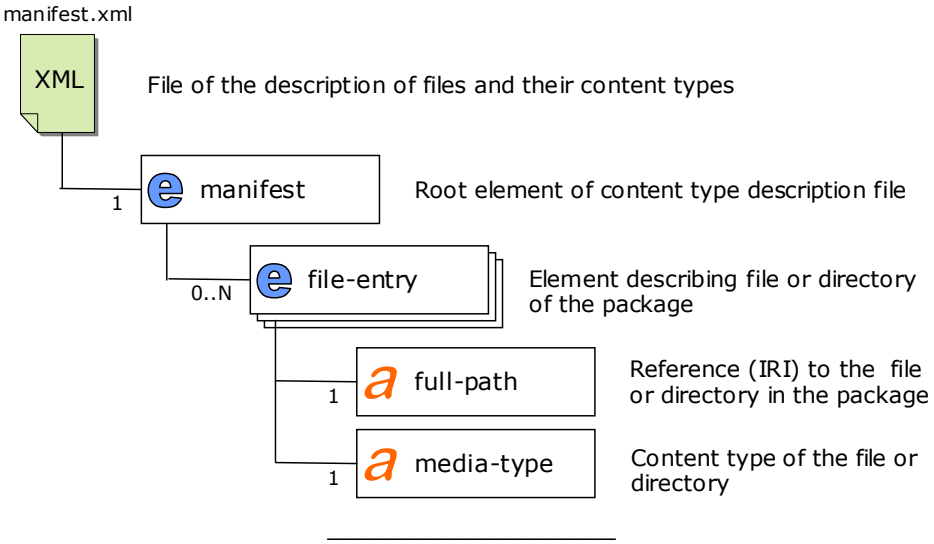
FORMATS FOR THE ATTACHED ELECTRONIC DOCUMENTS

No.	Requirement	Description
1.	**Format:**	Electronic document in conformance with this (ADOC-V1.0) specification, stored in the package
	File extension:	adoc
	MIME type label:	application/vnd.lt.archyvai.adoc-2008
2.	**Format:**	Electronic document stored in ZIP archive in conformance with the other specification of electronic document, confirmed by the Lithuanian Archives Department under the Government of the Republic of Lithuania, that does not conflict with this Specification.
	File extension:	*Should comply with that is defined in the appropriate specification of the electronic document*
	MIME type label:	

_____

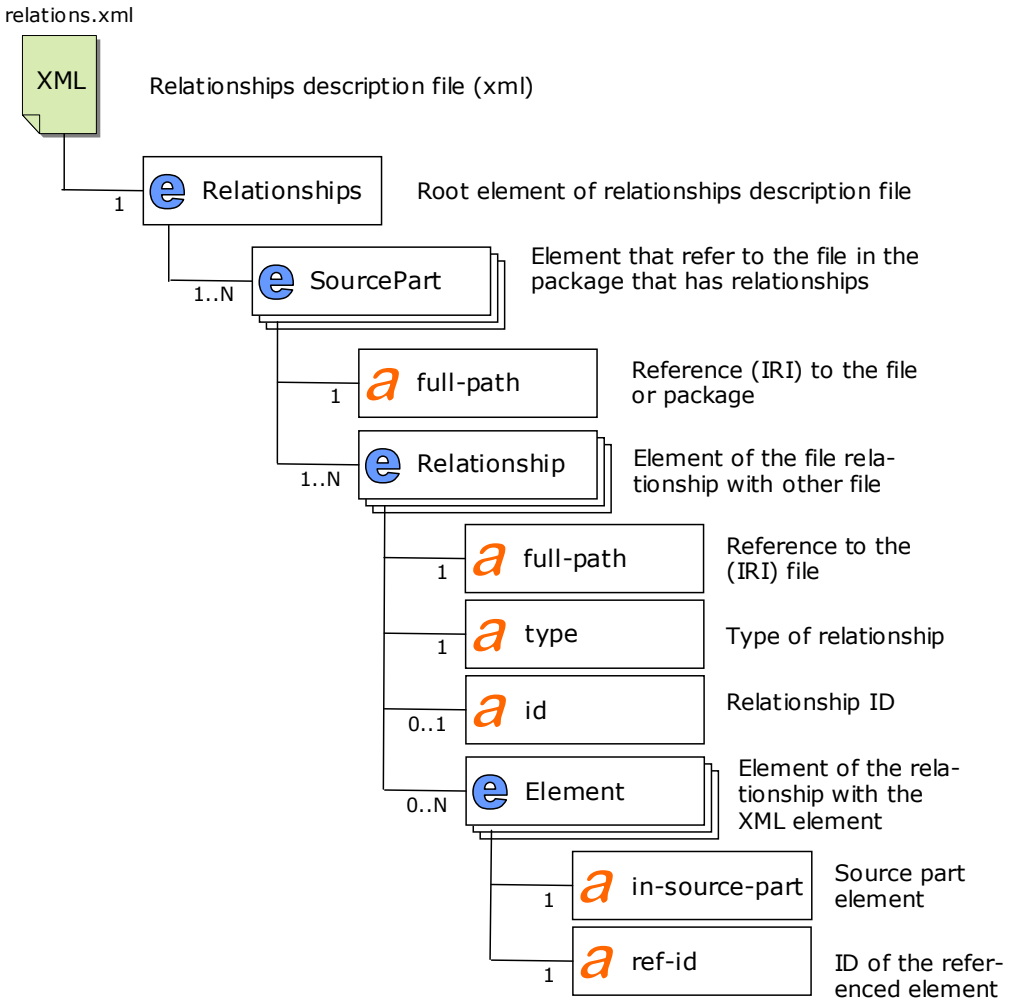
Specification ADOC-V1.0 of the electronic document signed by the electronic signature
Appendix 7

STRUCTURE OF THE MANIFEST FILE



Specification ADOC-V1.0 of the electronic document signed by the electronic signature
Appendix 8

STRUCTURE OF THE RELATIONSHIPS DESCRIPTION FILE



Specification ADOC-V1.0 of the electronic document signed by the electronic signature
Appendix 9

TYPES OF FILES AND DIRECTORIES COMPRISING THE ELECTRONIC DOCUMENT

Parts of the document	MIME type label of the file or directory
Package (*full-path="/"*)	application/vnd.lt.archyvai.adoc-2008
Document content files	*according to the file format (Appendices 0 and 0)*
Relationships file	text/xml
Metadata file	
Electronic signature file	
Thumbnail file	*empty value is indicated ("")*
Directory for metadata files	application/vnd.lt.archyvai.adoc-2008#metadata-folder
Directory for electronic signature files (if exist)	application/vnd.lt.archyvai.adoc-2008#signatures-folder
Other directories	*empty value is indicated ("")*

_____

Specification ADOC-V1.0 of the electronic document signed by the electronic signature
Appendix 10

TYPES OF RELATIONSHIPS

Type of relationship	Value of *type* attribute of the <*Relationship*> element
Main document file	http://www.archyvai.lt/adoc/2008/relationships/content/main
Appendix(es) file	http://www.archyvai.lt/adoc/2008/relationships/content/appendix
Attached document file	http://www.archyvai.lt/adoc/2008/relationships/content/attachment
Signable metadata file	http://www.archyvai.lt/adoc/2008/relationships/metadata/signable
Unsignable metadata file	http://www.archyvai.lt/adoc/2008/relationships/metadata/unsignable
Electronic signature file	http://www.archyvai.lt/adoc/2008/relationships/signatures
Thumbnail file	http://www.archyvai.lt/adoc/2008/relationships/thumbnail

_____

Specification ADOC-V1.0 of the electronic document signed by the electronic signature
Appendix 11

CATEGORIES OF ELECTRONIC DOCUMENT METADATA

Metadata category	Signing
Metadata for describing the document and its formation	signable
Metadata of document usage restrictions	signable
Metadata of document registration	signable[1]
Metadata of the registration of the received document	signable[1]
Metadata of electronic signature (s)	signable[2]
Technical metadata of the document	unsignable
Metadata of document storage	unsignable
Metadata of document encryption	unsignable
Other metadata	signable/unsignable
Additional metadata (not defined in this Specification)	signable/unsignable

NOTES:
[1] The technology of qualified or unqualified e-signature is used to ensure the integrity of metadata.
[2] Metadata should be signed by the same electronic signature that they complement.

Specification ADOC-V1.0 of the electronic document signed by the electronic signature
Appendix 12

METADATA OF THE ELECTRONIC DOCUMENT

Metadata	XML element	Data type	Recurrent	Mandatory for the category of e-documents				Signable
				GeDOC	GGeDOC	BeDOC	CeDOC	
SIGNABLE METADATA								
Root element of the metadata file	**metadata**	Element[7]	Yes[14]	Yes[10]	Yes[10]	Yes[10]	Yes[10]	Yes
Metadata for describing the document and its formation								
Metadata for describing content of e-document	**document**	Element[5]	No	Yes[10]	Yes[10]	Yes[10]	No[10]	Yes[3]
Title of e-document (heading)	title	Text	No	Yes	Yes	Yes	Yes[9]	Yes[1]
Document sort (e.g. order, statement, letter, contract, etc.)	sort	Text	No	No	No	No	No	Yes[3,1]
Authors:	**authors**	Element[5]	Yes[14]	Yes[10]	Yes[10]	Yes[10]	Yes[10]	Yes
Author:	**author**	Element[5]	Yes	Yes[10]	Yes[10]	Yes[10]	Yes[10]	Yes
Author (full name of the entity or first and last names of the person)	name	Text	No[15]	Yes	Yes	Yes	Yes	Yes[1]
Code of the author	code	Text	No[15]	Yes[13]	Yes[13]	Yes[13]	No	Yes[3, 1]
Address of the author	address	Text	No[15]	Yes	Yes	Yes	Yes	Yes[1]
The author is: person (yes) or legal entity (no).	individual	Logical	No[15]	Yes	Yes	Yes	No	Yes[3,1]
Document creation	**creation**	Element[5]	No	No[10]	No[10]	No[10]	No[10]	Yes[3]
Date of creation[12]	date	Date[2]	No	No[12]	No[12]	No[12]	No[12]	Yes[3, 1]
Recipients:	**recipients**	Element[5]	Yes[14]	No[10]	No[10]	No[10]	No[10]	Yes[3]
Recipient:	**recipient**	Element[5]	Yes	No[10]	No[10]	No[10]	No[10]	Yes[3]
Recipient (name or first and last names)	name	Text	No[15]	Yes[9]	Yes[9]	Yes[9]	Yes[9]	Yes[3, 1]
Code of the recipient	code	Text	No[15]	Yes[9,13]	Yes[9,13]	Yes[9,13]	Yes[9,13]	Yes[3, 1]
Address of the recipient	address	Text	No[15]	No	No	No	No	Yes[3, 1]

Metadata	XML element	Data type	Recurrent	Mandatory for the category of e-documents				Signable
				GeDOC	GGeDOC	BeDOC	CeDOC	
The recipient is: person (yes) or legal entity (no)	individual	Logical	No[15]	No	No	No	No	Yes[3,1]
Metadata of document usage restrictions								
Usage restrictions[4]:	**restrictions**	Element[5]	Yes[14]	No[10]	No[10]	No[10]	No[10]	Yes[3]
Usage restriction[4]:	**restriction**	Element[5]	Yes	No[10]	No[10]	No[10]	No[10]	Yes[3]
Use of document content is restricted[4]	contentRestriction	Logical	No[15]	No[6]	No[6]	No[6]	No	Yes[3, 1]
Use of metadata is restricted[4]	metadataRestriction	Logical	No[15]	No[6]	No[6]	No[6]	No	Yes[3, 1]
Date of introduction / removal of restrictions	date	Date[2]	No[15]	No	No	No	No	Yes[3, 1]
Reason for the introduction / removal of restrictions	reason	Text	No[15]	No	No	No	No	Yes[3, 1]
Metadata of document registration (metadata integrity is ensured with the e-signature as technology)								
Registrations of a document:	**registrations**	Element[5]	Yes[14]	Yes[10]	No[10]	No[10]	No[10]	Yes[3]
Registration of a document:	**registration**	Element[5]	Yes	Yes[10]	No[10]	No[10]	No[10]	Yes[3]
Date of registration	date	Date[2]	No[15]	Yes	Yes[9]	Yes[9]	Yes[9]	Yes[1]
Document registration No.	number	Text	No[15]	Yes	Yes[9]	Yes[9]	Yes[9]	Yes[1]
Employee who registered the document	registrar	Employee[8]	No[15]	No	No	No	No	Yes[3, 1]
Code of the entity (institution) that registered the document	code	Text	No[15]	No	No	No	No	Yes[3, 1]
Metadata of the received document registration (metadata integrity is ensured with the e-signature as technology)								
Registrations of a received documents:	**receptions**	Element[5]	Yes[14]	No[6, 10]	Yes[10, 11]	No[10]	No[10]	Yes[3]
Registration of a received document:	**reception**	Element[5]	Yes	No[6, 10]	Yes[10, 11]	No[10]	No[10]	Yes[3]
Reception date	date	Date[2]	No[15]	Yes[9]	Yes[9]	Yes[9]	Yes[9]	Yes[3, 1]
Registration No. of the reception	number	Text	No[15]	Yes[9]	Yes[9]	Yes[9]	Yes[9]	Yes[3, 1]

Metadata	XML element	Data type	Recurrent	Mandatory for the category of e-documents				Signable
				GeDOC	GGeDOC	BeDOC	CeDOC	
Employee who registered the document	registrar	Employee[8]	No[15]	No	No	No	No	Yes[3, 1]
Receiver of the document:	**receiver**	Text	No[15]	Yes[9,10]	Yes[9,10]	Yes[9,10]	Yes[9,10]	Yes[3, 1]
Receiver name or first name and last name)	name	Text	No[15]	Yes[9]	Yes[9]	Yes[9]	Yes[9]	Yes[3, 1]
Code of the receiver	code	Text	No[15]	Yes[9,13]	Yes[9]	Yes[9,13]	Yes[9,13]	Yes[3, 1]
Address of the receiver	address	Text	No[15]	No	No	No	No	Yes[3, 1]
The receiver is: person (yes) or legal entity (no)	individual	Logical	No[15]	Yes[9]	No	Yes[9]	Yes[9]	Yes[3,1]
Metadata of electronic signatures (metadata should be signed by the same electronic signature that they complement).								
Metadata of e-signatures:	**signatures**	Element[5]	Yes[14]	Yes	Yes	Yes	Yes	Yes[1]
Metadata of e-signature:	**signature**	Element[5]	Yes	Yes[10]	Yes[10]	Yes[10]	Yes[10]	Yes
E-signature ID	signatureID	Reference (IRI)	No[15]	Yes	Yes	Yes	Yes	Yes[1]
Signing date and time	signingTime	Date[2]	No[15]	Yes	Yes	Yes	Yes	Yes[1]
Purpose of e-signature: signature, confirmation, visa, conciliation, registration, registration-of-incomming-documents, acknowledgement, notarisation, copy-certification	signingPurpose	Text (selective)	No[15]	Yes	Yes	Yes	Yes	Yes[1]
Signer	signer	Employee[8]	No[15]	Yes	Yes	Yes	Yes	Yes[1]
E-signature metadata transferred from the original e-document to the copy of it with the transformed content (signed by e-signature with the purpose of copy certification)	**original_signatures**	Element (of the same type for "signatures" element)	Yes[14]	No	No	No	No	Yes[3, 1]

Metadata	XML element	Data type	Recurrent	Mandatory for the category of e-documents				Signable
				GeDOC	GGeDOC	BeDOC	CeDOC	
UNSIGNABLE METADATA								
Root element of the metadata file	**metadata**	Element[7]	Yes[14]	Yes	Yes	Yes	Yes	No
Technical metadata of the document								
Metadata for e-document usage:	**Use**	Element[5]	No	Yes	Yes	Yes	Yes	No
Technical information:	**technical_environment**	Element[5]	No	Yes	Yes	Yes	Yes	No
ID of the specification of the electronic document: ADOC-V1.0	standardVersion	Text	No	Yes	Yes	Yes	Yes	No
Group of the electronic document: GeDOC, GGeDOC, BeDOC, CeDOC	documentCategory	Text (selective)	No	No	Yes	Yes	Yes	No
Name and version of DMS used for creating the electronic document	generator	Text	No	No	No	No	No	No
Name and version of the operating system where the document was created	os	Text	No	No	No	No	No	No
Metadata for document storage and modification								
Location of e-document:	**Location**	Element[5]	No	Yes	Yes[11]	No	No	No
Index (es) of the case (volume) the document is assigned to	case_id	Text	Yes	Yes	Yes[11]	No	No	No
Storage location	storage	Text	No	No	No	No	No	No
Metadata for describing events of e-document:	**Event_history**	Element[7]	Yes[14]	No	No	No	No	No
Sending:	**sent**	Element[5]	Yes	No	No	No	No	No
The document sent date	date	Date[2]	No[15]	Yes[9]	Yes[9]	Yes[9]	Yes[9]	No
Employee responsible for sending the document	responsible	Employee[8]	No[15]	No	No	No	No	No
Reason for sending	reason	Text	No[15]	No	No	No	No	No

Metadata	XML element	Data type	Recurrent	Mandatory for the category of e-documents				Signable
				GeDOC	GGeDOC	BeDOC	CeDOC	
Sender:	**sender**	Text	No[15]	No	No	No	No	No
Sender (name or first and last names)	name	Text	No[15]	Yes[9]	Yes[9]	Yes[9]	Yes[9]	No
Sender code	code	Text	No[15]	No	No	No	No	No
Sender address	address	Text	No[15]	No	No	No	No	No
The sender is: person (yes) or legal body (no)	individual	Logical	No[15]	No	No	No	No	No
Re-classification:	**reclassified**	Element[5]	Yes	No	No	No	No	No
Assignment date	date	Date[2]	No[15]	Yes[9]	Yes[9]	Yes[9]	Yes[9]	No
Employee who performed the assignment	responsible	Employee[8]	No[15]	No	No	No	No	No
Reason for re-classification	reason	Text	No[15]	No	No	No	No	No
Index (es) of the newly assigned case (volume)	case_id	Text	Yes	No	No	No	No	No
Relocation:	**moved_from_location**	Element[5]	Yes	No	No	No	No	No
Date of relocation	date	Date[2]	No[15]	Yes[9]	Yes[9]	Yes[9]	Yes[9]	No
Employee who performed relocation	responsible	Employee[8]	No[15]	No	No	No	No	No
Reason for relocation	reason	Text	No[15]	No	No	No	No	No
New storage location	storage	Text	No[15]	No	No	No	No	No
Metadata (document) change:	**changed**	Element[5]	Yes	No	No	No	No	No
Change date	date	Date[2]	No[15]	Yes[9]	Yes[9]	Yes[9]	Yes[9]	No
Employee who performed the change	responsible	Employee[8]	No[15]	No	No	No	No	No
Reason for the change	reason	Text	No[15]	No	No	No	No	No
Change contents	abstract	Text	No[15]	No	No	No	No	No
Reference to the changed metadata element (*#ID* or *file#ID*)	reference	Reference (IRI)	No[15]	No	No	No	No	No

Metadata	XML element	Data type	Recurrent	Mandatory for the category of e-documents				Signable
				GeDOC	GGeDOC	BeDOC	CeDOC	
New metadata value (newly recorded content of the changed metadata element)	new_value	Undefined (should correspond to the type of the changed element)	No[15]	No	No	No	No	No
Information on transformation to other formats:	**transformed**	Element[5]	Yes	No	No	No	No	No
Transformation date	date	Date[2]	No[15]	Yes[9]	Yes[9]	Yes[9]	Yes[9]	No
Employee who performed the transformation	responsible	Employee[8]	No[15]	No	No	No	No	No
New format	format	Text	No[15]	No	No	No	No	No
Deletion:	**disposed**	Element[5]	Yes	No	No	No	No	No
Deletion date	date	Date[2]	No[15]	Yes[9]	Yes[9]	Yes[9]	Yes[9]	No
Employee who performed the deletion	responsible	Employee[8]	No[15]	No	No	No	No	No
Reason of the deletion	reason	Text	No[15]	No	No	No	No	No
Information on backup restore:	**restored**	Element[5]	Yes	No	No	No	No	No
Restore date	date	Date[2]	No[15]	Yes[9]	Yes[9]	Yes[9]	Yes[9]	No
Employee who performed restoration	responsible	Employee[8]	No[15]	No	No	No	No	No
Reason for reastoration	reason	Text	No[15]	No	No	No	No	No
Persons:	**Agent**	Element[5]	Yes[14]	No	No	No	No	No
Responsible persons:	**responsibilities**	Element[5]	Yes[14]	No	No	No	No	No
Responsibility:	**responsibility**	Element[5]	Yes	No	No	No	No	No
Responsibility area: creation, management, relocation, storage, deletion	area	Text (selective)	No[15]	No	No	No	No	No
Responsible employee	responsible	Employee[8]	No[15]	No	No	No	No	No
Metadata of document encryption (*N/A*; see Item 22 of this Specification)								*N/A*

Metadata	XML element	Data type	Recurrent	Mandatory for the category of e-documents				Signable
				GeDOC	GGeDOC	BeDOC	CeDOC	
Other metadata								
Metadata for describing the contents of e-document:	**Description**	Element[5]	Yes[14]	No	No	No	No	No
Information on e-document appendixes:	**appendixes**	Element[5]	Yes[14]	No	No	No	No	No
Information on e-document appendix:	**appendix**	Element[5]	Yes	No	No	No	No	No
Title of appendix	title	Text	No[15]	No	No	No	No	No
Appendix number	number	Text	No[15]	No	No	No	No	No
Metadata for describing events of e-document	**Event_history**	Element[7]	Yes[14]	No	No	No	No	No
Resolution:	**resolution**	Element[5]	Yes	No	No	No	No	No
Date of resolution	date	Date[2]	No[15]	Yes[9]	Yes[9]	Yes[9]	Yes[9]	No
Author of resolution	author	Employee[8]	No[15]	No	No	No	No	No
Text of resolution	text	Text	No[15]	No	No	No	No	No
Task executors	**executors**	Element[5]	No[15]	No	No	No	No	No
Task executor	executor	Employee[8]	Yes	No	No	No	No	No
Due by date of the task	due_by	Date[2]	No[15]	No	No	No	No	No
Information on task execution:	**executed**	Element[5]	Yes	No	No	No	No	No
Date of execution end	date	Date[2]	No[15]	Yes[9]	Yes[9]	Yes[9]	Yes[9]	No
Task executor	responsible	Employee[8]	No[15]	No	No	No	No	No
Comments on execution	abstract	Text	No[15]	No	No	No	No	No
Execution due date change:	**postponed**	Element[5]	Yes	No	No	No	No	No
Postponement date	date	Date[2]	No[15]	Yes[9]	Yes[9]	Yes[9]	Yes[9]	No
Resolution reference	reference	Reference (IRI)	No[15]	No	No	No	No	No
The new due by date	due_by	Date[2]	No[15]	No	No	No	No	No
Additional metadata, not defined in the Specification	**Custom**	Undefined	Yes[14]	No	No	No	No	Yes[10]/No

NOTES:

[1] Metadata element is not signed separately but it is signed as an integrated part of earlier defined metadata groups the element belongs to.

[2] Date or date and time as it is defined in the W3C recommendations for the XML format (*xsd:dateTime* or *xsd:date* type with the specified time zone fragment).

[3] Signed if element is provided.

[4] In case when there are no metadata provided, restrictions on the use are not applied.

[5] Element for group of metadata elements provided below and composing the same metadata group.

[6] Metadata are required for some specific kinds of documents.

[7] Element for group of metadata elements provided below as well as in other parts of the table and composing the same metadata group.

[8] Employee data type:

First name and last name of the responsible person	individualName	Text	No[15]	Yes	Yes	Yes	Yes	MBS
Position of the responsible person	positionName	Text	No[15]	Yes	Yes/No	Yes	No	MBS
Structural subdivision	structuralSubdivision	Text	No[15]	No	No	No	No	MBS

MBS – metadata may be signed.

[9] Mandatory when the parental element is present.

[10] Element (if defined) should have an ID attribute with the value that is unique within the metadata file.

[11] Element is mandatory when the document is registered in the institution or entity that has received the document.

[12] Document is considered as composed when it is signed and (if applicable) registered in the composing institution. If metadata element is provided, its value should comply with the latest of these dates: document signing date (dates in case it is signed by multiple persons) and (if applicable) document registration(s) date(s).

[13] Personal code of the natural person is not mandatory.

[14] Electronic document may contain several such metadata elements, but for a single element per metadata file.

[15] Within the parental element.

Specification ADOC-V1.0 of the electronic document signed by the electronic signature
Appendix 13

STRUCTURE OF ELECTRONIC SIGNATURES

1. Obligation of elements is defined as:

Mandatory	Creating application	Processing application
Yes	Should create such element	Should process such element
No	May create such element	May process such element if it exists
Inadmissible	Cannot create such element	Does not process such element and reports an error

2. Element namespace prefixes are defined as:

Prefix	Namespace	Standard
ds	http://www.w3.org/2000/09/xmldsig#	XMLDSIG
not defined	http://uri.etsi.org/01903/v1.3.2#	LST ETSI TS 101 903 V1.3.2:2006
xadesv141	http://uri.etsi.org/01903/v1.4.1#	LST ETSI TS 101 903 V1.3.2:2009

3. The structure of **XAdES-EPES** form signature:

XML element	Mandatory	Number	Note
ds:Signature	Yes	1	Each file may contain only one e-signature. Attribute *id* of the element is mandatory.
ds:SignedInfo	Yes	1	
ds:CanonicalizationMethod	Yes	1	
ds:SignatureMethod	Yes	1	
ds:Reference	Yes	2-N	References may refer only objects within the document package. One of them should refer to the signed properties (*SignedProperties*) being inside the signature file. If the reference refers to the object present in the same signature file, then the value of attribute *ds:URI* should contain only the fragment part of URI (e.g. *URI="#SignPr"*). If the reference refers to the object not present in the same signature file then the value of attribute *ds:URI* should

XML element	Mandatory	Number	Note
			contain only reference to the file (URI fragment should not be used).
ds:Transforms	No	0-N	
ds:DigestMethod	Yes	1	
ds:DigestValue	Yes	1	
ds:SignatureValue	Yes	1	
ds:KeyInfo	Yes	1	Should contain element *X509Data*, containing the certificate element *X509Certificate*
ds:Object	Yes	1	
QualifyingProperties	Yes	1	
SignedProperties	Yes	1	
SignedSignatureProperties	Yes	1	
SigningTime	No	0-1	If element is present its value should be of *xsd:dateTime* type, of the *Z* (*UTC*) time zone and should correspond to the metadata value "*Signing date*"
SigningCertificate	Yes	1	Should refer to the certificate stored in element *KeyInfo*
SignaturePolicyIdentifier	Yes	1	Implicit or explicit signature policy should be referenced
SignatureProductionPlace	No	0-1	
SignerRole	No	0-1	
SignedDataObjectProperties	No	0-1	
DataObjectFormat	No	0-N	If element is present, content types of the files referenced in this element should match MIME types referenced in the manifest file (*manifest.xml*) and the processing application should ensure that data will be displayed properly. Each element present should contain an element *MimeType*

XML element	Mandatory	Number	Note
CommitmentTypeIndication	No	0-N	If element is present, its value should correspond to the value of metadata element "*Purpose of e-signature*"
AllDataObjectsTimeStamp	Inadmissible	0	
IndividualDataObjectsTimeStamp	Inadmissible	0	
UnsignedProperties	No	0-1	
UnsignedSignatureProperties	No	0-1	
CounterSignature	Inadmissible	0	See item 65 of the Specification
xadesv141:TimeStampValidationData (for time stamp token *AllDataObjectsTimeStamp* verification data)	Inadmissible	0	
xadesv141:TimeStampValidationData (for time stamp token *IndividualDataObjectsTimeStamp* verification data)	Inadmissible	0	

4. The structure of **XAdES-T** form signature corresponds to the structure of XAdES-EPES form with the <*UnsignedSignatureProperties*> element complemented by the time stamp token and (if necessary) with the time stamp token verification data:

XML element	Mandatory	Number	Note
UnsignedSignatureProperties	Yes	1	
SignatureTimeStamp	Yes	1	Time stamp token is embedded into the *EncapsulatedTimeStamp* element; *XMLTimeStamp* element is inadmissible
xadesv141:TimeStampValidationData (for time stamp token *SignatureTimeStamp* verification data)	No	0-1	If element is present, it should be the next element after the *SignatureTimeStamp* element within the *UnsignedSignatureProperties* element and should contain the time stamp token verification data; attribute *URI* of the element is not used (XAdES standard, item 8.1.1).

5. The structure of **XAdES-C** form signature corresponds to the structure of XAdES-T form with the <*UnsignedSignatureProperties*> element complemented by these elements:

XML element	Mandatory	Number	Note
UnsignedSignatureProperties	Yes	1	
CompleteCertificateRefs	Yes	1	

XML element	Mandatory	Number	Note
CompleteRevocationRefs	Yes	1	Only *CRLRefs* and / or *OCSPRefs* elements are used (*OtherRefs* element is inadmissible)
AttributeCertificateRefs	Inadmissible	0	
AttributeRevocationRefs	Inadmissible	0	

6. The structure of **XAdES-X** form signature corresponds to the structure of XAdES-C form with the <*UnsignedSignatureProperties*> element complemented by these elements:

XML element	Mandatory	Number	Note
UnsignedSignatureProperties	Yes	1	
SigAndRefsTimeStamp	Yes	1	Implementation is based on not distributed case (XadES standard, item 7.5.1.1). Time stamp token is embedded into element *EncapsulatedTimeStamp*; *XMLTimeStamp* element is inadmissible
xadesv141:TimeStampValidationData (for time stamp token *SigAndRefsTimeStamp* verification data)	No	0-1	If element is present, it should be the next element after the *SigAndRefsTimeStamp* element within the *UnsignedSignatureProperties* element and should contain the time stamp token verification data; attribute *URI* of the element is not used (XAdES standard, item 8.1.1).
RefsOnlyTimeStamp	Inadmissible	0	
xadesv141:TimeStampValidationData (for time stamp token *RefsOnlyTimeStamp* verification data)	Inadmissible	0	

7. The structure of **XAdES-X-L** form signature corresponds to the structure of XAdES-X form with the <*UnsignedSignatureProperties*> element complemented by these elements:

XML element	Mandatory	Number	Note
UnsignedSignatureProperties	Yes	1	
CertificateValues	Yes	1	Element *EncapsulatedX509Certificate* is used for certificate values; element *OtherCertificate* is inadmissible

XML element	Mandatory	Number	Note
RevocationValues	Yes	1	Only *CRLValues* and/or *OCSPValues* elements are used for storing the certificate revocation values (element *OtherValues* is inadmissible)
AttrAuthoritiesCertValues	Inadmissible	0	
AttributeRevocationValues	Inadmissible	0	

8. The structure of **XAdES-A** form signature corresponds to the structure of XAdES-X-L form with the <*UnsignedSignatureProperties*> element complemented by the archive time stamp tokens:

XML element	Mandatory	Number	Note
UnsignedSignatureProperties	Yes	1	
xadesv141:ArchiveTimeStamp	Yes	1-N	Implementation is based on the not distributed case (XadES standard, 8.2.1 item). Archive time stamp token is embedded into element *EncapsulatedTimeStamp*; *XMLTimeStamp* element is inadmissible
xadesv141:TimeStampValidationData (for archive time stamp token *xadesv141:ArchiveTimeStamp* verification data)	No	0-N	If element is present, it should be the next element after the *xadesv141:ArchiveTimeStamp* element within the *UnsignedSignatureProperties* element and should contain the time stamp token verification data; attribute *URI* of the element is not used (XAdES standard, item 8.1.1).

Specification ADOC-V1.0 of the electronic document signed by the electronic signature
Appendix 14

ALGORITHMS FOR FORMATION OF E-SIGNATURES

Only the following set of algorithms should be used for e-signatures formation:

Algorithm	Identifier
Digest	
SHA1	http://www.w3.org/2000/09/xmldsig#sha1
SHA256	http://www.w3.org/2001/04/xmlenc#sha256
Encoding	
Base64	http://www.w3.org/2000/09/xmldsig#base64
Signature	
DSAwithSHA1 (DSS)	http://www.w3.org/2000/09/xmldsig#dsa-sha1
RSAwithSHA1	http://www.w3.org/2000/09/xmldsig#rsa-sha1
RSAwithSHA256	http://www.w3.org/2001/04/xmldsig-more#rsa-sha256
Canonicalization	
Canonical XML 1.0 (omits comments)	http://www.w3.org/TR/2001/REC-xml-c14n-20010315
Canonical XML 1.0 with Comments	http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments
Canonical XML 1.1 (omits comments)	http://www.w3.org/2006/12/xml-c14n11
Canonical XML 1.1 with Comments	http://www.w3.org/2006/12/xml-c14n11#WithComments
Transform	
XPath	http://www.w3.org/TR/1999/REC-xpath-19991116
Base64	http://www.w3.org/2000/09/xmldsig#base64

Specification ADOC-V1.0 of the electronic document signed by the electronic signature
Appendix 15

AN EXAMPLE OF E-DOCUMENT STRUCTURE DOCUMENTATION

1. If e-document consists of 4 content files:

- Main document file *Pagrindinis.docx*;
- 2 appendixes (*PriedasA.xlsx*, *Taisyklės.odt* (Regulations)) and appendix of the Regulations *PriedasT1.odp*;
- 1 attached document (*Pridedama2.adoc*);
- Signable metadata (*pasirašomi.xml*);
- Unsignable metadata (*istorija.xml*);
- 4 e-signatures:
  - *signatures1.xml* (visa) by which appendix A was approved and signable metadata with element ID "*parasas-signatures1"* was signed;
  - *signatures-dir.xml* (signature) by which all document content files and signable metadata with element IDs "*pavadinimas"*, "*sudarytojai"*, "*parasas-signatures-dir"* were signed;
  - *signatures-reg.xml* (document registration purpose), that has references to the main file of document content *Pagrindinis.docx*, e-signature *signatures-dir.xml* and signable metadata with element IDs "*registravimas"* and "*parasas-signatures-reg"*;
  - *signatures-greg.xml* (registration of received document purpose), that has references to the main file of document content *Pagrindinis.docx*, e-signatures *signatures-dir.xml, signatures-reg.xml* and signable metadata with element IDs "*gauto-dokumento-registravimas"* and "*parasas-signatures-greg"*.

2. If informational system of the document author records the information on other documents mentioned in the documents marking them by the means of relationships of type „*http://test.com/reference*“, and the main document file is interrelated with the attached document (two-way relationships).

3. The possible structure of e-document is provided as an example:

```
Pagrindinis.docx
priedai/PriedasA.xlsx
priedai/Taisyklės.odt
priedai/PriedasT1.odp
pridedami/Pridedama2.adoc
metadata/pasirašomi.xml
metadata/istorija.xml
META-INF/signatures/signatures1.xml
META-INF/signatures/signatures-dir.xml
META-INF/signatures/signatures-reg.xml
META-INF/signatures/signatures-greg.xml
META-INF/manifest.xml
META-INF/relations.xml
```

4. The content of relationships description file *META-INF/relations.xml* may be as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<Relationships xmlns="http://www.archyvai.lt/adoc/2008/relationships">
  <SourcePart full-path="/">
    <Relationship full-path="Pagrindinis.docx"
       type="http://www.archyvai.lt/adoc/2008/relationships/content/main"/>
    <Relationship full-path="metadata/pasirašomi.xml"
       type="http://www.archyvai.lt/adoc/2008/relationships/metadata/signable"/>
    <Relationship full-path="metadata/istorija.xml"
       type="http://www.archyvai.lt/adoc/2008/relationships/metadata/unsignable"/>
    <Relationship full-path="META-INF/signatures/signatures1.xml"
       type="http://www.archyvai.lt/adoc/2008/relationships/signatures"/>
    <Relationship full-path="META-INF/signatures/signatures-dir.xml"
       type="http://www.archyvai.lt/adoc/2008/relationships/signatures"/>
    <Relationship full-path="META-INF/signatures/signatures-reg.xml"
       type="http://www.archyvai.lt/adoc/2008/relationships/signatures"/>
    <Relationship full-path="META-INF/signatures/signatures-greg.xml"
       type="http://www.archyvai.lt/adoc/2008/relationships/signatures"/>
  </SourcePart>

  <SourcePart full-path="Pagrindinis.docx">
    <Relationship full-path="priedai/PriedasA.xlsx"
       type="http://www.archyvai.lt/adoc/2008/relationships/content/appendix"/>
    <Relationship full-path="priedai/Taisyklės.odt"
       type="http://www.archyvai.lt/adoc/2008/relationships/content/appendix"/>
    <Relationship full-path="pridedami/Pridedama2.adoc"
       type="http://www.archyvai.lt/adoc/2008/relationships/content/attachment"/>
    <Relationship full-path="META-INF/signatures/signatures-dir.xml"
       type="http://www.archyvai.lt/adoc/2008/relationships/signatures"/>
    <Relationship full-path="META-INF/signatures/signatures-reg.xml"
       type="http://www.archyvai.lt/adoc/2008/relationships/signatures"/>
    <Relationship full-path="META-INF/signatures/signatures-greg.xml"
       type="http://www.archyvai.lt/adoc/2008/relationships/signatures"/>
    <Relationship full-path="pridedami/Pridedama2.adoc"
       type="http://test.com/reference"/>
  </SourcePart>
  <SourcePart full-path="priedai/PriedasA.xlsx">
    <Relationship full-path="META-INF/signatures/signatures1.xml"
         type="http://www.archyvai.lt/adoc/2008/relationships/signatures"/>
    <Relationship full-path="META-INF/signatures/signatures-dir.xml"
         type="http://www.archyvai.lt/adoc/2008/relationships/signatures"/>
  </SourcePart>
  <SourcePart full-path="priedai/Taisyklės.odt">
    <Relationship full-path="priedai/PriedasT1.odp"
         type="http://www.archyvai.lt/adoc/2008/relationships/content/appendix"/>
    <Relationship full-path="META-INF/signatures/signatures-dir.xml"
         type="http://www.archyvai.lt/adoc/2008/relationships/signatures"/>
  </SourcePart>
  <SourcePart full-path="priedai/PriedasT1.odp">
    <Relationship full-path="META-INF/signatures/signatures-dir.xml"
         type="http://www.archyvai.lt/adoc/2008/relationships/signatures"/>
  </SourcePart>
  <SourcePart full-path="pridedami/Pridedama2.adoc">
    <Relationship full-path="META-INF/signatures/signatures-dir.xml"
         type="http://www.archyvai.lt/adoc/2008/relationships/signatures"/>
    <Relationship full-path="Pagrindinis.docx" type="http://test.com/reference"/>
  </SourcePart>

  <SourcePart full-path="metadada/pasirašomi.xml">
    <Relationship full-path="META-INF/signatures/signatures1.xml"
         type="http://www.archyvai.lt/adoc/2008/relationships/signature">
         <Element in-source-part="true" ref-id="parasas-signatures1" />
    </Relationship>
    <Relationship full-path="META-INF/signatures/signatures-dir.xml"
         type="http://www.archyvai.lt/adoc/2008/relationships/signature">
         <Element in-source-part="true" ref-id="pavadinimas" />
         <Element in-source-part="true" ref-id="sudarytojai" />
```

```
        <Element in-source-part="true" ref-id="parasas-signatures-dir" />
    </Relationship>
    <Relationship full-path="META-INF/signatures/signatures-reg.xml"
        type="http://www.archyvai.lt/adoc/2008/relationships/signature">
        <Element in-source-part="true" ref-id="registravimas" />
        <Element in-source-part="true" ref-id="parasas-signatures-reg" />
    </Relationship>
    <Relationship full-path="META-INF/signatures/signatures-greg.xml"
        type="http://www.archyvai.lt/adoc/2008/relationships/signature">
        <Element in-source-part="true" ref-id="gauto-dokumento-registravimas" />
        <Element in-source-part="true" ref-id="parasas-signatures-greg" />
    </Relationship>
  </SourcePart>

  <SourcePart full-path="META-INF/signatures/signatures-dir.xml">
    <Relationship full-path="META-INF/signatures/signatures-reg.xml"
        type="http://www.archyvai.lt/adoc/2008/relationships/signatures"/>
    <Relationship full-path="META-INF/signatures/signatures-greg.xml"
        type="http://www.archyvai.lt/adoc/2008/relationships/signatures"/>
  </SourcePart>

  <SourcePart full-path="META-INF/signatures/signatures-reg.xml">
    <Relationship full-path="META-INF/signatures/signatures-greg.xml"
        type="http://www.archyvai.lt/adoc/2008/relationships/signatures"/>
  </SourcePart>
</Relationships>
```

5. The content of the manifest file *META-INF/manifest.xml* may be as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<manifest:manifest
xmlns:manifest="urn:oasis:names:tc:opendocument:xmlns:manifest:1.0">
 <manifest:file-entry manifest:full-path="/" manifest:media-
type="application/vnd.lt.archyvai.adoc-2008" />
 <manifest:file-entry manifest:full-path="Pagrindinis.docx" manifest:media-
type="application/vnd.openxmlformats-officedocument.wordprocessingml.document" />
 <manifest:file-entry manifest:full-path="priedai/" manifest:media-type="" />
 <manifest:file-entry manifest:full-path="priedai/PriedasA.xlsx" manifest:media-
type="application/vnd.openxmlformats-officedocument.spreadsheetml.sheet" />
 <manifest:file-entry manifest:full-path="priedai/Taisyklės.odt" manifest:media-
type="application/vnd.oasis.opendocument.text" />
 <manifest:file-entry manifest:full-path="priedai/PriedasT1.odp" manifest:media-
type="application/vnd.oasis.opendocument.presentation" />
 <manifest:file-entry manifest:full-path="pridedami/" manifest:media-type="" />
 <manifest:file-entry manifest:full-path="pridedami/Pridedama2.adoc" manifest:media-
type="application/vnd.lt.archyvai.adoc-2008" />
 <manifest:file-entry manifest:full-path="metadata/" manifest:media-
type="application/vnd.lt.archyvai.adoc-2008#metadata-folder" />
 <manifest:file-entry manifest:full-path="metadata/pasirašomi.xml" manifest:media-
type="text/xml" />
 <manifest:file-entry manifest:full-path="metadata/istorija.xml" manifest:media-
type="text/xml" />
 <manifest:file-entry manifest:full-path="META-INF/" manifest:media-type="" />
 <manifest:file-entry manifest:full-path="META-INF/signatures/" manifest:media-
type="application/vnd.lt.archyvai.adoc-2008#signatures-folder" />
 <manifest:file-entry manifest:full-path="META-INF/signatures/signatures1.xml"
manifest:media-type="text/xml" />
 <manifest:file-entry manifest:full-path="META-INF/signatures/signatures-dir.xml"
manifest:media-type="text/xml" />
<manifest:file-entry manifest:full-path="META-INF/signatures/signatures-reg.xml"
manifest:media-type="text/xml" />
<manifest:file-entry manifest:full-path="META-INF/signatures/signatures-greg.xml"
manifest:media-type="text/xml" />
 <manifest:file-entry manifest:full-path="META-INF/relations.xml" manifest:media-
type="text/xml" />
</manifest:manifest>
```

AN EXAMPLE OF SIGNING OF SEPARATE METADATA ELEMENTS

This example shows how to create the reference element of e-signature file (<*ds:Reference*>) for referring the signable XML element (with attribute *ID* value "*viza_1*") stored in the file of signable metadata *metadata/signableMetadata.xml*:

```
<Reference URI="metadata/signableMetadata.xml">
  <Transforms>
    <Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
      <XPath>ancestor-or-self::*[@ID='viza_1']</XPath>
    </Transform>
    <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
  </Transforms>
  <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
  <DigestValue>mk0/wOnnwdzZ6uKvftiH1fJFNDY=</DigestValue>
</Reference>
```

Specification ADOC-V1.0 of the electronic document signed by the electronic signature
Appendix 17

XML SCHEMES

I. XML SCHEMES FOR METADATA FILES

XML scheme for signable metadata files

1. The XML scheme for signable metadata files is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.archyvai.lt/adoc/2008/metadata/signable"
           xmlns="http://www.archyvai.lt/adoc/2008/metadata/signable"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           elementFormDefault="qualified" attributeFormDefault="unqualified">

  <xs:element name="metadata" type="MetadataType" />
  <xs:complexType name="MetadataType">
    <xs:complexContent>
      <xs:extension base="SignableElementType">
        <xs:all>
          <xs:element name="document" minOccurs="0" maxOccurs="1" type="DocumentType" />
          <xs:element name="authors" minOccurs="0" maxOccurs="1" type="AuthorsType" />
          <xs:element name="creation" minOccurs="0" maxOccurs="1" type="EventType" />
          <xs:element name="recipients" minOccurs="0" maxOccurs="1"
type="RecipientsType" />
          <xs:element name="restrictions" minOccurs="0" maxOccurs="1"
type="RestrictionsType" />
          <xs:element name="registrations" minOccurs="0" maxOccurs="1"
type="RegistrationsType" />
          <xs:element name="receptions" minOccurs="0" maxOccurs="1"
type="ReceptionsType" />
          <xs:element name="signatures" minOccurs="0" maxOccurs="1"
type="SignaturesType" />
          <xs:element name="original_signatures" minOccurs="0" maxOccurs="1"
type="SignaturesType" />
          <xs:element name="Custom" minOccurs="0" maxOccurs="1" type="AnyType" />
        </xs:all>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="DocumentType">
    <xs:complexContent>
      <xs:extension base="SignableElementType">
        <xs:sequence>
          <xs:element name="title" type="xs:string" />
          <xs:element name="sort" minOccurs="0" maxOccurs="1" type="xs:string" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AuthorsType">
    <xs:complexContent>
      <xs:extension base="SignableElementType">
        <xs:sequence>
          <xs:element name="author" minOccurs="1" maxOccurs="unbounded"
type="AddresseeTypeRequired" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
```

```
  <xs:complexType name="AddresseeTypeRequired">
    <xs:complexContent>
      <xs:extension base="SignableElementType">
        <xs:sequence>
          <xs:element name="name" type="xs:string" />
          <xs:element name="code" minOccurs="0" maxOccurs="1" type="xs:string" />
          <xs:element name="address" type="xs:string" />
          <xs:element name="individual" minOccurs="0" maxOccurs="1" type="xs:boolean"
/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="RecipientsType">
    <xs:complexContent>
      <xs:extension base="SignableElementType">
        <xs:sequence>
          <xs:element name="recipient" minOccurs="0" maxOccurs="unbounded"
type="AddresseeType" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AddresseeType">
    <xs:complexContent>
      <xs:extension base="SignableElementType">
        <xs:sequence>
          <xs:element name="name" type="xs:string" />
          <xs:element name="code" minOccurs="0" maxOccurs="1" type="xs:string" />
          <xs:element name="address" minOccurs="0" maxOccurs="1" type="xs:string" />
          <xs:element name="individual" minOccurs="0" maxOccurs="1" type="xs:boolean"
/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="EventType">
    <xs:complexContent>
      <xs:extension base="SignableElementType">
        <xs:sequence>
          <xs:element name="date" minOccurs="0" maxOccurs="1" type="DateType" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="RegistrationsType">
    <xs:complexContent>
      <xs:extension base="SignableElementType">
        <xs:sequence>
          <xs:element name="registration" minOccurs="0" maxOccurs="unbounded"
type="RegistrationType" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="RegistrationType">
    <xs:complexContent>
      <xs:extension base="SignableElementType">
        <xs:sequence>
          <xs:element name="date" type="DateType" />
          <xs:element name="number" type="xs:string" />
          <xs:element name="registrar" minOccurs="0" maxOccurs="1" type="OfficerType"
/>
          <xs:element name="code" minOccurs="0" maxOccurs="1" type="xs:string" />
        </xs:sequence>
```

```
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="ReceptionsType">
    <xs:complexContent>
      <xs:extension base="SignableElementType">
        <xs:sequence>
          <xs:element name="reception" minOccurs="0" maxOccurs="unbounded"
type="ReceptionType" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="ReceptionType">
    <xs:complexContent>
      <xs:extension base="SignableElementType">
        <xs:sequence>
          <xs:element name="date" type="DateType" />
          <xs:element name="number" type="xs:string" />
          <xs:element name="registrar" minOccurs="0" maxOccurs="1" type="OfficerType"
/>
          <xs:element name="receiver" type="AddresseeType" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="RestrictionsType">
    <xs:complexContent>
      <xs:extension base="SignableElementType">
        <xs:sequence>
          <xs:element name="restriction" minOccurs="0" maxOccurs="unbounded"
type="RestrictionType" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="RestrictionType">
    <xs:complexContent>
      <xs:extension base="EventType">
        <xs:sequence>
          <xs:element name="reason" minOccurs="0" maxOccurs="1" type="xs:string" />
          <xs:element name="contentRestriction" minOccurs="0" maxOccurs="1"
type="xs:boolean" />
          <xs:element name="metadataRestriction" minOccurs="0" maxOccurs="1"
type="xs:boolean" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="SignaturesType">
    <xs:sequence>
       <xs:element name="signature" minOccurs="0" maxOccurs="unbounded"
type="SignatureType" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="SignatureType">
    <xs:complexContent>
      <xs:extension base="SignableElementType">
        <xs:sequence>
          <xs:element name="signatureID"    type="xs:anyURI" />
          <xs:element name="signingTime"    type="DateType" />
          <xs:element name="signingPurpose" type="SigningPurposes" />
          <xs:element name="signer"         type="OfficerType" />
        </xs:sequence>
```

```
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:simpleType name="SigningPurposes">
    <xs:restriction base="xs:string">
      <xs:enumeration value="signature" />
      <xs:enumeration value="confirmation" />
      <xs:enumeration value="visa" />
      <xs:enumeration value="conciliation" />
      <xs:enumeration value="acknowledgement" />
      <xs:enumeration value="registration" />
      <xs:enumeration value="registration-of-incomming-documents" />
      <xs:enumeration value="notarisation" />
      <xs:enumeration value="copy-certification" />
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="OfficerType">
    <xs:sequence>
      <xs:element name="individualName" type="xs:string" />
      <xs:element name="positionName" minOccurs="0" maxOccurs="1" type="xs:string" />
      <xs:element name="structuralSubdivision" minOccurs="0" maxOccurs="1"
type="xs:string" />
    </xs:sequence>
    <xs:attribute name="ID" type="xs:ID" use="optional" />
  </xs:complexType>

  <xs:complexType name="SignableElementType" abstract="true">
    <xs:sequence/>
    <xs:attribute name="ID" type="xs:ID" use="required" />
  </xs:complexType>

  <xs:simpleType name="dateTimeWithTimeZone">
    <xs:restriction base="xs:dateTime">
      <xs:pattern value=".+T.+(Z|[+\-].+)"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="dateWithTimeZone">
    <xs:restriction base="xs:date">
      <xs:pattern value=".+[:Z].*"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="DateType">
    <xs:union memberTypes="dateTimeWithTimeZone dateWithTimeZone"/>
  </xs:simpleType>

  <xs:complexType name="AnyType" mixed="true">
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:any namespace="##any" processContents="lax"/>
    </xs:sequence>
    <xs:attribute name="ID" type="xs:ID" use="required" />
    <xs:anyAttribute namespace="##any"/>
  </xs:complexType>
</xs:schema>
```

XML schemes of unsignable metadata files

2. The XML scheme of unsignable metadata files (except for the relationships file for the manifest file) is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.archyvai.lt/adoc/2008/metadata/unsignable"
           xmlns="http://www.archyvai.lt/adoc/2008/metadata/unsignable"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           elementFormDefault="qualified" attributeFormDefault="unqualified">
```

```
  <xs:element name="metadata" type="MetadataType" />

  <xs:complexType name="MetadataType">
    <xs:complexContent>
      <xs:extension base="AbstractElementType">
        <xs:all>
          <xs:element name="Description" minOccurs="0" maxOccurs="1"
type="DescriptionType" />
          <xs:element name="Location" minOccurs="0" maxOccurs="1" type="LocationType"
/>
          <xs:element name="Agent" minOccurs="0" maxOccurs="1" type="AgentType" />
          <xs:element name="Use" minOccurs="0" maxOccurs="1" type="UseType" />
          <xs:element name="Event_history" minOccurs="0" maxOccurs="1"
type="EventHistoryType" />
          <xs:element name="Custom" minOccurs="0" maxOccurs="1" type="AnyType" />
        </xs:all>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="DescriptionType">
    <xs:complexContent>
      <xs:extension base="AbstractElementType">
        <xs:sequence>
          <xs:element name="appendixes" minOccurs="0" maxOccurs="1"
type="AppendixesType" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AppendixesType">
    <xs:complexContent>
      <xs:extension base="AbstractElementType">
        <xs:sequence>
          <xs:element name="appendix" minOccurs="0" maxOccurs="unbounded"
type="AppendixType" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AppendixType">
    <xs:complexContent>
      <xs:extension base="AbstractElementType">
        <xs:sequence>
          <xs:element name="title" minOccurs="0" maxOccurs="1" type="xs:string" />
          <xs:element name="number" minOccurs="0" maxOccurs="1" type="xs:string" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="LocationType">
    <xs:complexContent>
      <xs:extension base="AbstractElementType">
        <xs:sequence>
          <xs:element name="case_id" minOccurs="0" maxOccurs="unbounded"
type="xs:string" />
          <xs:element name="storage" minOccurs="0" maxOccurs="1" type="xs:string" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AgentType">
    <xs:complexContent>
      <xs:extension base="AbstractElementType">
        <xs:sequence>
```

```
          <xs:element name="responsibilities" minOccurs="0" maxOccurs="1"
type="ResponsibilitiesType" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="ResponsibilitiesType">
    <xs:complexContent>
      <xs:extension base="AbstractElementType">
        <xs:sequence>
          <xs:element name="responsibility" minOccurs="0" maxOccurs="unbounded"
type="ResponsibilityType" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="ResponsibilityType">
    <xs:complexContent>
      <xs:extension base="AbstractElementType">
        <xs:sequence>
          <xs:element name="area" type="ResponsibilityArea" />
          <xs:element name="responsible" type="OfficerType" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="UseType">
    <xs:complexContent>
      <xs:extension base="AbstractElementType">
        <xs:sequence>
          <xs:element name="technical_environment" type="TechnicalEnvironmentType" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="TechnicalEnvironmentType">
    <xs:complexContent>
      <xs:extension base="AbstractElementType">
        <xs:sequence>
          <xs:element name="standardVersion" type="xs:string" />
          <xs:element name="documentCategory" minOccurs="0" maxOccurs="1"
type="DocumentCategories" />
          <xs:element name="generator" minOccurs="0" maxOccurs="1" type="xs:string" />
          <xs:element name="os" minOccurs="0" maxOccurs="1" type="xs:string" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:simpleType name="DocumentCategories">
    <xs:restriction base="xs:string">
      <xs:enumeration value="GeDOC" />
      <xs:enumeration value="GGeDOC" />
      <xs:enumeration value="BeDOC" />
      <xs:enumeration value="CeDOC" />
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="EventHistoryType">
    <xs:complexContent>
      <xs:extension base="AbstractElementType">
       <xs:choice maxOccurs="unbounded">
         <xs:element name="resolution" type="ResolutionEventType" />
         <xs:element name="executed" type="ExecutionEventType" />
         <xs:element name="postponed" type="PostponeEventType" />
         <xs:element name="reclassified" type="ReclassificationEventType" />
```

```
          <xs:element name="sent" type="SendingEventType" />
          <xs:element name="moved_from_location" type="RelocationEventType" />
          <xs:element name="changed" type="ChangeEventType" />
          <xs:element name="transformed" type="TransformationEventType" />
          <xs:element name="restored" type="ReasonableEventType" />
          <xs:element name="disposed" type="ReasonableEventType" />
        </xs:choice>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="EventType">
    <xs:complexContent>
      <xs:extension base="AbstractElementType">
        <xs:sequence>
          <xs:element name="date" type="DateType" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="ResponsibleEventType">
    <xs:complexContent>
      <xs:extension base="EventType">
        <xs:sequence>
          <xs:element name="responsible" minOccurs="0" maxOccurs="1"
type="OfficerType" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="ReasonableEventType">
    <xs:complexContent>
      <xs:extension base="ResponsibleEventType">
        <xs:sequence>
          <xs:element name="reason" minOccurs="0" maxOccurs="1" type="xs:string" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="ReclassificationEventType">
    <xs:complexContent>
      <xs:extension base="ReasonableEventType">
        <xs:sequence>
          <xs:element name="case_id" minOccurs="0" maxOccurs="unbounded"
type="xs:string" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="SendingEventType">
    <xs:complexContent>
      <xs:extension base="ReasonableEventType">
        <xs:sequence>
          <xs:element name="sender" minOccurs="0" maxOccurs="1" type="AddresseeType"
/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AddresseeType">
    <xs:complexContent>
      <xs:extension base="AbstractElementType">
        <xs:sequence>
          <xs:element name="name" type="xs:string" />
          <xs:element name="code" minOccurs="0" maxOccurs="1" type="xs:string" />
```

```
          <xs:element name="address" minOccurs="0" maxOccurs="1" type="xs:string" />
          <xs:element name="individual" minOccurs="0" maxOccurs="1" type="xs:boolean"
/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="RelocationEventType">
    <xs:complexContent>
      <xs:extension base="ReasonableEventType">
        <xs:sequence>
          <xs:element name="storage" minOccurs="0" maxOccurs="1" type="xs:string" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="ChangeEventType">
    <xs:complexContent>
      <xs:extension base="ReasonableEventType">
        <xs:sequence>
          <xs:element name="abstract" minOccurs="0" maxOccurs="1" type="xs:string" />
          <xs:element name="reference" minOccurs="0" maxOccurs="1" type="xs:string" />
          <xs:element name="new_value" minOccurs="0" maxOccurs="1" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="TransformationEventType">
    <xs:complexContent>
      <xs:extension base="ResponsibleEventType">
        <xs:sequence>
          <xs:element name="format" minOccurs="0" maxOccurs="1" type="xs:string" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="ResolutionEventType">
    <xs:complexContent>
      <xs:extension base="EventType">
        <xs:sequence>
          <xs:element name="author" minOccurs="0" maxOccurs="1" type="OfficerType" />
          <xs:element name="text" minOccurs="0" maxOccurs="1" type="xs:string" />
          <xs:element name="executors" minOccurs="0" maxOccurs="1"
type="ExecutorsType" />
          <xs:element name="due_by" minOccurs="0" maxOccurs="1" type="DateType" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="ExecutorsType">
    <xs:complexContent>
      <xs:extension base="AbstractElementType">
        <xs:sequence>
          <xs:element name="executor" minOccurs="0" maxOccurs="unbounded"
type="OfficerType" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="ExecutionEventType">
    <xs:complexContent>
      <xs:extension base="ResponsibleEventType">
        <xs:sequence>
          <xs:element name="abstract" minOccurs="0" maxOccurs="1" type="xs:string" />
```

```
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="PostponeEventType">
    <xs:complexContent>
      <xs:extension base="EventType">
        <xs:sequence>
          <xs:element name="reference" minOccurs="0" maxOccurs="1" type="xs:string" />
          <xs:element name="due_by" minOccurs="0" maxOccurs="1" type="DateType" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:simpleType name="ResponsibilityArea">
    <xs:restriction base="xs:string">
      <xs:enumeration value="creation" />
      <xs:enumeration value="management" />
      <xs:enumeration value="relocation" />
      <xs:enumeration value="storage" />
      <xs:enumeration value="deletion" />
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="OfficerType">
    <xs:complexContent>
      <xs:extension base="AbstractElementType">
        <xs:sequence>
          <xs:element name="individualName" type="xs:string" />
          <xs:element name="positionName" minOccurs="0" maxOccurs="1" type="xs:string"
/>
          <xs:element name="structuralSubdivision" minOccurs="0" maxOccurs="1"
type="xs:string" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="AbstractElementType" abstract="true">
    <xs:sequence/>
    <xs:attribute name="ID" type="xs:ID" use="optional" />
  </xs:complexType>

  <xs:simpleType name="dateTimeWithTimeZone">
    <xs:restriction base="xs:dateTime">
      <xs:pattern value=".+T.+(Z|[+\-].+)"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="dateWithTimeZone">
    <xs:restriction base="xs:date">
      <xs:pattern value=".+[:Z].*"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="DateType">
    <xs:union memberTypes="dateTimeWithTimeZone dateWithTimeZone"/>
  </xs:simpleType>

  <xs:complexType name="AnyType" mixed="true">
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:any namespace="##any" processContents="lax"/>
    </xs:sequence>
    <xs:attribute name="ID" type="xs:ID" use="optional" />
    <xs:anyAttribute namespace="##any"/>
  </xs:complexType>

</xs:schema>
```

3. The XML scheme for the relationships description file *relations.xml* is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.archyvai.lt/adoc/2008/relationships"
xmlns="http://www.archyvai.lt/adoc/2008/relationships"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" elementFormDefault="qualified">

  <xs:element name="Relationships" type="RelationshipsType" />
  <xs:complexType name="RelationshipsType">
    <xs:sequence>
      <xs:element ref="SourcePart" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:element name="SourcePart" type="SourcePartType" />
  <xs:complexType name="SourcePartType">
    <xs:sequence>
      <xs:element ref="Relationship" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="full-path" type="xs:anyURI" use="required"/>
  </xs:complexType>

  <xs:element name="Relationship" type="RelationshipType" />
  <xs:complexType name="RelationshipType">
    <xs:sequence>
      <xs:element ref="Element" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="full-path" type="xs:anyURI" use="required"/>
    <xs:attribute name="type"      type="xs:anyURI" use="required"/>
    <xs:attribute name="id"        type="xs:ID"     use="optional"/>
  </xs:complexType>

  <xs:element name="Element" type="ElementType" />
  <xs:complexType name="ElementType">
    <xs:attribute name="in-source-part" type="xs:boolean" use="required"/>
    <xs:attribute name="ref-id"         type="xs:NCName"  use="required"/>
  </xs:complexType>
</xs:schema>
```

4. The XML scheme for the manifest file *manifest.xml* is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:oasis:names:tc:opendocument:xmlns:manifest:1.0"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:manifest="urn:oasis:names:tc:opendocument:xmlns:manifest:1.0"
attributeFormDefault="qualified" elementFormDefault="qualified">

    <xs:element name="manifest" type="manifest:CT_manifest"/>

    <xs:complexType name="CT_manifest">
         <xs:sequence minOccurs="1" maxOccurs="unbounded">
             <xs:element name="file-entry" type="manifest:CT_fileentry"/>
         </xs:sequence>
     </xs:complexType>

    <xs:complexType name="CT_fileentry">
        <xs:attribute name="full-path" type="manifest:ST_attribs"></xs:attribute>
        <xs:attribute name="media-type" type="xs:anyURI"></xs:attribute>
    </xs:complexType>

    <xs:simpleType name="ST_attribs">
        <xs:restriction base="xs:string">
            <xs:minLength value="1"/>
        </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

II. XML FILES FOR THE VERIFICATION OF METADATA CONFORMITY WITH THE REQUIREMENTS

5. These XML files are stored together with other XML schemes of metadata and may be used as an additional verification material for complementing XML schemes. The verification of metadata conformity with the requirements should be performed so as if all the metadata elements of the same XML namespace actually stored in different metadata files belong to the same metadata file. Four XML files are present – one for the verification e-documents of each of four electronic document categories (GeDOC, GGeDOC, BeDOC, CeDOC).

6. The main rule should be observed while using these XML files: all metadata are not mandatory, repetitive and not required to be signed (not required to be protected against change) by e-signature except those metadata that are defined in these XML files.

7. Rules are defined in XML files for each metadata element listed:

Attribute and its value	Verification rule
`mandatory="true"`	Metadata element should be present in e-document and each repetitive parental element of metadata (if metadata has repetitions).
`mandatory-when="after-reception"`	Metadata element is mandatory (rule `mandatory="true"` is applied) only if document is registered in institution or entity that received the e-document;
`mandatory-when="not-individual"`	Metadata element is mandatory (rule `mandatory="true"` is applied) only if metadata element does not belong to the group of elements for identification of a natural person (i.e. metadata element is not mandatory if elements group to which element belongs includes an element "individual" with the value "true");
`single="true"`	Metadata element should not be repeated in the document (at most one or none metadata element should exist);
`must-sign="true"`	Metadata element (including all its repetitions) should be signed (protected against changes) by at least one e-signature.

8. XML file for the verification of e-documents of GeDOC category is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<metadataProfile xmlns="http://www.archyvai.lt/adoc/2008/metadata/profile"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.archyvai.lt/adoc/2008/metadata/profile Profile.xsd ">
  <target>
    <standardVersion>ADOC-V1.0</standardVersion>
    <documentCategory>GeDOC</documentCategory>
  </target>
  <namespace prefix="" url="http://www.archyvai.lt/adoc/2008/metadata/signable"/>
  <namespace prefix="u" url="http://www.archyvai.lt/adoc/2008/metadata/unsignable"/>
  <properties>
    <property name="document/title" mandatory="true" single="true" must-sign="true" />
    <property name="document/sort" mandatory="false" single="true" must-sign="true" />
    <property name="authors/author/name" mandatory="true" must-sign="true" />
    <property name="authors/author/code" mandatory="true" mandatory-when="not-
individual" must-sign="true" />
    <property name="authors/author/address" mandatory="true" must-sign="true" />
    <property name="authors/author/individual" mandatory="true" must-sign="true" />
    <property name="creation/date" mandatory="false" single="true" must-sign="true" />
    <property name="recipients/recipient/name" mandatory="false" must-sign="true" />
    <property name="recipients/recipient/code" mandatory="false" must-sign="true" />

    <property name="restrictions/restriction/contentRestriction" mandatory="false"
must-sign="true" />
    <property name="restrictions/restriction/metadataRestriction" mandatory="false"
must-sign="true" />
```

```
    <property name="registrations/registration/date" mandatory="true" must-sign="true"
/>
    <property name="registrations/registration/number" mandatory="true" must-
sign="true" />
    <property name="receptions/reception/date" mandatory="false" must-sign="true" />
    <property name="receptions/reception/number" mandatory="false" must-sign="true" />
    <property name="receptions/reception/receiver/name" mandatory="false" must-
sign="true" />
    <property name="receptions/reception/receiver/code" mandatory="false" must-
sign="true" />

    <property name="signatures/signature/signatureID" mandatory="true" must-
sign="true" />
    <property name="signatures/signature/signingTime" mandatory="true" must-
sign="true" />
    <property name="signatures/signature/signingPurpose" mandatory="true" must-
sign="true" />
    <property name="signatures/signature/signer/individualName" mandatory="true" must-
sign="true" />
    <property name="signatures/signature/signer/positionName" mandatory="true" must-
sign="true" />
    <property name="original_signatures/signature/signingTime" must-sign="true" />
    <property name="original_signatures/signature/signingPurpose" must-sign="true" />
    <property name="original_signatures/signature/signer/individualName" must-
sign="true" />
    <property name="original_signatures/signature/signer/positionName" must-
sign="true" />

    <property name="Use/technical_environment/standardVersion" namespace="u"
mandatory="true" single="true" />
    <property name="Use/technical_environment/documentCategory" namespace="u"
mandatory="false" single="true" />
    <property name="Use/technical_environment/generator" namespace="u" single="true"
/>
    <property name="Use/technical_environment/os" namespace="u" single="true" />
    <property name="Location/case_id" namespace="u" mandatory="true" />
    <property name="Location/storage" namespace="u" single="true" />
  </properties>
</metadataProfile>
```

9. XML file for the verification of e-documents of GGeDOC category is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<metadataProfile xmlns="http://www.archyvai.lt/adoc/2008/metadata/profile"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.archyvai.lt/adoc/2008/metadata/profile Profile.xsd ">
  <target>
    <standardVersion>ADOC-V1.0</standardVersion>
    <documentCategory>GGeDOC</documentCategory>
  </target>
  <namespace prefix="" url="http://www.archyvai.lt/adoc/2008/metadata/signable"/>
  <namespace prefix="u" url="http://www.archyvai.lt/adoc/2008/metadata/unsignable"/>
  <properties>
    <property name="document/title" mandatory="true" single="true" must-sign="true" />
    <property name="document/sort" mandatory="false" single="true" must-sign="true" />
    <property name="authors/author/name" mandatory="true" must-sign="true" />
    <property name="authors/author/code" mandatory="true" mandatory-when="not-
individual" must-sign="true" />
    <property name="authors/author/address" mandatory="true" must-sign="true" />
    <property name="authors/author/individual" mandatory="true" must-sign="true" />
    <property name="creation/date" mandatory="false" single="true" must-sign="true" />
    <property name="recipients/recipient/name" mandatory="false" must-sign="true" />
    <property name="recipients/recipient/code" mandatory="false" must-sign="true" />

    <property name="restrictions/restriction/contentRestriction" mandatory="false"
must-sign="true" />
    <property name="restrictions/restriction/metadataRestriction" mandatory="false"
must-sign="true" />
```

```
    <property name="registrations/registration/date" mandatory="false" must-
sign="true" />
    <property name="registrations/registration/number" mandatory="false" must-
sign="true" />
    <property name="receptions/reception/date" mandatory="true" mandatory-when="after-
reception" must-sign="true" />
    <property name="receptions/reception/number" mandatory="true" mandatory-
when="after-reception" must-sign="true" />
    <property name="receptions/reception/receiver/name" mandatory="true" mandatory-
when="after-reception" must-sign="true" />
    <property name="receptions/reception/receiver/code" mandatory="true" mandatory-
when="after-reception" must-sign="true" />

    <property name="signatures/signature/signatureID" mandatory="true" must-
sign="true" />
    <property name="signatures/signature/signingTime" mandatory="true" must-
sign="true" />
    <property name="signatures/signature/signingPurpose" mandatory="true" must-
sign="true" />
    <property name="signatures/signature/signer/individualName" mandatory="true" must-
sign="true" />
    <property name="signatures/signature/signer/positionName" mandatory="true" must-
sign="true" />
    <property name="original_signatures/signature/signingTime" must-sign="true" />
    <property name="original_signatures/signature/signingPurpose" must-sign="true" />
    <property name="original_signatures/signature/signer/individualName" must-
sign="true" />
    <property name="original_signatures/signature/signer/positionName" must-
sign="true" />

    <property name="Use/technical_environment/standardVersion" namespace="u"
mandatory="true" single="true" />
    <property name="Use/technical_environment/documentCategory" namespace="u"
mandatory="true" single="true" />
    <property name="Use/technical_environment/generator" namespace="u" single="true"
/>
    <property name="Use/technical_environment/os" namespace="u" single="true" />
    <property name="Location/case_id" namespace="u" mandatory="true" mandatory-
when="after-reception" />
    <property name="Location/storage" namespace="u" single="true" />
  </properties>
</metadataProfile>
```

10. XML file for the verification of e-documents of BeDOC category is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<metadataProfile xmlns="http://www.archyvai.lt/adoc/2008/metadata/profile"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.archyvai.lt/adoc/2008/metadata/profile Profile.xsd ">
  <target>
    <standardVersion>ADOC-V1.0</standardVersion>
    <documentCategory>BeDOC</documentCategory>
  </target>
  <namespace prefix="" url="http://www.archyvai.lt/adoc/2008/metadata/signable"/>
  <namespace prefix="u" url="http://www.archyvai.lt/adoc/2008/metadata/unsignable"/>
  <properties>
    <property name="document/title" mandatory="true" single="true" must-sign="true" />
    <property name="document/sort" mandatory="false" single="true" must-sign="true" />
    <property name="authors/author/name" mandatory="true" must-sign="true" />
    <property name="authors/author/code" mandatory="true" mandatory-when="not-
individual" must-sign="true" />
    <property name="authors/author/address" mandatory="true" must-sign="true" />
    <property name="authors/author/individual" mandatory="true" must-sign="true" />
    <property name="creation/date" mandatory="false" single="true" must-sign="true" />
    <property name="recipients/recipient/name" mandatory="false" must-sign="true" />
    <property name="recipients/recipient/code" mandatory="false" must-sign="true" />

    <property name="restrictions/restriction/contentRestriction" mandatory="false"
must-sign="true" />
```

```
    <property name="restrictions/restriction/metadataRestriction" mandatory="false"
must-sign="true" />

    <property name="registrations/registration/date" mandatory="false" must-
sign="true" />
    <property name="registrations/registration/number" mandatory="false" must-
sign="true" />
    <property name="receptions/reception/date" mandatory="false" must-sign="true" />
    <property name="receptions/reception/number" mandatory="false" must-sign="true" />
    <property name="receptions/reception/receiver/name" mandatory="false" must-
sign="true" />
    <property name="receptions/reception/receiver/code" mandatory="false" must-
sign="true" />

    <property name="signatures/signature/signatureID" mandatory="true" must-
sign="true" />
    <property name="signatures/signature/signingTime" mandatory="true" must-
sign="true" />
    <property name="signatures/signature/signingPurpose" mandatory="true" must-
sign="true" />
    <property name="signatures/signature/signer/individualName" mandatory="true" must-
sign="true" />
    <property name="signatures/signature/signer/positionName" mandatory="true" must-
sign="true" />
    <property name="original_signatures/signature/signingTime" must-sign="true" />
    <property name="original_signatures/signature/signingPurpose" must-sign="true" />
    <property name="original_signatures/signature/signer/individualName" must-
sign="true" />
    <property name="original_signatures/signature/signer/positionName" must-
sign="true" />

    <property name="Use/technical_environment/standardVersion" namespace="u"
mandatory="true" single="true" />
    <property name="Use/technical_environment/documentCategory" namespace="u"
mandatory="true" single="true" />
    <property name="Use/technical_environment/generator" namespace="u" single="true"
/>
    <property name="Use/technical_environment/os" namespace="u" single="true" />
    <property name="Location/storage" namespace="u" single="true" />
  </properties>
</metadataProfile>
```

11. XML file for the verification of e-documents of CeDOC category is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<metadataProfile xmlns="http://www.archyvai.lt/adoc/2008/metadata/profile"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.archyvai.lt/adoc/2008/metadata/profile Profile.xsd ">
  <target>
    <standardVersion>ADOC-V1.0</standardVersion>
    <documentCategory>CeDOC</documentCategory>
  </target>
  <namespace prefix="" url="http://www.archyvai.lt/adoc/2008/metadata/signable"/>
  <namespace prefix="u" url="http://www.archyvai.lt/adoc/2008/metadata/unsignable"/>
  <properties>
    <property name="document/title" mandatory="false" single="true" must-sign="true"
/>
    <property name="document/sort" mandatory="false" single="true" must-sign="true" />
    <property name="authors/author/name" mandatory="true" must-sign="true" />
    <property name="authors/author/code" mandatory="false" must-sign="true" />
    <property name="authors/author/address" mandatory="true" must-sign="true" />
    <property name="creation/date" mandatory="false" single="true" must-sign="true" />
    <property name="recipients/recipient/name" mandatory="false" must-sign="true" />
    <property name="recipients/recipient/code" mandatory="false" must-sign="true" />

    <property name="restrictions/restriction/contentRestriction" mandatory="false"
must-sign="true" />
    <property name="restrictions/restriction/metadataRestriction" mandatory="false"
must-sign="true" />
```

```
    <property name="receptions/reception/date" mandatory="false" must-sign="true" />
    <property name="receptions/reception/number" mandatory="false" must-sign="true" />
    <property name="receptions/reception/receiver/name" mandatory="false" must-
sign="true" />
    <property name="receptions/reception/receiver/code" mandatory="false" must-
sign="true" />

    <property name="signatures/signature/signatureID" mandatory="true" must-
sign="true" />
    <property name="signatures/signature/signingTime" mandatory="true" must-
sign="true" />
    <property name="signatures/signature/signingPurpose" mandatory="true" must-
sign="true" />
    <property name="signatures/signature/signer/individualName" mandatory="true" must-
sign="true" />
    <property name="original_signatures/signature/signingTime" must-sign="true" />
    <property name="original_signatures/signature/signingPurpose" must-sign="true" />
    <property name="original_signatures/signature/signer/individualName" must-
sign="true" />

    <property name="Use/technical_environment/standardVersion" namespace="u"
mandatory="true" single="true" />
    <property name="Use/technical_environment/documentCategory" namespace="u"
mandatory="true" single="true" />
    <property name="Use/technical_environment/generator" namespace="u" single="true"
/>
    <property name="Use/technical_environment/os" namespace="u" single="true" />
    <property name="Location/storage" namespace="u" single="true" />
  </properties>
</metadataProfile>
```

12. The XML scheme for checking the integrity of XML file for verification of metadata conformity with the requirements:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.archyvai.lt/adoc/2008/metadata/profile"
xmlns="http://www.archyvai.lt/adoc/2008/metadata/profile"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" elementFormDefault="qualified">

  <xs:element name="metadataProfile" type="MetadataProfileType" />

  <xs:complexType name="MetadataProfileType">
    <xs:sequence>
      <xs:element name="target" type="TargetType" nillable="false" />
      <xs:element name="namespace" type="NamespaceType" minOccurs="0"
maxOccurs="unbounded" />
      <xs:element name="properties" type="PropertyList" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="TargetType">
    <xs:sequence>
      <xs:element name="standardVersion" maxOccurs="unbounded" type="xs:string" />
      <xs:element name="documentCategory" maxOccurs="4" type="DocumentCategories" />
    </xs:sequence>
  </xs:complexType>

  <xs:simpleType name="DocumentCategories">
    <xs:restriction base="xs:string">
      <xs:enumeration value="GeDOC" />
      <xs:enumeration value="GGeDOC" />
      <xs:enumeration value="BeDOC" />
      <xs:enumeration value="CeDOC" />
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="NamespaceType">
    <xs:sequence>
```

```
    </xs:sequence>
    <xs:attribute name="prefix" type="xs:string" />
    <xs:attribute name="url" type="xs:string" />
  </xs:complexType>

  <xs:complexType name="PropertyList">
    <xs:sequence>
      <xs:element name="property" type="PropertyType" minOccurs="0"
maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PropertyType">
    <xs:sequence>
    </xs:sequence>
    <xs:attribute name="name" type="xs:string" use="required" />
    <xs:attribute name="namespace" type="xs:string"  use="optional" default="" />
    <xs:attribute name="mandatory" type="xs:boolean" use="optional" default="false" />
     <xs:attribute name="mandatory-when" type="MandatoryConditions"
use="optional" default="" />
    <xs:attribute name="single"    type="xs:boolean" use="optional" default="false" />
    <xs:attribute name="must-sign" type="xs:boolean" use="optional" default="false" />
  </xs:complexType>

  <xs:simpleType name="MandatoryConditions">
    <xs:restriction base="xs:string">
      <xs:enumeration value="" />
      <xs:enumeration value="after-reception" />
      <xs:enumeration value="not-individual" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

III. XML SCHEME FOR E-SIGNATURE FILE

13. XML scheme for of e-signature files is as follows. Signature elements (<*Signature*>) existing in e-signature files are verified according to the XMLDSIG and XAdES XML schemes. However, checking the signature file against the XML scheme provided is not sufficient. Obligations and restrictions set for the signature elements defined in this Specification (Appendix 13) should be performed by using additional software tools.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="urn:oasis:names:tc:opendocument:xmlns:digitalsignature:1.0"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:dsig="urn:oasis:names:tc:opendocument:xmlns:digitalsignature:1.0"
xmlns:xmldsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:xades="http://uri.etsi.org/01903/v1.3.2#" elementFormDefault="qualified">

  <xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-core-
schema.xsd"/>
  <xs:import namespace="http://uri.etsi.org/01903/v1.3.2#"
schemaLocation="http://uri.etsi.org/01903/v1.3.2/XAdES.xsd" />
  <xs:import namespace="http://uri.etsi.org/01903/v1.4.1#"
schemaLocation="http://uri.etsi.org/01903/v1.4.1/XAdESv141.xsd" />

  <xs:element name="document-signatures">
    <xs:complexType>
      <xs:sequence minOccurs="1" maxOccurs="unbounded">
        <xs:element ref="xmldsig:Signature" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Specification ADOC-V1.0 of the electronic document signed by the electronic signature
Appendix 18

STANDARTŲ IR KITŲ DOKUMENTŲ SĄRAŠAS

1. LST CWA 14169:2009 Secure signature-creation devices "EAL 4+".

2. LST CWA 14170:2005 Security requirements for signature creation applications.

3. LST CWA 14171:2005 General guidelines for electronic signature verification.

4. LST ETSI TS 101 861 V1.2.1:2002 Time stamping profile.

5. LST ETSI TS 101 862 V1.3.3:2007 Qualified certificate profile.

6. LST ETSI TS 101 903 V1.3.2:2006 XML Advanced Electronic Signatures (XAdES);
LST ETSI TS 101 903 V1.4.1:2009 XML Advanced Electronic Signatures (XAdES).

7. LST ETSI TS 102 023 v1.2.1:2007 Electronic Signatures and Infrastructures (ESI), Policy requirements for time-stamping authorities

8. LST ISO/IEC 10918-1:2009 Information technology - Digital compression and coding of continuous-tone still images - Part 1: Requirements and guidelines (ISO/IEC 10918-1:1994, identical).

9. LST ISO 12234-2:2008 Electronic still-picture imaging - Removable memory - Part 2: TIFF/EP image data format (ISO 12234-2:2001, identical)

10. LST ISO/IEC 15948:2009 Information technology - Computer graphics and image processing - Portable Network Graphics (PNG): Functional specification (ISO/IEC 15948:2004, identical).

11. LST ISO 19005-1:2008 Document management - Electronic document file format for long-term preservation - Part 1: Use of PDF 1.4 (PDF/A-1) (ISO 19005-1:2005, identical); corrections: LST ISO 19005-1:2008/AC:2008.

12. LST ISO/IEC 26300:2007 Information technology – Open Document Format for Office Applications (OpenDocument) v1.0 (ISO/IEC 26300:2006, identical).

13. LST ISO/IEC 29500-1:2009 Information technology - Document description and processing languages - Office Open XML File Formats - Part 1: Fundamentals and Markup Language Reference (ISO/IEC 29500-1:2008, identical);
Information technology - Document description and processing languages - Office Open XML File Formats - Part 2: Open Packaging Conventions (ISO/IEC 29500-2:2008, identical).
LST ISO/IEC 29500-3:2009 Information technology - Document description and processing languages - Office Open XML File Formats - Part 3: Markup Compatibility and Extensibility (ISO/IEC 29500-3:2008, identical);
LST ISO/IEC 29500-4:2009 Information technology - Document description and processing languages - Office Open XML File Formats - Part 4: Transitional Migration Features (ISO/IEC 29500-4:2008, identical).

14. ISO 32000-1:2008 „Document management. Portable document format. Part 1: PDF 1.7“.

15. RFC 2560 „X.509 Internet Public Key Infrastructure: Online Certificate Status Protocol – OCSP“, http://www.ietf.org/rfc/rfc2560.txt, 1999.

16. RFC 3161 „Internet X.509 Public Key Infrastructure: Time-Stamp Protocol (TSP)“, http://www.ietf.org/rfc/rfc3161.txt, 2001.

17. RFC 3739 „Internet X.509 Public Key Infrastructure: Qualified Certificates Profile“, http://www.ietf.org/rfc/rfc3739.txt, 2004.

18. RFC 3986 „Uniform Resource Identifier (URI): Generic Syntax“, http://www.ietf.org/rfc/rfc3986.txt, 2005.

19. RFC 3987 „Internationalized Resource Identifiers (IRIs)“, http://www.ietf.org/rfc/rfc3987.txt, 2005.

20. RFC 4288 „Media Type Specifications and Registration Procedures“), http://www.ietf.org/rfc/rfc4288.txt, 2005.

21. RFC 5280 „Internet X.509 Public Key Infrastructure: Certificate and Certificate Revocation List (CRL) Profile“), http://www.ietf.org/rfc/rfc5280.txt, 2008.

22. W3C „XML Signature Syntax and Processing (Second Edition)“, http://www.w3.org/TR/2008/REC-xmldsig-core-20080610/, 2008.

23. *„Info-ZIP Application* Note, *19970311“*, ftp://ftp.uu.net/pub/archiving/zip/doc/appnote-970311-iz.zip, 1997.
