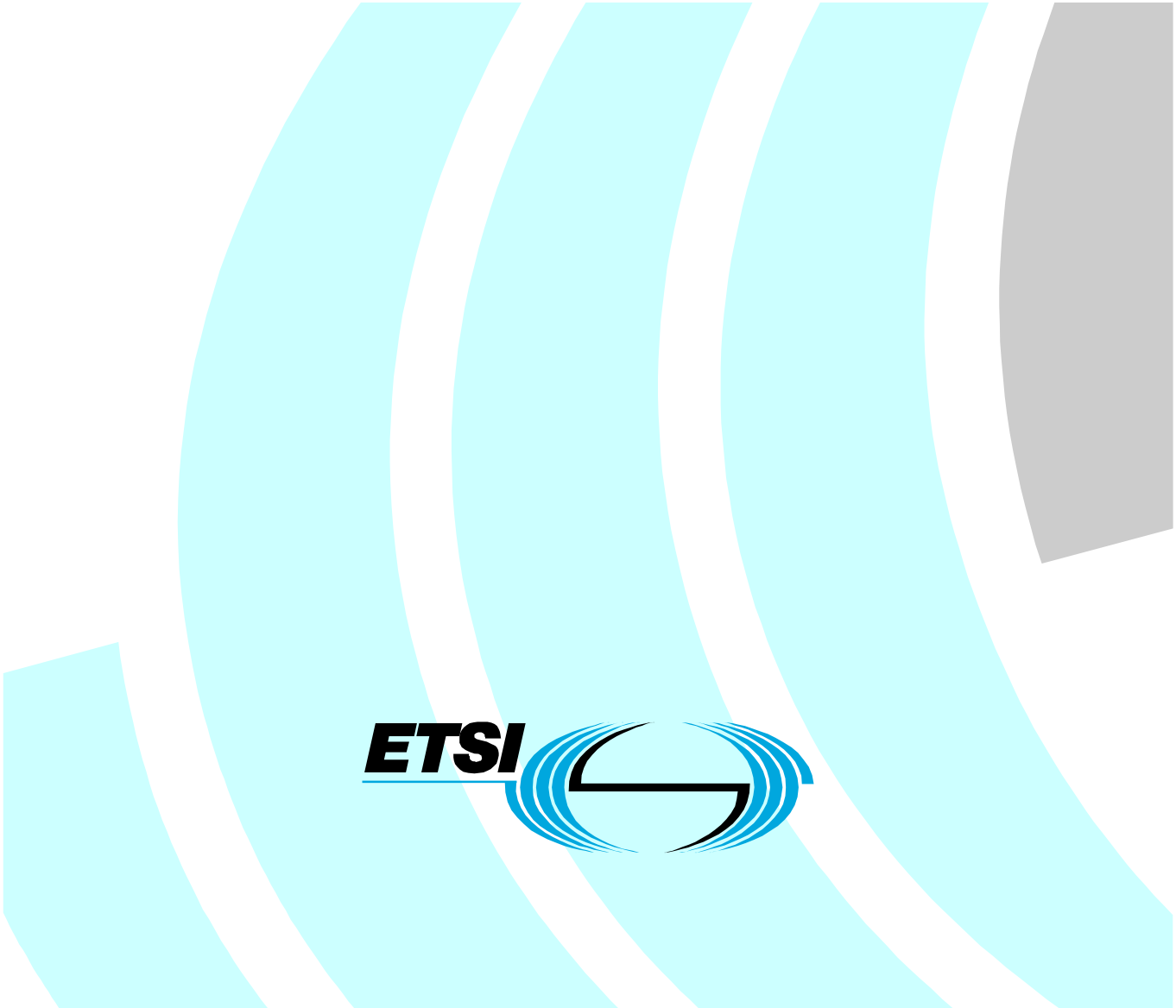
ETSI TS 102 231 V2.1.1 (2006-03)

Technical Specification

Electronic Signatures and Infrastructures (ESI); Provision of harmonized Trust-service status information



Reference
RTS/ESI-000038

Keywords
e-commerce, electronic signature, security, trust services

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Content

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Electronic Signatures and Infrastructures (ESI).

Acknowledgement

The building blocks of the Localization, Access and Authentication Mechanisms described in this Technical Specification were submitted by UNINFO, the Italian standardization body for ICT, federated to UNI, Italian member body of CEN and ISO.

Introduction

The purpose of a Trust-service Status List (TSL), and hence of the present document, is to provide a harmonized way in which assessment schemes having an oversight role with regards to trust services and their providers (trust service providers - TSPs) can publish information about the services and TSPs which they currently oversee, or indeed (through the provision of historical information) have overseen. Assessment schemes may also use the TSL to refer to other assessment schemes, in which case they would be represented as a special form of trust service.

The present document is based upon the reasoning that it will enhance the confidence of parties relying on certificates or other services related to electronic signatures if they had access to information that would allow them to know whether a given TSP was operating under the approval of any recognized scheme at the time of providing their services and of any dependent transaction that took place.

The assurance provided by information available within a TSL is intended to serve as a secondary source of trust, rather than a primary source of trust which might be derived by parsing a certificate chain. The present document is not intended to be a replacement for certificate chains and the assurance which may be obtained from parsing them to establish the validity of certificates (or other forms of trust service tokens) associated with providers of trust services of any kind.

The information should be available for a wide range of services and schemes, including the use of Qualified Certificates. The importance of this information is especially significant for cross-domain and international transactions. This information should preferably be accessible using an on-line protocol, although accessibility both off-line and on-line should be possible.

Entities having such an oversight role could be supervisory systems or voluntary approval schemes as defined in Directive 1999/93/EC [1] similar schemes established by other sovereign states or economies (e.g. certain government e-authentication frameworks), and those established by specific industry sectors or for international promotion of trust services.

All previous versions of this document (as listed below) are to be considered as "historical", with effect from the publication date of this present version. Although there may remain in existence for some time TSLs which were created compliant to previous versions all future TSL's published should be conformant to the specifications set out in the present document. Parsers should be upgraded to accommodate the version defined herein whilst retaining their ability to parse previous versions where they continue to be used.

This version renders **historical** these previous versions:

- Version 1.1.1, downloadable from ETSI as file "ts_102231v010101p".

1 Scope

The present document specifies a standard for a Trust-service Status List (TSL) which makes available trust service status information such that interested parties may determine whether a trust service is **or was** operating under the approval of any recognized scheme at either the time the service was provided, or **the time at which a transaction reliant on that service took place**.

The normative specification defines the structure and meaning of a TSL which fulfils these requirements and specifies the mechanisms to be used for locating, accessing and authenticating TSLs. In addition, this document gives informative guidance for the management of and access to TSLs and the use of status information held within them. Within the present document the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in RFC 2119 [5].

The present document is applicable to assessment scheme operators responsible for the approval of trust services and to those who wish to rely on such information.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

[1] Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures.

[2] ETSI TS 101 733: "Electronic Signatures and Infrastructures (ESI); CMS Advanced Electronic Signatures (CAdES)".

[3] IETF RFC 959: "File Transfer Protocol (FTP)".

[4] IETF RFC 2045: "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies".

[5] IETF RFC 2119: "Key words for use in RFCs to indicate Requirement Levels".

[6] IETF RFC 2141: "URN Syntax".

[7] IETF RFC 2251: "Lightweight Directory Access Protocol (v3)".

[8] IETF RFC 2252: "Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions".

[9] IETF RFC 2253: "Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names".

[10] IETF RFC 2256: "A Summary of the X.500(96) User Schema for use with LDAPv3".

[11] IETF RFC 2368: "The mailto URL scheme".

[12] IETF RFC 2616: "Hypertext Transfer Protocol - HTTP/1.1".

[13] IETF RFC 2634: "Enhanced Security Services for S/MIME".

[14] IETF RFC 2822: "Internet Message Format".

[15] IETF RFC 3023: "XML Media Types".

[16] IETF RFC 3066: "Tags for the Identification of Languages".

[17] IETF RFC 3280: "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile".

[18] IETF RFC 3305: "Report from the Joint W3C/IETF URI Planning Interest Group: Uniform Resource Identifiers (URIs), URLs, and Uniform Resource Names (URNs): Clarifications and Recommendations".

[19] IETF RFC 3986: "Uniform Resource Identifiers (URI): Generic Syntax".

[20] IETF RFC 4050: "Using the Elliptic Curve Signature Algorithm (ECDSA) for XML Digital Signatures".

[21] ISO 3166-1: "Codes for the representation of names of countries and their subdivisions - Part 1: Country codes".

[22] ISO 8601: "Data elements and interchange formats - Information interchange - Representation of dates and times".

[23] ISO 10646: "Information technology - Universal Multiple-Octet Coded Character Set (UCS)".

[24] ITU-T Recommendation X.208: "Specification of Abstract Syntax Notation One (ASN.1)".

[25] ITU-R Recommendation TF.460-5: "Standard-frequency and time-signal emissions".

[26] ITU-T Recommendation X.509: "Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks".

[27] ITU-T Recommendation X.680: "Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation".

[28] ITU-T Recommendation X.690: "Information Technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)".

[29] W3C Recommendation (2002): "XHTMLTM 1.0 - The Extensible HyperText Markup Language (Second Edition) - A Reformulation of HTML 4 in XML 1.0".

[30] W3C Recommendation (2001): "XHTMLTM 1.1 - Module-based XHTML".

[31] W3C Recommendation (1999): "HTML 4.01 Specification".

[32] W3C Recommendation (2004): "XML Schema Part 2: Data types Second Edition".

[33] W3C Technical Report #20 Revision 7: "Unicode in XML and other Markup Languages".

[34] W3C Recommendation (2002): "XML-Signature Syntax and Processing".

[35] ETSI TS 101 903: "XML Advanced Electronic Signatures (XAdES)".

[36] IETF RFC 4055: "Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile".

[37] IETF RFC 3852: "Cryptographic Message Syntax (CMS)".

3 Definitions, terms and abbreviations

3.1 Definitions

For the purposes of the present document, the following definitions apply:

approval: assertion that a(n electronic trust) service, falling within the oversight of a particular scheme, has been either positively endorsed (active approval) or has received no explicit restriction since the time at which the scheme was aware of the existence of the said service (passive approval)

assessment scheme: any organized process of supervision, monitoring, approval or such practices that are intended to apply oversight with the objective of ensuring adherence to specific criteria in order to maintain confidence in the services under the scope of the scheme

(electronic) Trust Service: service which enhances trust and confidence in electronic transactions (typically but not necessarily using cryptographic techniques or involving confidential material)

Qualified Certificate: public key certificate issued in accordance with the requirements of Directive 1999/93/EC [1]

scheme operator: body responsible for the operation and/or management of any kind of scheme, whether they be governmental, industry or private, etc.

Trust Service Provider (TSP): body operating one or more **(electronic) Trust Services**

NOTE: This term is used in preference to and with a broader application than, the term certification-service-provider (CSP) used in Directive 1999/93/EC [1]. Moreover, the term can also refer to other assessment schemes, which the issuer of a TSL may include as trust service providers whose schemes are a specific type of trust service.

Trust Service Token (TrST): a physical or binary (logical) object generated or issued as a result of the use of a Trust Service.

NOTE: Examples of binary Trust Service Tokens are: certificates, CRLs, Time Stamp Tokens, OCSP responses. Where the TSP is a scheme the TrSTs are the TSLs it issues. Physical tokens may be devices on which binary objects (tokens or credentials) are stored. Equally, a token may be the performance of an act and the generation of an electronic record, e.g. an insurance policy or share certificate.

3.2 Terms

For the purposes of the present document, the following terms apply:

implementation specific: used throughout the present document and refers principally to the annexes A and B implementation specifications for ASN.1 and XML. It does not mean that implementers of TSL applications have a free choice.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ASN.1	Abstract Syntax Notation One
CA	Certification Authority
CMS	Cryptographic Message Syntax
CRL	Certificate Revocation List
ESS	Enhanced Security Services
EU	European Union
OCSP	Online Certificate Status Protocol
PKC	Public Key Certificate
PKI	Public Key Infrastructure
ToSch	TSL "of Schemes"
TSL	Trust-service Status List

TSP	Trust Service Provider
URI	Uniform Resource Identifier
URN	Uniform Resource Name
UTC	Coordinated Universal Time
WWW	World Wide Web
XML	eXtensible Markup Language

4 Trust-service status information

The present document specifies a standard for the provision of trust service status information and mechanisms for locating, accessing and authenticating that information. In recognition of the selection of a form of signed list as the basis for presentation of this information, the term Trust-service Status List (TSL) is adopted. Each assessment scheme (scheme operator) which maintains a TSL in accordance with the present document MUST comply with the format and semantics specified in clause 5. Each such assessment scheme MUST operate against specific criteria for determining the status of trust services which it recognizes: an assessment scheme operator could, therefore, operate more than one discrete scheme, according to different criteria it might apply for different purposes.

With regard to the information provided within a TSL, it should be noted that the present document addresses only the type, format and meaning of information which MAY be presented in a TSL and does not define how that information should be sourced, i.e. what steps the scheme operator takes to collect that information. Nor does it specify the criteria which assessment schemes should use to determine the status of any trust services falling within their remit - such criteria remain the responsibility of the scheme operators. Furthermore, it does not specify how any status or scheme-related information should be presented outside the context of a TSL, e.g. on schemes' websites.

It should also be stressed that the information which is available within a TSL is intended to serve as a secondary source of trust, rather than a primary source of trust which might be derived by parsing a certificate chain. The present document is not intended to be a replacement for certificate chains and the assurance which may be obtained from parsing them to establish the validity of certificates (or other forms of trust service tokens) associated with providers of trust services of any kind.

Each assessment scheme adopting this TSL standard MUST be able to support the provision of status information in each of the following forms:

- Human readable in a format readily down-loadable and printable.
- Machine processable to allow automatic verification of status information.

The TSL specified by the present document enables any interested party to determine whether a trust service is or was operating under the approval of any recognized scheme at either the time the service was provided, or the time at which a transaction reliant on that service took place. In order to fulfil this requirement, Trust-service Status Lists MUST necessarily contain information from which it can be established whether the TSP's service was, at the time of the transaction, known by the assessment scheme operator and if so the status of the service, i.e. whether it was approved, suspended, cancelled, revoked, etc. The Trust-service Status List MUST therefore contain not only the service's current status, but also the history of its status. Because of this requirement upon it, the TSL MUST therefore be specified in a manner which can support both "positive approval" lists and "delinquents" lists, including historical information.

The TSL specified by the present document therefore has four major components, in a structured relationship. These components:

- provide information on the issuing scheme;
- identify the TSPs recognized by the scheme;
- indicate the service(s) provided by these TSPs and the current status of those service(s);
- indicate for each service the status history of that service.

The logic of the list is that, once the assessment scheme operator has become aware of the existence of the TSP (whether by some pro-active action on the part of the TSP or by the scheme's own supervision of the marketplace), the particular status as determined according to the scheme rules is either the present status of the TSP's service (i.e. only current status, no history) or is seamlessly followed by a sequence of one or more statuses (current status and history). Note that if a trust service was approved until a certain date/time and there was a period in between the expiry of the approval and the start of the re-approval, then a status identifier would provide the information for that interim period. The "interim status" would either be expired (i.e. voluntarily, by the TSP) or revoked (by the scheme, with reasons).

5 Trust-service Status List structure

This clause specifies the Trust-service Status List structure. Each of the fields within the TSL is described to a level of detail sufficient to permit any assessment scheme operator to implement a standardized TSL, consistent with any other TSL conformant to the present document, with specified values, meanings and interpretations given for each field. Whether the inclusion of a field is REQUIRED or OPTIONAL is indicated.

5.1 Structure of the Trust-service Status List

The logical model of the Trust-service Status List is shown in figure 1. It has four logical component parts, all but the first of which MAY be replicated as required.

The list commences with key information about the list itself and the nature of the scheme which has determined the information found in, and through, the list (component 1). The specified set of information MUST include a pointer (URI) to details of the scheme and how its operator MAY be contacted. Whilst the objective has been to keep the size of the TSL to the minimum consistent with its purpose and the requirements placed upon it, certain key information which one would expect to be found in the scheme details MUST be provided directly within the TSL itself so as to facilitate either easy recognition and contact with the scheme or machine processing.

Following this scheme-related information there comes information relating to the Trust Service Providers (TSPs) whose services are within the scope of the scheme (component 2), and for each of those TSPs, the details of their specific trust services whose current status is recorded within the TSL (component 3). For each service, any available historical status information is recorded (component 4). The number of TSPs, of services per TSP, and of history sections per service is unbounded.

The TSL is a signed list for authentication purposes and is tagged to facilitate identification for electronic searches. The structure of the TSL is described in the following clauses by each component part and its fields.

Where fields are defined as being of type URI, implementers MAY in future use the URN (a particular subset of URIs that provides with persistent names and whose syntax is specified by RFC 2141 [6]) once such names become technically resolvable. Until such time implementers should use other URI types whose general syntax is specified by RFC 3986 [19]. See RFC 3305 [18] for clarification about URI and URN.

5.1.1 Trust-service Status List information

Description:

This field represents all the structured information and SHALL contain the following:

a) A Trust-service Status List tag to facilitate identification of the TSL for electronic searches. The contents of the tag are specified in clause 5.2.1.

b) Scheme information, as specified in clause 5.2.

c) A sequence of fields holding information on the TSPs that the scheme oversees. This sequence is OPTIONAL. The contents of the TSP information field are specified in clause 5.4.

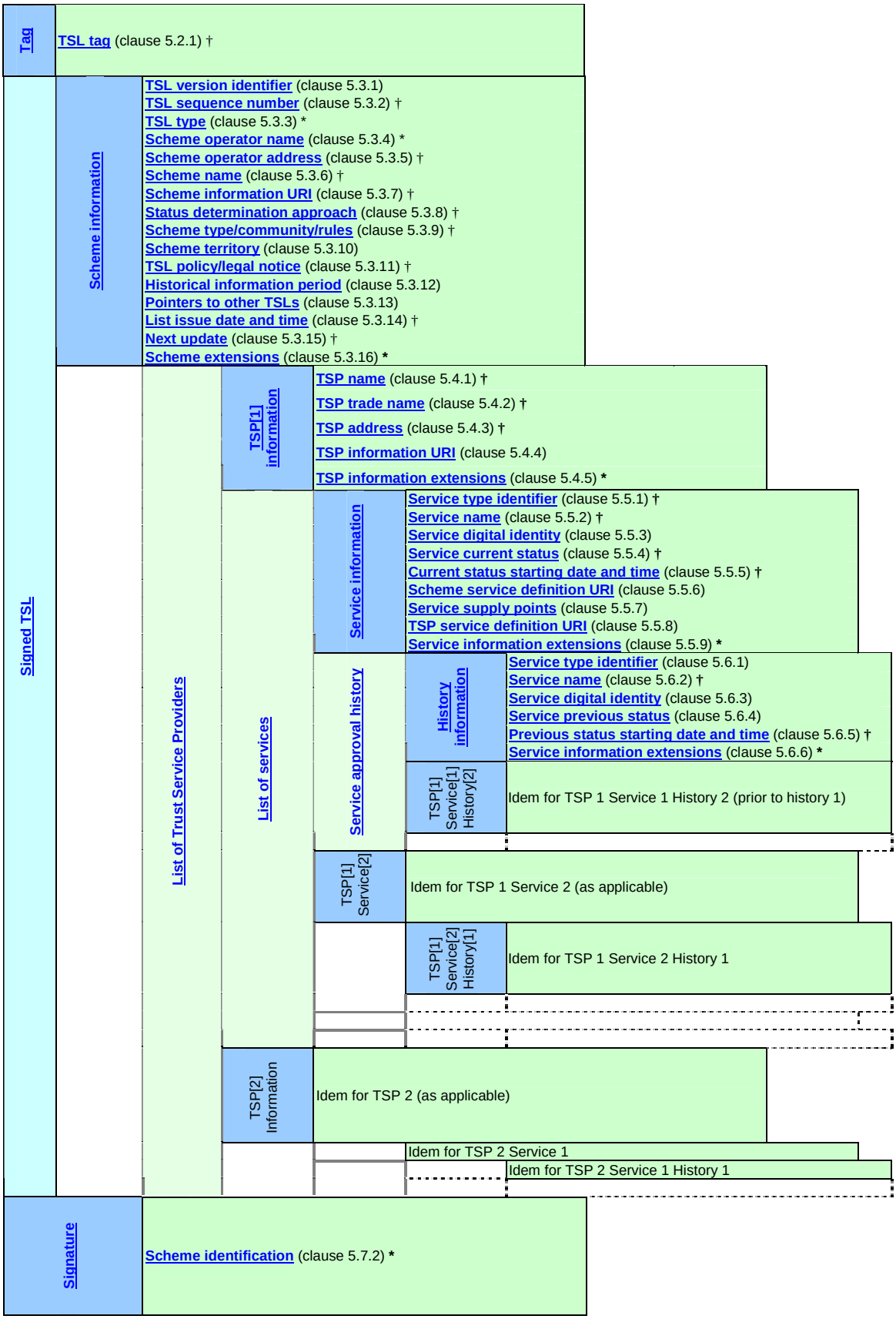
d) For each TSP, a sequence of fields holding information on the service(s) provided by that TSP. This sequence is REQUIRED and MUST have a minimum of one entry. The contents of the service information field are specified in clause 5.5.

e) For each service, a sequence of fields holding information on the status history of that service. This sequence is REQUIRED when the scheme declares that history information is held. The contents of the history information field are specified in clause 5.6.

f) A signature computed over all fields of the TSL except the signature value specified in clause 5.7.4. The contents of the signature field are specified in clause 5.7.

5.1.2 Logical model

Figure 1 should be used as a manual index to the TSL field definitions when using a printed copy of the present document.



"*" indicates the field is new in this version of the specification.
"†" indicates that the field's definition has changed significantly since the previous version of the present document.

Figure 1: Logical model of the TSP Status List

5.1.3 Language support

Trust Status Lists MAY be issued supporting multiple (natural) languages. For all fields, where multiple language versions are possible, the following general rules apply:

1) A **multilingual character string** is an ISO 10646 [23] character string encoded in UTF-8. Each **multilingual character string** consists of two parts: a tag, conformant to RFC 3066 [16], that identifies the language in which the string is expressed, and the text in that language. The same content MAY be represented in multiple languages by a sequence of **multilingual character strings**.

2) A **multilingual pointer** is a URI that identifies a resource expressed in a particular language. Each **multilingual pointer** consists of two parts: a tag, conformant to RFC 3066 [16], that identifies the language in which the content pointed-to by the URI is expressed, and the URI expressed as a character string with the syntax specified by RFC 3986 [19], in the given language. The same content MAY be represented in multiple languages by a sequence of **multilingual pointers**.

Further detail requirements regarding multilingual implementation are given in annex E.

5.1.4 Date-time indication

All fields carrying date-time values SHALL be of the format "YYYYMMDDhhmmssZ" and therefore SHALL comply with the following rules:

1) the date-time values SHALL be a character string formatted according to ISO 8601 [22], without any separators between any components of the date-time;

2) the date-time value SHALL be expressed as "Zulu" (Coordinated Universal Time or UTC) and its value SHALL NOT include fractional seconds. The time scale MUST be based on the second as defined in ITU-R Recommendation TF.460-5 [25].

5.1.5 Use of Uniform Resource Identifiers

In the definitions of TSL fields given in this clause, many use uniform resource identifiers (URIs) to indicate the meaning of the field concerned. Within these definitions a "common name" is used to broadly and simply describe the specific values or meanings of the field. These common names are linked to their declaration in annex D, which formally states all URIs used in the present document, with their meanings.

5.2 Trust-service Status List tag

5.2.1 TSL tag

Description: This field is REQUIRED. The TSL SHALL be tagged to facilitate its identification during electronic searches and also to confirm its purposes when in human-readable form.

Format: A character string which indicates that the data structure is a TSL. This SHALL be the character representation of the TSLtag URI.

Meaning: A unique value enabling a web-searching tool to establish during a WWW-wide search for TSLs that a resource it has located is indeed a TSL. Only the characters required to fully represent the URI SHALL be present.

Back to Logical model.

5.3 Scheme information

5.3.1 TSL version identifier

Description: This field is REQUIRED. It SHALL specify the version of the TSL format.

Format: Integer.

Meaning: The value of the identifier for TSLs conforming to this version of the present document, which SHALL be "**2**".

Note: This field will only be incremented when the rules for parsing the TSL change, e.g. through addition/removal of a field or a change to the values or meaning of an existing field. Revisions to the specification which do not change the parsing rules of the TSL MAY be made without revision to this field -there should be no reliance placed upon the continuing alignment of the TSL version and the specification issue after the initial publication of this document at version 01.01.01 which defined TSL version "**1**".

5.3.2 TSL sequence number

Description: This field is REQUIRED. It SHALL specify the sequence number of the TSL.

Format: Integer.

Meaning: At the first release of the TSL, the value of the sequence number SHALL be 1. The value SHALL be incremented by 1 at each subsequent release of the TSL and SHALL NOT be re-cycled to "1" when the "TSL version identifier" field is incremented.

5.3.3 TSL type (new this version)

Description: This field is REQUIRED. It SHALL specify the type of the TSL.

Format: A TSL type indicator expressed as one of the following URIs:

- Generic;
- Schemes.

Meaning: The quoted URI SHALL be one of those listed in clause D.2, pertaining to this field, or another URI having the same purpose, registered and described by the scheme operator or another entity, such as a community or federation of schemes, a standards body, etc. It SHALL indicate the type of the TSL which will permit a parser to determine which form of any following fields to expect, where those fields have alternative meanings according to the type of TSL represented.

5.3.4 Scheme operator name *(new this version)*

Description: This field is REQUIRED. It SHALL specify the formal name under which the scheme operator does business or is given its mandate (e.g. for governmental administrative agencies).

Format: A sequence of multilingual character strings (see clause 5.1.3).

Meaning: The name of the scheme operator MUST be the name which is used in formal legal registrations or authorizations and to which any formal communication, whether physical or electronic, should be addressed.

Local language and cross-border (international) trading considerations MAY require that this information be provided both in a national language (and script) and in a commonly accepted internationally-used language.

5.3.5 Scheme operator address

Description: This field is REQUIRED. It SHALL specify the address of the legal identity identified in clause 5.3.3, for both postal and electronic communications. Users (subscribers, relying parties) should use this address as the contact point for enquiries, complaints, etc. to the scheme operator.

This is a multi-part field consisting of the scheme operator physical address specified in clause 5.3.5.1 and the scheme operator electronic address specified in clause 5.3.5.2.

5.3.5.1 Scheme operator postal address

Description: This field is REQUIRED. It SHALL specify the postal address of the legal entity identified in clause 5.3.3, with the provision for the inclusion of the address in multiple languages.

Format: A sequence of multilingual character strings (see clause 5.1.3).

Each sequence of character strings SHALL give the following attributes pertaining to the legal entity:

- Street address (sub-components internally delimited by ";");
- Locality (town/city);
- Optionally, if applicable, State or Province name;
- Postal code;
- Country name as a two-character code in accordance with ISO 3166-1 [21].

Meaning: This MUST be a postal address at which the scheme operator provides a regularly-serviced capability for conventional (physical) mail.

5.3.5.2 Scheme operator electronic address

Description: This field is REQUIRED. It SHALL specify the address of the legal entity identified in clause 5.3.3 for electronic communications.

Format: Sequence of character strings giving: e-mail address as a URI, in the form specified by RFC 3986 [19] and with the URI scheme defined in RFC 2368 [11], and; web-site as a URI, in the form specified by RFC 3986 [19].

At least one such character string MUST be present.

Meaning: In the case of an e-mail address, this MUST be an address at which the scheme operator provides a regularly serviced help line capability. In the case of a web-site URI, this MUST lead to a capability whereby the user MAY communicate with a regularly serviced help line capability.

5.3.6 Scheme name

Description: This field is REQUIRED. It SHALL specify the name under which the scheme operates.

Format: A sequence of multilingual character strings (see clause 5.1.3).

Meaning: The name of the scheme MUST be the name which is used in formal references to the scheme in question, and MUST be unique and MUST NOT be used by any other scheme operated by the same entity.

Local language and cross-border (international) trading considerations MAY require that this information be provided both in a national language (and script) and in a commonly accepted internationally-used language.

Note: The scheme name is required to uniquely identify by name the scheme referred to by the "Scheme information URI", and also to ensure that in the event that a scheme operator operates more than one scheme, there is a distinct name given to each of them. Thus if a scheme name is the same as the scheme operator's name that name may only be used for one scheme.

5.3.7 Scheme information URI

Description: This field is REQUIRED. It SHALL specify the URI(s) where users (subscribers, relying parties) can obtain scheme-specific information.

Format: A sequence of multilingual pointers (see clause 5.1.3).

Meaning: The referenced URI(s) MUST provide a path to information describing the general terms and conditions of the scheme, its criteria for TSP and service approval and other generic information which applies to the scheme operations.

Note: The URI(s) could differ from the URI(s) provided in clause 5.3.5.2, e.g. if the scheme operator wanted to have a different service or facility for handling e-mails.

5.3.8 Status determination approach

Description: This field is REQUIRED. It SHALL specify the identifier of the status determination approach.

Format: A status determination approach indicator expressed as one of the following URIs:

For "Generic" TSL types:

- Active;
- Passive;
- Delinquent.

For "Schemes" TSL type the URI Shall be one of those listed above (i.e. as for "Generic" TSL type) or null MAY alternatively be used.

Meaning: The quoted URI SHALL be one of those listed in clause D.2, pertaining to this field. When the TSL type is "Schemes" the field may be another URI having the same purpose, registered and described by the scheme operator or another entity, such as a community or federation of schemes, a standards body, etc.

5.3.9 Scheme type/community/rules

Description: This field is OPTIONAL. If present, it SHALL contain one or more registered URIs.

Format: A sequence of strings each one compliant with RFC 3986 [19].

Meaning: This field MAY be used by any community of users which establishes and registers a URI by which to denote participation within that community. Such communities MAY be legislative, inter-governmental, industry or other, which have registered a URI for the purposes of identifying themselves. The referenced URI(s) MUST identify the specific policy/rules against which services included in the list SHALL be assessed and from which the type of scheme or community MAY be determined. Where more than one URI is provided each MUST be a complete subset of the policy defined by its predecessor (e.g. a corporate policy might be over-arching; separate divisions MAY have their own implementations which are fully within the corporate high-level policy).

Note: By permitting a string of hierarchical URIs the scheme MAY indicate a broad set of rules within which it operates and a specific set of detailed implementation rules. E.g. consider two URIs, the first of which confirms adherence to the supervision requirements relating to Certificates as defined by Directive 1999/93/EC [1], the second of which specifies the particular rules of an individual Member State's scheme. The hierarchy of the URIs is only a logical one: the URIs themselves need not directly represent that structure.

5.3.10 Scheme territory

Description: This field is OPTIONAL. If present, it SHALL specify the country in which the scheme is established.

Format: Character string giving a Country name, as a two-character code in accordance with ISO 3166-1 [21] Alpha-2 code.

Meaning: A two-letter code which specifies the country in which the scheme is established.

5.3.11 TSL policy/legal notice

Description: This field is OPTIONAL. If present, it SHALL specify the scheme's policy or provide a notice concerning the legal status of the scheme or legal requirements met by the scheme for the jurisdiction in which the scheme is established and/or any constraints and conditions under which the TSL is maintained and offered.

Format: Either:

a) A sequence of multilingual pointers (see clause 5.1.3) for specific use as a pointer to the policy or notice; or

b) the actual text of any such policy or notice, as a multilingual character string (see clause 5.1.3).

Meaning: Any referenced URI MUST provide a path to information describing the policy under which the TSP operates or any relevant legal notices with which users of the TSL should be aware. If plain text is provided, this MUST serve the same purpose.

In either case, local language and cross-border (international) trading considerations MAY require that this information be provided both in a national language and in a commonly accepted internationally-used language.

Note: If this field is implemented using format (a) then TAB, CR and LF control characters MAY be used, irrespective of the requirements of annex E.

5.3.12 Historical information period

Description: This field is REQUIRED. It SHALL specify the duration over which historical information in the TSL is provided.

Format: Integer.

Meaning: a) 0 (zero) SHALL signify that the scheme does not retain history information;

b) 1 through 65 534 SHALL signify the number of days over which historical information in the TSL is provided;

c) 65 535 or greater SHALL signify an indefinite duration.

Note: The period chosen should take due account of the legal requirements for data retention in the host jurisdiction. A range of values 1 through 65 534 allows for a specific duration of up to at least 179 years, which is considered to be sufficient for most foreseen purposes.

5.3.13 Pointers to other TSLs

Description: This field is OPTIONAL. It MAY be used to indicate other TSLs.

Format: Sequence of one or more tuples, each tuple giving:

a) a string containing the URI of another TSL; and

b) additional information in a scheme-specific format.

Meaning: A series of pointers to the location of other TSLs, with additional information whose meaning is scheme-specific. Such TSLs MAY be maintained by other parties or by the operator of the TSL in question.

5.3.14 List issue date and time

Description: This field is REQUIRED. It SHALL specify the date and time on which the list was issued.

Format: Date-time value (see clause 5.1.4).

Meaning: Coordinated Universal Time (UTC) at which the TSL was issued.

5.3.15 Next update

Description: This field is REQUIRED. It SHALL specify the latest date and time by which the next TSL will be issued or be null to indicate a closed TSL.

Format: Date-time value (see clause 5.1.4).

Meaning: Coordinated Universal Time (UTC) by which the next TSL SHALL be issued, expressed as Zulu. If a scheme ceases operations or halts publication of its TSL a final version SHALL be published with all services' status shown as "expired" (see Service current status) and this field set null.

In the event of no interim status changes to any TSP or service covered by the scheme, the TSL MUST be re-issued by the time of expiration of the last TSL issued.

5.3.16 Scheme extensions *(new this version)*

Description: This field is OPTIONAL. It MAY be used by scheme operators (or communities thereof) to provide specific service-related information and enhancements to the present document that do not require a change in the version number, which MAY be interpreted by all accessing parties according to the specific scheme's rules.

Format: Sequence of scheme extensions, each of which MUST be selected by the scheme operator according to the meaning and information it wishes to convey within its TSL. Each extension MUST have an indication of its criticality.

Meaning: The meaning of each extension is defined by its source specification, that specification being either the scheme operator's own definition or any other extension definition produced by another entity, such as a community or federation of schemes, a standards body, etc. The criticality indication will have the same semantics as with extensions in X.509-certificates ITU-T Recommendation X.509 [26]. A system using TSLs MUST reject the TSL if it encounters a critical extension it does not recognize, while a non-critical extension MAY be ignored if it is not recognized.

Back to Logical model.

5.3.17 List of Trust Service Providers

Description: This field is OPTIONAL. In the case where no TSPs are or were recognized by the scheme (according to the scheme type and criteria), this field SHALL be absent. If one or more Trust services are or were recognized by the scheme then the field SHALL contain a sequence identifying each TSP providing one or more of those services, with details on the approval status and (where provided - see clause 5.3.12) history of each of the TSP's services.

Format: Sequence of TSP information (see clause 5.4).

Meaning: The presence or absence of TSPs within this list can only have meaning when taken in the context of the scheme's status determination approach (see clause 5.3.8). E.g. absence of any listed TSPs under a scheme working solely on a delinquent list principle suggests that there are no known TSPs which are also known to be not operating within the permissible or acknowledged bounds, whereas a similar absence of TSPs in a positive approval-list driven scheme would suggest that no TSPs are approved by the scheme.

Note: The term "TSP" is used liberally in the above text, since service providers whose services are listed under a "delinquency" scheme MAY not be deserving of the term "trusted" in the context of the scheme's rules.

5.4 TSP information

5.4.1 TSP name

Description: This field is REQUIRED. It SHALL specify the name of the legal entity responsible for the TSP's services that are or were recognized by the scheme.

Format: A sequence of multilingual character strings (see clause 5.1.3).

Meaning: The name of the legal entity responsible for the TSP MUST be the name which is used in formal legal registrations and to which any formal communication, whether physical or electronic, should be addressed.

Note: Local language and cross-border (international) trading considerations MAY require that this information be provided both in a national language (and script) and in a commonly accepted internationally-used language.

5.4.2 TSP trade name

Description: This field is OPTIONAL. If present, it SHALL specify an alternative name under which the TSP identifies itself in the provision of its services.

Format: A sequence of multilingual character strings (see clause 5.1.3).

Meaning: Any name under which the legal entity responsible for the TSP operates, in the specific context of the delivery of those of its services which are to be found in this TSL.

Note: Local language and cross-border (international) trading considerations MAY require that this information be provided both in a national language (and script) and in a commonly accepted internationally-used language.

5.4.3 TSP address

Description: This field is REQUIRED. It SHALL specify the address of the legal entity identified in clause 5.4.1, for both physical and electronic communications. Users (subscribers, relying parties) should use this address as the single contact point for enquiries, complaints, etc. to the TSP.

This is a multi-part field consisting of the TSP physical address specified in clause 5.4.3.1 and the TSP electronic address specified in clause 5.4.3.2.

5.4.3.1 TSP postal address

Description: This field is REQUIRED. It SHALL specify the postal address of the legal entity identified in clause 5.4.1, with the provision for the inclusion of the address in multiple languages.

Format: The format SHALL be the same as that specified in clause 5.3.5.1.

Meaning: This MUST be a postal address at which the TSP provides a regularly serviced capability for conventional (physical) mail.

5.4.3.2 TSP electronic address

Description: This field is REQUIRED. It SHALL specify the address of the legal entity identified in clause 5.4.1, to be used for electronic communications.

Format: The format SHALL be the same as that specified in clause 5.3.5.2.

Meaning: In the case of an e-mail address, this MUST be an address at which the TSP provides a regularly serviced customer care or help line capability. In the case of a web-site URI, this MUST lead to a capability whereby the user MAY communicate with a regularly serviced customer care or help line capability.

5.4.4 TSP information URI

Description: This field is REQUIRED. It SHALL specify the URI(s) where users (subscribers, relying parties) can obtain TSP-specific information.

Format: Multilingual pointer (see clause 5.1.3).

Meaning: The referenced URI(s) MUST provide a path to information describing the general terms and conditions of the TSP, legal issues, its customer care policies and other generic information which applies to all of its services.

Note: The URI(s) could differ from the URI provided in clause 5.4.3.2, e.g. if the scheme operator wanted to have a different service or facility for handling e-mails.

5.4.5 TSP information extensions *(new this version)*

Description: This field is OPTIONAL. It MAY be used by scheme operators to provide specific TSP-related information, to be interpreted according to the specific scheme's rules.

Format: Sequence of TSP extensions, each of which MUST be selected by the scheme operator according to the meaning and information it wishes to convey within its TSL. Each extension MUST have an indication of its criticality.

Meaning: The meaning of each extension is defined by its source specification, that specification being either the scheme operator's own definition or any other extension definition produced by another entity, such as a community or federation of schemes, a standards body, etc. The criticality indication will have the same semantics as with extensions in X.509-certificates ITU-T Recommendation X.509 [26]. A system using TSLs MUST reject the TSL if it encounters a critical extension it does not recognize, while a non-critical extension MAY be ignored if it is not recognized.

5.4.6 List of services

Description: This field is REQUIRED. It SHALL contain a sequence identifying each of the TSP's recognized services and the approval status of that service. At least one service MUST be listed, even if the information held is entirely historical.

Format: Sequence of service information (see clause 5.5).

Meaning: The presence or absence of services within this list can only have meaning when taken in the context of the scheme's status determination approach (see clause 5.3.8). E.g. no services under a scheme working solely on a delinquency list principle suggests that there are ***no*** known services which are ***not*** operating within the permissible or acknowledged bounds, whereas a similar absence of services in a positive approval list driven scheme would suggest that no services meet the scheme's criteria.

If a scheme retains historical information then that information MUST be retained even if the service's present status would not normally require it to be listed (e.g. in a positive list, the service is withdrawn; in a delinquency list, the service conforms to the required standards). Thus a TSP MUST be included even when its only listed service is in such a state, so as to preserve the history. However, if the scheme does not retain historical information then in such a situation, again as the only service related to the TSP in question, when that service needs no longer to be listed then the TSP MUST be removed as well.

Back to Logical model.

5.5 Service information

5.5.1 Service type identifier

Description: This field is REQUIRED. It SHALL specify the identifier of the service type, according to the type of TSL being presented.

Format: An identifier expressed as one of the following URIs:

For TSL type "Generic":

- CA (PKC);
- CA (QC);
- Time-stamp Authority;
- Certificate status (OCSP);
- Certificate status (CRL);
- RA;
- Id verification;
- Certificate generation;
- Attribute CA;
- Archive;
- Key escrow;
- Pin/password credential authority;
- unspecified.

For TSL type "Schemes":

- EC supervisory systems;
- EC Voluntary approval scheme;
- unspecified.

OR for any other TSL type:

- unspecified.

Meaning: The quoted URI SHALL be one of those listed in clause D.2, pertaining to this field, or another URI having the same purpose, registered and described by the scheme operator or another entity, such as a community or federation of schemes, a standards body, etc. and which is recognized by the intended user community.

5.5.2 Service name

Description: This field is REQUIRED. It SHALL specify the name under which the TSP provides the service identified in clause 5.5.1.

Format: A sequence of multilingual character strings (see clause 5.1.3).

Meaning: The name under which the TSP provides the service.

Local language and cross-border (international) trading considerations MAY require that this information be provided both in a mother language (and script) and in a commonly accepted internationally-used natural language.

5.5.3 Service digital identity

Description: This field is REQUIRED. It SHALL be either null or SHALL specify at least one representation of a digital identifier unique to the service specified in clause 5.5.1 by which the service can be unambiguously identified. The digital identifier MAY be present more than once and in different formats. If the digital identifier is present more than once, all variants MUST refer to the same identity.

Format: Character string or bit string or data structure specifying for each occurrence of the digital identifier the type of format and the data representing the digital identity. When using public-key technology (i.e. PKI), this field MUST be a representation of the public key(s) the TSP uses for providing its services; e.g. the key used for signing certificates or OCSP responses. Implementation dependent - see annexes A and B.

Meaning: The digital identifier can be of different types depending on the service. It could be a Distinguished Name (DN), a certificate which can be used to verify electronic signatures of the service provider, a public-key or a subject key identifier. If the field is null the scheme responsible for publishing the specific TSL SHALL determine and publish the meaning and significance of a null value.

Note: It is RECOMMENDED that, in order to avoid unnecessary processing overhead of parsing a public key certificate, where a DN is available it is stated before any other forms of service digital-identity (e.g. before a public key certificate, which would require parsing to extract include the DN).

5.5.4 Service current status

Description: This field is REQUIRED. It SHALL specify the identifier of the status of the service.

Format: An identifier expressed as one of the following URIs:

- in accordance;
- expired;
- suspended;
- revoked;
- not in accordance.

Meaning: The quoted URI SHALL be one of those listed in clause D.2, pertaining to this field.

Note: This is the fundamental aspect of the TSL - i.e. the service's status. That status, whilst having one of the five distinct values as specified above, needs to be interpreted with regard to the scheme's status determination approach (see clause 5.3.8) which indicates the general types of criteria being applied.

Table 1 is intended to assist in that understanding. The meanings given apply to a status given in either the current or historical part of the TSL, for a scheme which is known still to be operational.

Should the scheme no longer be operational (which MAY be determined by all the current statuses indicating "expired", or implied by the "next update" time having been exceeded or set to null) only the historic information should be relied upon. This is because either the status will have been set to "expired" when the scheme ceased operations and hence no subsequent status information will have been maintained, or the scheme ceased operations before it could effect a re-issue of the TSL in which case it could be uncertain the extent to which the indicated current status remained valid after the publication of the list.

In table 1, grey shading indicates an unlikely combination of approach vs. status, black indicates such a combination is not possible.

Table 1: Meaning of Service status in relation to the Status determination approach

		Status determination approach		
		positive assessment (active approval)	**nomination/observation (passive approval)**	**delinquent**
Service current or previous status	**in accordance**	An assessment has been performed on behalf of the scheme operator and the TSP and its service found to be in compliance.	The service is known to be operational and has not been found to be non-compliant with the scheme's criteria.	This combination cannot exist (since only those non-compliant with the scheme's criteria are listed).
	expired, not renewed	The validity of the assessment has lapsed without the service being re-assessed.	The service is understood to have ceased operations.	This combination cannot exist (since only those TSPs and services non-compliant with the scheme's criteria are listed).
	suspended	No specific conclusion should be drawn - it could be because the service's validity is being verified (for reasons which are likely to be specific to the scheme) or there could be a delay in renewal.	Although no explicit approval is granted under these schemes, such a status could be used if a scheme's possible non-compliance was under investigation.	This combination unlikely to exist (since only those which are non-compliant are listed), although a scheme could, at its own discretion, use such a status if it was investigating a scheme's possible flagging as "non-compliant".
	revoked	Having once been found to be in conformance with the scheme's criteria, the TSP and/or the service have failed to continue to fulfil the criteria set by the scheme.	Essentially as per "not in accordance" (below), except that this combination is unlikely to exist since a scheme applying passive observation is not generally likely to have granted any right or recognition to explicitly revoke, and would there apply the status "not in accordance".	This combination cannot exist, since no positive recognition is granted, hence it cannot be withdrawn (revoked).
	not in accordance	Essentially as per "revoked" (above), except that this combination is unlikely to exist since a scheme exercising positive assessment is more likely to want to remove a positive assertion in the TSP or scheme when there has been a failure to continue to fulfil the criteria set by the scheme, and would therefore apply the status "revoked'.	The TSP and/or the service have been found to be non-compliant with the criteria required by the scheme.	The TSP and/or the service have been found to be non-compliant with the criteria required by the scheme for the TSPs/services listed.

It should be understood that few schemes could state with absolute certitude that all services which potentially fall within their scope are actually listed within the TSL, irrespective e of their status determination approach.

5.5.5 Current status starting date and time

Description: This field is REQUIRED. It SHALL specify the date and time on which the current approval status became effective.

Format: Date-time value (see clause 5.1.4).

Meaning: Coordinated Universal Time (UTC) at which the current approval status became effective.

Note: The user (subscribers, relying parties) could apply this information by comparing it with other available information, e.g. the date and time on which a certificate or a time stamp was issued. From the comparison, the user could determine whether the specific service of the TSP had the desired approval status under the scheme at the date and time of provision of the service.

5.5.6 Scheme service definition URI

Description: This field is OPTIONAL. If present, it SHALL specify the URI(s) where users (subscribers, relying parties) can obtain service-specific information provided by the scheme operator.

Format: A sequence of multilingual pointers (see clause 5.1.3).

Meaning: The referenced URI(s) MUST provide a path to information describing the service as specified by the scheme.

5.5.7 Service supply points

Description: This field is OPTIONAL. If present, it SHALL specify one or more URIs where users (subscribers, relying parties) can access the service.

Format: A sequence of character strings whose syntax MUST be compliant with RFC 3986 [19].

Meaning: The referenced URI(s) MUST specify where and how the service can be accessed.

5.5.8 TSP service definition URI

Description: This field is OPTIONAL. If present, it SHALL specify the URI(s) where users (subscribers, relying parties) can obtain service-specific information provided by the TSP.

Format: A sequence of multilingual pointers (see clause 5.1.3).

Meaning: The referenced URI(s) MUST provide a path to information describing the service as specified by the TSP.

5.5.9 Service information extensions *(new this version)*

Description: This field is OPTIONAL. It MAY be used by scheme operators (or communities thereof) to provide specific service-related information and enhancements to the present document that do not require a change in the version number, to be interpreted by all accessing parties according to the specific scheme's rules.

Format: Sequence of service information extensions, each of which MUST be selected by the scheme operator according to the meaning and information it wishes to convey within its TSL. Each extension MUST have an indication of its criticality.

Meaning: The meaning of each extension is defined by its source specification, that specification being either the scheme operator's own definition or any other extension definition produced by another entity, such as a community or federation of schemes, a standards body, etc. The criticality indication will have the same semantics as with extensions in X.509-certificates ITU-T Recommendation X.509 [26]. A system using TSLs MUST reject the TSL if it encounters a critical extension it does not recognize; while a non-critical extension MAY be ignored if it is not recognized.

5.5.10 Service approval history

Description: This field is OPTIONAL but MUST be present if Historical information period is non-zero (i.e. the scheme retains or intends to retain historical information at all). In the case where historical information is intended to be retained but the service has no history prior to the current status (i.e. a first recorded status or history information not retained by the scheme operator) this field SHALL be empty. Otherwise, for each change in TSP service approval status which occurred within in the historical information period as specified in clause 5.3.12, information on the now previous approval status SHALL be provided in descending order of status change date and time (i.e. the date and time on which the subsequent approval status became effective).

Format: Sequence of History information (see clause 5.6).

Meaning: When present, a sequence of all previous status entries which the scheme has recorded for the given TSP and service, within the period over which historical information is retained.

Back to Logical model.

5.6 Service approval history information

5.6.1 Service type identifier

Description: This field is REQUIRED. It SHALL specify the identifier of the service type, with the Format and Meaning used in clause 5.5.1.

5.6.2 Service name

Description: This field is REQUIRED. It SHALL specify the name under which the TSP provided the service identified in clause 5.5.1, with the Format and Meaning used in clause 5.5.2.

Note: This clause does not require that the name be the same as that specified in clause 5.5.2. A change of name MAY be one of the circumstances requiring a new status.

5.6.3 Service digital identity

Description: This field is REQUIRED. It SHALL specify at least one representation of a digital identifier unique to the service specified in clause 5.5.1, with the Format and Meaning used in clause 5.5.3.

5.6.4 Service previous status

Description: This field is REQUIRED. It SHALL specify the identifier of the previous status of the service, with the Format and Meaning used in clause 5.5.4.

5.6.5 Previous status starting date and time

Description: This field is REQUIRED. It SHALL specify the date and time on which the previous status in question became effective, with the Format and Meaning used in clause 5.5.5.

5.6.6 Service information extensions *(new this version)*

Description: This field is OPTIONAL. It MAY be used by scheme operators to provide specific service-related information, to be interpreted according to the specific scheme's rules, with the Format and Meaning used in clause 5.5.9.

Back to Logical model.

5.7 Signature

5.7.1 Signed TSL

The Trust-service status list SHALL be signed by the scheme operator to ensure its authenticity and integrity. This clause does not prescribe the format of the signature but refers to normative annexes A and B for implementations using ASN.1 and XML respectively, and additional informative guidance given in annex F. Only general requirements regarding the signature are stated in this present clause. The fields defined in this clause are all REQUIRED but to accommodate implementation dependent issues, they need not necessarily appear in the following order. The present document REQUIRES that scheme operators acquire and use to sign their TSL a public-key cryptography signing key which is bound into a public-key certificate conformant with ITU-T Recommendation X.509 [26].

5.7.2 Scheme identification

Description: This field is REQUIRED. It SHALL specify a reference assigned by the scheme operator which uniquely identifies the specific scheme and this TSL, and MUST be included in the calculation of the signature.

Format: Character string or Bit string.

Meaning: MUST represent of one of the following:

- an X.509-certificate conformant to ITU-T Recommendation X.509 [26];
- a value of an SubjectKeyIdentifier extension conformant to ITU-T Recommendation X.509 [26];
- an implementation-specific X.509-certificate identifier;
- a public key.

The actual choice is implementation dependent and will depend on constraints imposed by the implementation framework (like CMS or XML-Signature).

Note: If the scheme operator operates more than one scheme for which it publishes a TSL they should use a unique reference in this field for **each** TSL they publish.

5.7.3 Signature algorithm identifier

Description: This field is REQUIRED. It SHALL specify the cryptographic algorithm that has been used to create the signature and MUST be included in the calculation of the signature.

Format: Character string or Bit string, depending on format used.

Meaning: Depending on the algorithm used, this field MAY require additional parameters. This field MUST be included in the calculation of the signature.

5.7.4 Signature value

Description: This field is REQUIRED. It SHALL contain the actual value of the digital signature. Since the signature protects the signed information from undetected manipulation, all fields of the TSL except the signature value itself MUST be included in the calculation of the signature. The calculation of the digital signature SHALL cover all fields described in clauses 5.2 to 5.6 as well as clauses 5.7.2 and 5.7.3.

Format: Implementation dependent - see annexes A and B.

Meaning: Contains the actual value of the digital signature.

Back to Logical model.

6 Operations

6.1 TSL publication

Schemes will likely make TSLs available to TSL-users by publishing them in a Directory. The Directory is also the normal distribution mechanism for certificates. The Hypertext Transfer Protocol (HTTP) defined in RFC 2616 [12] and the File Transfer Protocol (FTP) defined in RFC 959 [3] offer alternate methods for certificate and TSL distribution. The transport protocols specified below allow end entities to access TSLs. Repository providers must support at least LDAP **or** HTTP transports, but it is recommended to support both. They may also support FTP. An application processing TSLs must support at least HTTP or LDAP transport and may support FTP.

If the scheme operator allows issuing new TSLs before the time indicated in the Next update field, there is a possibility for attacks, where the most recent TSL is replaced by an older issue still appearing to be valid. To counter such attacks, the use of secure channels, like TLS, is strongly recommended. Otherwise, there is no requirement for specific security mechanisms to be applied at this level, since the TSLs are signed data structures and thus suitably protected.

Any file containing a TSL must either only contain a DER-encoded ASN.1 representation or an XML representation of the TSL as specified in the present document. There MUST be no extraneous header or trailer information in the file.

6.1.1 Transport Protocols

6.1.1.1 LDAP transport

This text following in this clause refers explicitly to LDAP v3.

6.1.1.1.1 Attributes and Object class definition

In order to use an LDAP-server-like repository to publish the TSLs in compliance with the present document, these servers MUST be compliant with LDAP version 3: therefore they MUST support the syntax notation defined by RFC 2252 [8] and they must be also compliant with RFC 2251 [7] and RFC 2256 [10].

1) cn: this attribute MUST be present and the value MUST be the Relative Distinguished Name (RDN) of the entry, in form of Common Name; this attribute is defined by RFC 2256 [10]. It is RECOMMENDED to use the Scheme name field of the TSL as the value or as part of the value for the CN. This helps to search the directory for TSLs more efficiently.

2) tdpIndicator: this attribute MUST be present and the value MUST be the OID 0.4.0.2231.1.1; in order to speed-up the search operations, the indexing of this attribute is RECOMMENDED; the attribute is defined according to the RFC 2252 [8] syntax as:

```
(  0.4.0.2231.5.2
NAME 'tdpIndicator'
DESC 'Indexed. Indicates that the entry contains a TSL (the value of the OID is
0.4.0.2231.1.1)'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.38
EQUALITY objectIdentifierMatch
SINGLE-VALUE
)
```

3) tslDer: this attribute MAY be present; in this case the value must be the sequence of bytes that represents the DER-encoded TSL; the attribute is defined according to the RFC 2252 [8] syntax as:

```
(  0.4.0.2231.5.3
NAME 'tslDer'
DESC 'DER-encoded TSL'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
EQUALITY asn1Match
SINGLE-VALUE
)
```

4) tslXml: this attribute MAY be present; in this case the value must be the sequence of bytes that represents the XML-encoded TSL; the attribute is defined according to the RFC 2252 [8] syntax as:

```
(  0.4.0.2231.5.4
NAME 'tslXml'
DESC ' XML-encoded TSL'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
EQUALITY octetStringMatch
SINGLE-VALUE
)
```

At least one of the optional attributes tslDer and tslXml MUST contain a value: it is RECOMMENDED that both have a value.

A TSL published on an LDAP server MUST be stored within a dedicated entry. The structural Object Class of such an entry MUST be tslDistributionPoint and MUST use the attributes previously defined. This Object Class is defined according to RFC 2252 [8] syntax as:

```
(0.4.0.2231.5.1
NAME 'tslDistributionPoint'
DESC 'OC containing the TSL'
STRUCTURAL
SUP top
MUST ( cn $ tslTag )
MAY  ( tslDer $ tslXml )
)
```

Each TSL is stored within a specific entry of the LDAP server and this entry MAY be located in any point of the Directory Information Tree (DIT). Multiple TSLs can be stored within the DIT. In this case, each TSL MUST be stored in a different entry so as to be uniquely identified by the Distinguished Name (DN) of the entry that contains it.

For each TSL it is possible to store both the DER-encoded and the XML-encoded TSL, but at least one of the two formats MUST be present (i.e. the corresponding attribute MUST have a value). If both formats are published, they MUST be stored in the same entry. Each entry constitutes a TSL Distribution Point (TDP).

Within the DIT, the tslDistributionPoint SHOULD be hierarchically located under an entry whose class is one of the following:

- domain;
- locality;
- organization;
- organizationalUnit;
- organizationalPerson;
- organizationalRole;
- applicationProcess.

6.1.1.2 HTTP-Transport

This clause specifies a means for transport of TSLs via the Internet using HTTP.

6.1.1.2.1 HTTP-Media Type

TSL payloads MUST be sent using one of the following two media types, depending on the version of the TSL (ASN.1 or XML):

- application/vnd.etsi.tsl+der.
- application/vnd.etsi.tsl+xml.

The client MAY, when sending requests, provide an HTTP Accept header field. This header field SHOULD indicate an ability to accept, as a minimum "*application/vnd.etsi.tsl+der*" OR "*application/vnd.etsi.tsl+xml*".

6.1.1.3 FTP-Transport

TSL-repository-providers may also offer FTP as a way to access TSLs similar to the HTTP transport. Since FTP does not support media types, as does HTTP, it is RECOMMENDED that the file extension defined in clause 6.1.1.5 be used, to enable media type recognition by filename.

6.1.1.4 Email Transport

This clause specifies the message format required for transport of TSLs via Internet mail. A scheme or another service provider may want to "push" automatically newly-published TSLs to its users, using email as the transport mechanism.

The email containing the TSL payloads MUST be compliant to RFC 2822 [14] and the RFC 2045 [4] Message.

6.1.1.4.1 Content-Types

TSL payloads must be sent with one of the following two content types, depending on the representation of the TSL (ASN.1 or XML):

- application/vnd.etsi.tsl+der.
- application/vnd.etsi.tsl+xml.

6.1.1.4.2 Encoding considerations

For the DER version it is RECOMMENDED to use base64-transfer encoding. For the XML version, the encoding considerations of clause 3.2 of RFC 3023 [15] as well as clause 6.1.1.5 of this document are applicable.

6.1.1.5 MIME registrations

Two MIME-Types and file-extensions support the transfer of TSLs:

NOTE: At the time of publication the MIME-Types are undergoing registration procedure with IANA and users are advised to make their own checks for completion of these formalities (the list of Directories of Content Types and Subtypes can be found here: http://www.iana.org/assignments/media-types/application/).

MIME media type name:	Application.
MIME subtype name:	vnd.etsi.tsl+der.
Required parameters:	none.
encoding considerations:	will be none for 8-bit transports and base64 for SMTP or other 7-bit transports.
File extension:	dtsl or dts.

MIME media type name:	Application.
MIME subtype name:	vnd.etsi.tsl+xml.
Required parameters:	charset="utf-8".
encoding considerations:	will be none for 8-bit transports and quoted printable for SMTP or other 7-bit transports.
File extension:	xtsl or xts.

Security considerations:	TSLs do not contain any active code or invoke any automated processing by itself. It is expected that clients only parse the TSL and that there is no security risk.
Published specification:	The TSL format as defined in the present document.

6.2 TSL Signer Certificate

Scheme operators MAY want to restrict the use of key-pairs to sign TSLs only. In this case, they MUST use an X.509 v3 certificate with the following key purpose id in the extended key usage extension:

```
-- OID for TSL signing KeyPurposeID for ExtKeyUsageSyntax

   id-tsl OBJECT IDENTIFIER { itu-t(0) identified-organization(4)
                    etsi(0) tsl-specification (2231) }
   id-tsl-kp OBJECT IDENTIFIER ::= { id-tsl kp(3) }
   id-tsl-kp-tslSigning OBJECT IDENTIFIER ::= { id-tsl-kp tsl-signing(0) }
```

6.3 TSL Distribution Points

Trust Service Providers may wish to give information on how to locate a TSL of the scheme they operate under. To do so, they MAY include the following extension in their trust service tokens (certificates, CRLs, time stamp tokens, OCSP responses and other). If the extension mechanism allows for the expression of criticality, this extension MUST NOT be marked critical. The value of this extension will be a sequence of URIs.

```
-- OID for TSLDistributionPoints extension

   id-tsl OBJECT IDENTIFIER { itu-t(0) identified-organization(4)
                    etsi(0) tsl-specification (2231) }
   id-tsl-extensions OBJECT IDENTIFIER ::= { id-tsl extensions(4) }
   id-tsl-extensions-tdp OBJECT IDENTIFIER ::= { id-tsl-extensions tdp(0) }

TSLDistributionPoints ::= SEQUENCE SIZE(1..MAX) OF IA5String
```

Annex A (normative): Implementation in ASN.1

A.1 Structure of the Trust-service Status List

A.1.1 ASN.1 versioning

This clause specifies the ASN.1 structures to be used when implementing an ASN.1-version of the present document. The field names used reflect those assigned to fields in clause 5.

The ASN.1 syntax used in this annex is the 1988 version, as defined by ITU-T Recommendation X.208 [24] with the addition of "UTF8String" type imported from the hybrid ASN.1 module of RFC 3280 [17]. These additions are imported so as to enhance interoperability by avoiding ambiguity concerning signature algorithms and digest calculation. The following schema requires the use of a "relaxed compiler' to accommodate these two special types.

The ASN.1 in this Annex may be converted into the 1997 syntax by using the Information Object Classes introduced by that version to replace the type "ANY DEFINED BY" (this type not being supported by the 1997 version) and removing the importation of "UTF8String" type, plus amending the module header appropriately.

The ASN.1 implementation of the TSL must be encoded by using the Distinguished Encoding Rules defined by ITU-T Recommendation X.690 [28].

The header of the ASN.1 module is specified as follows:

```
ETSI-TSL-v2-88syntax { itu-t(0) identified-organization(4) etsi(0)
  tsl-specification (2231) id-mod(0) v2-88syntax (1)}
DEFINITIONS EXPLICIT TAGS ::=

BEGIN

-- EXPORTS All
IMPORTS
  -- Internet X.509 Public Key Infrastructure - Certificate and CRL Profile: RFC 3280
  Extensions, Certificate, CertificateSerialNumber, AlgorithmIdentifier,
    UTF8String, SubjectPublicKeyInfo, Name
    FROM PKIX1Explicit88 {iso(1) identified-organization(3) dod(6) internet(1)
      security(5) mechanisms(5) pkix(7) id-mod(0) id-pkix1-explicit(18)}
  KeyIdentifier
    FROM PKIX1Implicit88 {iso(1) identified-organization(3) dod(6) internet(1)
      security(5) mechanisms(5) pkix(7) id-mod(0) id-pkix1-implicit(19)}
  -- Cryptographic Message Syntax (CMS): RFC 3852
  ContentInfo, ContentType, id-signedData, SignedData, EncapsulatedContentInfo,
    SignerInfo
    FROM CryptographicMessageSyntax2004 {iso(1) member-body(2) us(840) rsadsi(113549)
      pkcs(1) pkcs-9(9) smime(16) modules(0) cms-2004(24) };
```

A.1.2 Basic types

The following are basic types used more than once within the ASN.1 module.

A.1.2.1 The `NonEmptyURI` type

The following type is used to carry a non-empty URI.

```
NonEmptyURI ::= IA5String (SIZE (1..MAX))
```

A.1.2.2 The LanguageTag type

The following type is used to carry a language tag according to RFC 3066 [16].

```
LanguageTag ::= PrintableString (SIZE (1..MAX))
```

A.1.2.3 The CountryCode type

The following type is used to carry the country code according to ISO 3166-1 [21].

```
CountryCode ::= PrintableString (SIZE (2))
```

A.1.2.4 The MultiLangPointer type

This definition specifies a format for giving alternative pointers (URIs) to the same text translated in different languages and scripts. The value of the languageTag field MUST be a language tag as specified by RFC 3066 [16] and indicates the language of the text pointed by the URI contained within the companion uRI field. The text pointed by the URI can be expressed by using any format or language (plain text, HTML, XML, etc.).

```
MultiLangPointer ::= SEQUENCE SIZE (1..MAX) OF LangPointer

LangPointer ::= SEQUENCE {
  languageTag LanguageTag,
  uRI         NonEmptyURI
  }
```

A.1.2.5 The MultiLangString type

This definition specifies a format for giving alternative text strings in different languages and scripts. The text field contains plain text, with characters from the ISO 10646 [23] character set without any escape sequence and UTF-8 encoded. The value of the languageTag field MUST be a language tag as specified by RFC 3066 [16] and indicates the language of the text contained within the companion text field.

```
MultiLangString ::= SEQUENCE SIZE (1..MAX) OF LangString

LangString ::= SEQUENCE {
  languageTag LanguageTag,
  string      UTF8String (SIZE (1..MAX))
}
```

A.1.2.6 The PhysicalAndElectronicAddresses type

This definition specifies a format for giving physical addresses in different languages and scripts and for giving the electronic addresses.

The streetAddress, locality, stateOrProvince, postalCode, countryName fields contain plain text, with characters from the ISO 10646 [23] character set without any escape sequence and UTF-8 encoded. The value of the languageTag field MUST be a language tag as specified by RFC 3066 [16] and indicates the language of the text contained within the companion streetAddress, locality, stateOrProvince, postalCode, countryName fields within the same sequence.

The electronicAddresses field MUST include at least one electronic address and MAY include more than one. Each electronic address is a non-empty URI that MUST represent either:

- a RFC 2822 e-mail address, expressed by using the "mailto:" URI scheme as defined by RFC 2368 [11]; or
- a web-site.

```
PhysicalAndElectronicAddresses ::= SEQUENCE {
  physicalDeliveryAddress  MultiLangAddress,
  electronicAddresses      ElectronicAddresses
  }

MultiLangAddress ::= SEQUENCE SIZE (1..MAX) OF LangAddress

LangAddress ::= SEQUENCE {
  languageTag        LanguageTag,
  streetAddress      UTF8String(SIZE (1..MAX)),
  locality           UTF8String(SIZE (1..MAX)),
  stateOrProvince    UTF8String (SIZE (1..MAX)) OPTIONAL,
  postalCode         UTF8String(SIZE (1..MAX)),
  countryName        CountryCode
  }

ElectronicAddresses ::= SEQUENCE SIZE (1..MAX) OF NonEmptyURI
```

A.1.3 General Structure

The main structure of the ASN.1 implementation of a TSL is defined as follows:

```
TSL ::= ContentInfo

ToBeSignedTSL ::=SEQUENCE {
  tSLtag                       TSLtag,
  version                      Version,
  sequenceNumber               SequenceNumber,
  tSLtype                      TSLtype,
  schemeOperatorName           SchemeOperatorName,
  schemeOperatorAddress        SchemeOperatorAddress,
  schemeName                   SchemeName,
  schemeInformationURI         SchemeInformationURI,
  statusDeterminationApproach  StatusDeterminationApproach,
  schemeTypeCommunityRules     [0] SchemeTypeCommunityRules OPTIONAL,
  schemeTerritory              [1] SchemeTerritory OPTIONAL,
  tSLpolicy                    [2] TSLpolicy OPTIONAL,
  historicalInformationPeriod  HistoricalInformationPeriod,
  pointersToOtherTSLs          [3] PointersToOtherTSLs OPTIONAL,
  listIssueDateTime            ListIssueDateTime,
  nextUpdate                   NextUpdate,
  schemeExtensions             [4] Extensions OPTIONAL,
  tSPlist                      TSPlist OPTIONAL
  }
```

A.2 Scheme information fields

A.2.1 The tSLtag field

This field is REQUIRED. It shall facilitate the identification of the TSL as such, when electronic searches are conducted across the Internet. The type of this field is TSLtag, defined as follows:

```
TSLtag ::= NonEmptyURI
```

The tag is implemented as a string (with an embedded URI) whose unique value MUST be:

```
tslTag-value NonEmptyURI ::="http://uri.etsi.org/02231/TSLtag"
```

A.2.2 The version field

This REQUIRED field specifies the version of the TSL format. In this version of the TSL it must have the value "2". The type of this field is Version, defined as follows:

```
Version ::= INTEGER {  v2(2)  }
```

A.2.3 The sequenceNumber field

This REQUIRED field specifies the sequence number of the TSL. At the first release of the TSL, the value of the sequence number shall be "1". The value shall be increased by "1" at each subsequent release of the TSL. The type of this field is SequenceNumber, defined as follows:

```
SequenceNumber ::= INTEGER (1..MAX)
```

A.2.4 The tSLtype field

This REQUIRED field specifies the type of the TSL. The value SHALL be one of the URIs listed in clause D.2 or another registered URI having the same purpose. The type of this field is TSLtype, defined as follows:

```
TSLtype ::= NonEmptyURI
```

A.2.5 The schemeOperatorName field

This REQUIRED field specifies the name(s) of the scheme operator. The type of this field is SchemeOperatorName, defined as follows:

```
SchemeOperatorName ::= MultiLangString
```

A.2.6 The schemeOperatorAddress field

This REQUIRED field includes the scheme operator postal address (see clause 5.3.5.1) and the scheme operator electronic address (see clause 5.4.3.2). The type of this field is SchemeOperatorAddress, defined as follows:

```
SchemeOperatorAddress ::= PhysicalAndElectronicAddress
```

A.2.7 The schemeName field

This REQUIRED field specifies the name(s) under which the scheme operates. The type of this field is SchemeName, defined as follows:

```
SchemeName ::= MultiLangString
```

A.2.8 The schemeInformationURI field

This REQUIRED field specifies the URI where users can obtain scheme-specific information. The type of this field is SchemeInformationURI, defined as follows:

```
SchemeInformationURI ::= MultiLangPointer
```

A.2.9 The statusDeterminationApproach field

This REQUIRED field specifies the status determination approach. The value SHALL be one of the URIs listed in clause D.2 or another registered URI having the same purpose. The type of this field is StatusDeterminationApproach, defined as follows:

```
StatusDeterminationApproach  ::= NonEmptyURI
```

A.2.10 The schemeTypeCommunityRules field

This OPTIONAL field is a sequence of registered Uniform Resource Identifiers (URIs), used as unique identifiers when required to indicate one or more sets of rules/policies under which the TSL has been issued. If this field is present, at least one URI MUST be present. The type of this field is SchemeTypeCommunityRules, defined as follows:

```
SchemeTypeCommunityRules ::= SEQUENCE SIZE (1..MAX) OF NonEmptyURI
```

A.2.11 The schemeTerritory field

This OPTIONAL field specifies the country in which the scheme is established. The type of this field is SchemeTerritory, defined as follows:

```
SchemeTerritory ::= CountryCode
```

A.2.12 The tSLpolicy field

This OPTIONAL field can be used to specify the scheme's policy or provide a notice concerning the legal status of the scheme or legal requirements met by the scheme for the jurisdiction in which the scheme is established and/or any constraints and conditions under which the TSL is maintained and offered. It can be provided in multiple languages. This string is either recognized as a registered URI or represents the textual form of the legal notice. The type of this field is TSLpolicy, defined as follows:

```
TSLpolicy  ::= CHOICE {
  pointer  [0] MultiLangPointer,
  text     [1] MultiLangString
  }
```

A.2.13 The historicalInformationPeriod field

This REQUIRED field contains the duration over which historical information in this TSL is provided (see clause 5.3.12). The type of this field is HistoricalInformationPeriod, defined as follows:

```
HistoricalInformationPeriod ::= INTEGER (0..MAX)
```

A.2.14 The pointersToOtherTSLs field

This OPTIONAL field specifies the URI where users can obtain other TSLs. The field can contain a list of couples holding a URI pointing to the TSL and additional information about that TSL. If this field (pointersToOtherTSLs) is present, at least one couple MUST be present. The additionalInformation field is implementation-specific and it can be empty (zero-length string), free text with characters from ISO 10646 [23], some character-based and machine-readable code (e.g. a URI or a MIME object) or other, with an optional language indication.

The type of this field is PointersToOtherTSLs, defined as follows:

```
PointersToOtherTSLs  ::= SEQUENCE SIZE (1..MAX) OF OtherTSLPointer

OtherTSLPointer ::= SEQUENCE {
  tSLLocation            NonEmptyURI,
  additionalInformation  TSLqualifiers
  }

TSLqualifiers ::= SEQUENCE (1..MAX) OF TSLqualifier

TSLqualifier ::= CHOICE {
  textualQualifier  [0] MultiLangString,
  otherQualifier    [1] OtherQualifier
  }

OtherQualifier ::= SEQUENCE {
  type   OBJECT IDENTIFIER,
  value  ANY DEFINED BY type
  }
```

A.2.15 The listIssueDateTime field

This REQUIRED field gives date and time of the issuance of the TSL, expressed as UTC time. All encoding requirements mandated by the Distinguished Encoding Rules ITU-T Recommendation X.690 [28] apply. In addition, the time indication MUST not include fractional seconds. The type of this field is ListIssueDateTime, defined as follows:

```
ListIssueDateTime ::= GeneralizedTime
```

A.2.16 The nextUpdate field

This REQUIRED field specifies the latest date and time by which the next TSL will be issued expressed as UTC time. All encoding requirements mandated by the Distinguished Encoding Rules ITU-T Recommendation X.690 [28] apply. In addition, the time indication MUST not include fractional seconds. The type of this field is NextUpdate, defined as follows:

```
NextUpdate ::= CHOICE {
  never  NULL,
  update GeneralizedTime
  }
```

A.2.17 The schemeExtensions field

This is an OPTIONAL field useful to carry additional data at the "scheme" hierarchical level. The type of this field is Extensions that is imported from RFC 3280 [17]. The structure of the Extensions type, the meaning of the fields it contains and the processing rules are the same as in RFC 3280 [17]. The additional data are conveyed through one or more "extensions" that MAY be present within the schemeExtensions field. Each "extension" is uniquely identified by the field extnID and may be marked as critical through the critical field. Applications MUST reject the TSL if they encounter a critical "extension" that they do not recognize. However, they MAY ignore a non-critical extension that they do not recognize.

A.2.18 The tSPlist field

This OPTIONAL field includes the list of all TSP information. If present it SHALL contain at least one TSP instance. For each service provider a name field, an alternative trading name, an address, and a pointer to a web page are REQUIRED.

The list of services offered is REQUIRED and at least one service MUST be listed. The type of this field is TSPlist, defined as follows:

```
TSPlist ::=SEQUENCE SIZE (1..MAX) OF TrustServiceProviderInformation

TrustServiceProviderInformation ::= SEQUENCE {
  tSPname            TSPname,
  tSPtradeName       [0] TSPtradeName OPTIONAL,
  tSPaddress         TSPaddress,
  tSPinformationURI  TSPinformationURI,
  tSPextensions      [1] Extensions OPTIONAL,
  listOfServices     [2] ListOfServices
  }
```

A.3 TSP information fields

A.3.1 The tSPname field

This REQUIRED field specifies the name of the Trust Service Provider and supports multiple languages. The type of the field is TSPname, defined as follows:

```
TSPname ::= MultiLangString
```

A.3.2 The tradeName field

This OPTIONAL field contains alternative trading names of the Trust Service Provider and supports multiple languages. The type of this field is TSPtradeName, defined as follows:

```
TSPtradeName ::= MultiLangString
```

A.3.3 The tSPaddress field

This REQUIRED field contains the address of the Trust Service Provider. The type of this field is TSPaddress, defined as follows:

```
TSPaddress ::= PhysicalOrElectronicAddress
```

A.3.4 The tSPinformationURI field

This REQUIRED field contains a pointer to a web page holding service-specific information. The type of this field is TSPinformationURI, defined as follows:

```
TSPinformationURI ::= MultiLangPointer
```

A.3.5 The tSPextensions field

This is an OPTIONAL field useful to carry additional information at the "TSP" hierarchical level. The type of this field is Extensions that is imported from RFC 3280 [17]. The structure of the Extensions type, the meaning of the fields it contains and the processing rules are the same as in RFC 3280 [17]. The additional data are conveyed through one or more "extensions" that MAY be present within the tSPExtensions field. Each "extension" is uniquely identified by the field extnID and may be marked as critical through the critical field. Applications MUST reject the TSL if they encounter a critical "extension" that they do not recognize. However, they MAY ignore a non-critical extension that they do not recognize.

A.3.6 The listOfServices field

This REQUIRED field contains information of a list of Trust Services the TSP offers. At least one service MUST be listed. The type of this field is ListOfServices, defined as follows:

```
ListOfServices ::= SEQUENCE SIZE (1..MAX) OF TSPserviceInformation

TSPserviceInformation ::= SEQUENCE {
  serviceType               ServiceType,
  serviceName               ServiceName,
  serviceDigitalIdentity    ServiceDigitalIdentity,
  currentServiceStatus      ServiceStatus,
  currentStatusStartingTime StatusStartingTime,
  schemeURI                 [0] SchemeURI OPTIONAL,
  tspURI                    [1] TspURI OPTIONAL,
  serviceSupplyPoints       [2] ServiceSupplyPoints OPTIONAL,
  srvcExtensions            [3] Extensions OPTIONAL,
  serviceApprovalHistory    [4] ServiceApprovalHistory OPTIONAL
 }
```

A.4 TSP service information fields

A.4.1 The serviceType field

This REQUIRED field specifies the identifier of the service type. The value SHALL be one of the URIs listed in clause D.2 or another registered URI having the same purpose. The type of this field is ServiceType, defined as follows:

```
ServiceType ::= NonEmptyURI
```

A.4.2 The serviceName field

This REQUIRED field specifies the name under which the service is provided. The type of this field is ServiceName, defined as follows:

```
ServiceName ::= MultiLangString
```

A.4.3 The serviceDigitalIdentity field

This is a REQUIRED field. The service digital identity can be realized in a number of different ways, depending on the service offered. It could be a certificate which can be used to verify electronic signatures of the service provider, a public key or a key identifier or a collection of these types. Each of the included attributes can be used for the identification of the service. How many have to be considered for a complete identification is beyond the scope of the present document, it being dependent on the policy of the TSP as well as that of the user/relying party.

This REQUIRED field MAY be empty; this means that serviceDigitalIdentity MUST be present but no instance of IdentityAttributeTypeAndValue SHALL be. This is implemented by having the content of SET OF empty: according to the Distinguished Encoding Rules ITU-T Recommendation X.690 [28] the tag of SET OF will be present while its content will be zero octets long.

NOTE: The key identifier MUST be used only if there exists an X.509 certificate ITU-T Recommendation X.509 [26] where the subject is the service to be digitally identified. In this case the content of the key identifier MUST be the same as the content of the X.509 SubjectKeyIdentifier extension.

The type of this field is ServiceDigitalIdentity, defined as follows:

```
ServiceDigitalIdentity ::= IdentityAttributeTypeAndValues

IdentityAttributeTypeAndValues ::= SET OF IdentityAttributeTypeAndValue

IdentityAttributeTypeAndValue ::= SEQUENCE {
  type   OBJECT IDENTIFIER,
  value  ANY DEFINED BY type
  }
```

If the service digital identity is a certificate, then the type field MUST assume the following value:

```
id-certificateIdentityType OBJECT IDENTIFIER ::=
        { itu-t(0) identified-organization(4)
                   etsi(0) tsl-specification(2231) identity-types(2) certificate(0) }
```

and the value field MUST be the sequence of octets of a DER-encoded Certificate field imported from RFC 3280 [17].

If the service digital identity is a public key, then the type field MUST assume the following value:

```
id-publicKeyIdentityType OBJECT IDENTIFIER ::=
        { itu-t(0) identified-organization(4)
                   etsi(0) tsl-specification(2231) identity-types(2) public-key(1) }
```

and the value field MUST be the sequence octets of the DER-encoded SubjectPublicKeyInfo field, whose definition MUST be imported from RFC 3280 [17]. The content of SubjectPublicKeyInfo MUST be compliant with [RFC3279] or RFC 4055 [36]; it MAY be compliant with future specifications listing new algorithms and defining the formats for the related parameters.

If the service digital identity is a key identifier, then the type field MUST assume the following value:

```
id-keyIdentifierIdentityType OBJECT IDENTIFIER ::=
        { itu-t(0) identified-organization(4)
                   etsi(0) tsl-specification(2231) identity-types(2) key-identifier(2) }
```

and the value field MUST be the sequence octets of the DER-encoded KeyIdentifier type, whose definition MUST be imported from RFC 3280 [17] and the content of the imported KeyIdentifier MUST be the same as the content of SubjectKeyIdentifier within the Subject Key Identifier extension present in the X.509 certificate issued to the service.

If the service digital identity is a distinguished name, then the type field MUST assume the following value:

```
id-directoryNameIdentityType OBJECT IDENTIFIER ::=
        { itu-t(0) identified-organization(4)
                   etsi(0) tsl-specification(2231) identity-types(2) directory-name(3) }
```

and the value field MUST be the sequence of bytes of the DER-encoded Name type, whose definition MUST be imported from RFC 3280 [17].

A.4.4 The currentServiceStatus field

This REQUIRED field specifies the identifier of the current status of the service. The value SHALL be one of the URIs listed in clause D.2 or another registered URI having the same purpose. The type of this field is ServiceStatus, defined as follows:

```
ServiceStatus  ::= NonEmptyURI
```

A.4.5 The currentStatusStartingTime field

This REQUIRED field specifies the date and time on which the current status became effective. The type of this field is StatusStartingTime, defined as follows:

```
StatusStartingTime ::= GeneralizedTime
```

A.4.6 The schemeURI field

This OPTIONAL field specifies the URI where users can obtain service-specific information provided by the scheme operator. The type of this field is SchemeURI, defined as follows:

```
SchemeURI ::= MultiLangPointer
```

A.4.7 The tspURI field

This OPTIONAL field specifies the URI where users can obtain service-specific information provided by the TSP. The type of this field is TspURI, defined as follows:

```
TspURI ::= MultiLangPointer
```

A.4.8 The serviceSupplyPoints field

This OPTIONAL field carries one or more URIs that indicate the electronic point or points where a service can be accessed. The type of this field is ServiceSupplyPoints, defined as follows:

```
ServiceSupplyPoints ::= SEQUENCE SIZE (1..MAX) OF ServiceSupplyPoint

ServiceSupplyPoint ::= NonEmptyURI
```

A.4.9 The srvcExtensions field

This is an OPTIONAL field useful to carry additional information at the "service" hierarchical level. The type of this field is Extensions that is imported from RFC 3280 [17]. The structure of the Extensions type, the meaning of the fields it contains and the processing rules are the same as in RFC 3280 [17]. The additional data are conveyed through one or more "extensions" that MAY be present within the srvcExtensions field. Each "extension" is uniquely identified by the field extnID and may be marked as critical through the critical field. Applications MUST reject the TSL if they encounter a critical "extension" that they do not recognize. However, they MAY ignore a non-critical extension that they do not recognize.

A.4.10 The serviceApprovalHistory field

This OPTIONAL field provides any historical status information of the service.

This field MAY be absent or present. If present, it MAY be empty; this means that serviceApprovalHistory SHALL be present but no instance of TSPhistoryInformation will be. This is implemented by having the content of SEQUENCE OF empty: according to the Distinguished Encoding Rules ITU-T Recommendation X.690 [28] the tag of SEQUENCE OF will be present while its content will be zero octets long. The history information replicates the current status information. The type of this field is ServiceApprovalHistory, defined as follows:

```
ServiceApprovalHistory ::= SEQUENCE OF TSPhistoryInformation

TSPhistoryInformation ::= SEQUENCE {
  serviceType                ServiceType,
  serviceName                ServiceName,
  serviceDigitalIdentity     ServiceDigitalIdentity,
  previousStatus             ServiceStatus,
  previousStatusStartingTime StatusStartingTime
  srvcExtensions             [0] Extensions OPTIONAL
  }
```

A.5 Service history information fields

A.5.1 The serviceType field

This REQUIRED field specifies the previous service type. Its definition and meaning are as defined in clause A.4.1.

A.5.2 The serviceName field

This REQUIRED field specifies the previous service name. Its definition and meaning are as defined in clause A.4.2.

A.5.3 The serviceDigitalIdentity field

This REQUIRED field specifies the previous service digital identity. Its definition and meaning are as defined in clause A.4.3.

A.5.4 The previousServiceStatus field

This REQUIRED field specifies the identifier of the previous service status. Its definition and meaning are as defined in clause A.4.4.

A.5.5 The previousStatusStartingTime field

This REQUIRED field specifies the date and time on which the previous status became effective. Its definition and meaning are as defined in clause A.4.5.

A.5.6 The srvcExtensions field

This OPTIONAL field specifies the previous service extensions. Its definition and meaning are as defined in clause A.4.6.

A.6 TSL signature fields

A.6.1 The signedTSL field

This REQUIRED field contains the signature value and the signing key information.

This field SHALL contain a signature according to RFC 3852 [37]. The signature MAY include additional security feature provided by TS 101 733 [2]; therefore the content of this field MAY be also compliant with the latter which is in turn compliant with RFC 3852 [37]. The additional informative guidance given in Annex F MUST be considered when implementing the signature and selecting the security features.

The value of this field is the octets string of the DER encoding CMS ContentInfo value with the signed-data content type as defined by RFC 3852 [37]. Therefore the CMS contentType field is assigned the OID id-signedData value and the CMS content field contains the octet string of the DER-encoded SignedData type. The CMS eContent field within SignedData SHALL contain the data to be signed, namely the octet string of the DER-encoded ToBeSignedTSL value with the inclusion of the tag and length octets.

The CMS eContentType field MUST be assigned the following OID:

```
id-eContentType-signedTSL OBJECT IDENTIFIER ::=
        { itu-t(0) identified-organization(4)
                   etsi(0) tsl-specification (2231) identifiers (1) tsl-info(0) }
```

According to RFC 3852 [37] the following rules apply:

1) Since the value of eContentType is other than id-data, the value of the Version field within SignedData MUST be "3".

2) For the value of the Version field within SignerInfo the following options are possible: if the CMS SignerIdentifier field is the "CHOICE" issuerAndSerialNumber, then the version MUST be "1". If the SignerIdentifier is subjectKeyIdentifier, then the version MUST be "3".

3) Since the value of eContentType is other than id-data, the signedAttrs field MUST be present and MUST contain at least the following two signed attributes: MessageDigest and ContentType. The value of the former MUST contain the digest calculated over the eContent field. The value of the latter MUST be the same as eContentType, namely id-eContentType-signedTSL.

The following profile specific for signing TSLs applies.

Only one SignerInfo within the SET OF SignerInfos MUST be present, namely only one signature MUST be present.

The certificates field (within SignedData) MUST be either absent or present with only one certificate inside, the one of the signer of TSL. If the signer certificate is present, its type (namely the CHOICE of types among the CertificateChoices) MUST be only the X.509 certificate (namely the certificate CHOICE).

The crls field (within SignedData) MUST be absent.

According to this profile, other signed attributes and also unsigned attributes MAY be present.

A.6.2 The scheme operator identifier

Since this ASN.1 implementation of the signature is based on the CMS specification, it supports the methods natively provided by CMS to identify the scheme operator, namely the signer of TSL; therefore the use of the scheme operator public key as identifier is not supported.

Instead the following combinations are supported by CMS and one of them SHALL be used:

- The issuer/serial number pair only: the issuerAndSerialNumber CHOICE of SignerIdentifier that identifies the scheme operator certificate *not present* within the certificates field within SignedData.
- The issuer/serial number pair with the related X.509 certificate: the issuerAndSerialNumber CHOICE of SignerIdentifier that identifies the scheme operator certificate *present* within the certificates field within SignedData.
- The value of SubjectKeyIdentifier only: the subjectKeyIdentifier CHOICE of SignerIdentifier that identifies the scheme operator certificate *not present* within the certificates field within SignedData; the content of subjectKeyIdentifier MUST be identical to the content of the SubjectKeyIdentifier type of the Subject Key Identifier extension contained within the scheme operator certificate.
- The value of SubjectKeyIdentifier with the related X.509 certificate: the subjectKeyIdentifier CHOICE of SignerIdentifier that identifies the scheme operator certificate *present* within the certificates field within SignedData; the content of subjectKeyIdentifier MUST be identical to the content of the SubjectKeyIdentifier type of the Subject Key Identifier extension contained within the scheme operator certificate.

The choice of one of the listed methods is REQUIRED according to RFC 3852 [37].

Since the inclusion of the signer (i.e. the Scheme Operator) identifier in the signature calculation is REQUIRED as specified in clause 5.7.2, also a signed X.509-certificate identifier MUST be present. This identifier MUST be implemented as a CMS signed attribute in either the following ways.

A.6.2.1 ESS signing certificate attribute

The syntax of the signing certificate attribute is defined in Enhanced Security Services (ESS) RFC 2634 [13] and further qualified in the present document.

- The sequence of policy information field is not used in the present document.
- The ESS signing-certificate attribute shall be a signed attribute.
- The encoding of the ESSCertID for this certificate shall include the issuerSerial field.

The issuerAndSerialNumber present in the SignerInfo shall be consistent with issuerSerial field. The certificate identified shall be used during the signature verification process. If the hash of the certificate does not match the certificate used to verify the signature, the signature shall be considered invalid.

This way of implementing the X.509-certificate identifier is identical to the one defined in clause 5.7.3.1 of TS 101 733 [2].

NOTE: Should RFC 2634 [13] be updated with a new version the ESSCertID attribute that also supports digest algorithms other than SHA-1, then that updated attribute definition SHOULD be used according to the recommendations included within the updated specification.

A.6.2.2 CAdES other signing certificate attribute

This way of implementing the X.509-certificate identifier is the one defined in clause 5.7.3.2 of TS 101 733 [2] and SHOULD be used only when a digest algorithm other than SHA-1 is to be used.

NOTE: should RFC 2634 [13] be updated with a new version the ESSCertID attribute that also supports digest algorithms other than SHA-1, then this (TS 101 733 [2]) way of implementing the X.509-certificate identifier SHOULD NOT be used.

A.6.3 Algorithms and parameters

The algorithms and parameters and their formats supported by the present document for the CMS fields digestAlgorithms (within SignedData and SignerInfo) and signatureAlgorithm (within SignerInfo) are those specified by [RFC3370]. Further algorithms and parameters and their format MAY be specified.

Annex B (normative): Implementation in XML

This annex specifies an XML schema to be used when implementing an XML-version of the present document. The field names used reflect those assigned to fields in clause 5.

B.1 Structure of the Trust-service Status List

This annex specifies an XML schema to be used when implementing an XML-version of the present document. The field names used reflect those assigned to fields in clause 5.

B.1.1 General Rules

This clause contains general rules that apply to the XML version of the TSL.

- Applications MUST use UTF-8 encoding for XML TSLs.
- All time values are in Coordinated Universal Time (UTC) expressed as Zulu. Its value MUST NOT include fractional seconds.

B.1.2 XML-namespace and basic types

The XML namespace URI that must be used by implementations of the present document is:
http://uri.etsi.org/02231/v2#

The following namespace declarations apply for the XML Schema definitions throughout the present document:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema     targetNamespace="http://uri.etsi.org/02231/v2#"
                xmlns:tsl="http://uri.etsi.org/02231/v2#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
                    elementFormDefault="qualified"
                    attributeFormDefault="unqualified">

<xsd:import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="http://www.w3.org/2001/xml.xsd"/>
<xsd:import namespace="http://www.w3.org/2000/09/xmldsig#"
schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-core-schema.xsd"/>
```

Several types are better specified separately. These types are specified in the clauses B.1.2.1 through B.1.2.6.

B.1.2.1 The `InternationalNamesType` and `MultiLangString` Types

The `InternationalNamesType` specifies a format for giving alternative names in different languages and scripts.

It is built on the `MultiLangNormStringType` type. This type contains:

- A non-empty normalized string whose contents follow the rules established for the type `xsd:normalizedString` defined in XML Schema Part 2 [32].
- The `xml:lang` attribute identifying the language used in the string.

The `MultiLangNormStringType` type is used thorough the present document whenever there is the possibility to use normalized textual information in different languages as specified in RFC 3066 [16].

In addition, the `MultiLangStringType` type is defined for those strings that require a qualification of the language they are written but do not require normalization.

All of them are based on two non empty string types: `NonEmptyStringType` for regular strings and `NonEmptyNormStringType` for normalized strings.

Below follow their schema definitions.

```
<xsd:complexType name="InternationalNamesType">
    <xsd:sequence>
        <xsd:element name="Name" type="tsl:MultiLangNormStringType"
        maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="MultiLangNormStringType">
    <xsd:complexContent>
        <xsd:extension base="tsl:NonEmptyNormalizedString">
            <xsd:attribute ref="xml:lang" use="required"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="MultiLangStringType">
    <xsd:complexContent>
        <xsd:extension base="tsl:NonEmptyString">
            <xsd:attribute ref="xml:lang" use="required"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:simpleType name="NonEmptyString">
    <xsd:restriction base="xsd:string">
        <xsd:minLength value="1"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="NonEmptyNormalizedString">
    <xsd:restriction base="xsd:normalizedString">
        <xsd:minLength value="1"/>
    </xsd:restriction>
</xsd:simpleType>
```

B.1.2.2 The `AddressType` Type

This type is used for addresses holding postal addresses and electronic addresses.

```
<xsd:complexType name="AddressType">
    <xsd:sequence>
        <xsd:element ref="tsl:PostalAddresses"/>
        <xsd:element ref="tsl:ElectronicAddress"/>
    </xsd:sequence>
</xsd:complexType>
```

B.1.2.3 The `PostalAddresses` Element

The `PostalAddressses` element contains a list of `PostalAddress` element. Each `PostalAddress` element contains a postal address in a specific language and script identified by the `xml:lang` attribute.

```
<xsd:element name="PostalAddresses" type="tsl:PostalAddressListType"/>
<xsd:complexType name="PostalAddressListType">
    <xsd:sequence>
        <xsd:element ref="tsl:PostalAddress" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:element name="PostalAddress" type="tsl:PostalAddressType"/>
<xsd:complexType name="PostalAddressType">
    <xsd:sequence>
        <xsd:element name="StreetAddress" type="tsl:NonEmptyString"/>
        <xsd:element name="Locality" type="tsl:NonEmptyString"/>
        <xsd:element name="StateOrProvince" type="tsl:NonEmptyString" minOccurs="0"/>
        <xsd:element name="PostalCode" type="tsl:NonEmptyString"/>
        <xsd:element name="CountryName" type="tsl:NonEmptyString"/>
    </xsd:sequence>
    <xsd:attribute ref="xml:lang" use="required"/>
</xsd:complexType>
```

B.1.2.4 The ElectronicAddressType Type

The ElectronicAddressType Type allows the specification of one electronic address.

```
<xsd:element name="ElectronicAddress" type="tsl:ElectronicAddressType"/>
<xsd:complexType name="ElectronicAddressType">
    <xsd:sequence >
        <xsd:element name="URI" type="tsl:NonEmptyURIType" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
```

The contents of each URI element MUST represent either a RFC 2822 [14] e-mail address, expressed by using the "mailto:" URI scheme as defined by RFC 2368 [11], or a web site address.

B.1.2.5 Types for managing the extensions

The present document allows for extending the content of certain elements in TSLs. This clause defines the elements and types that will be used for such purposes. Below follow their schema definition.

```
<xsd:complexType name="AnyType" mixed="true">
    <xsd:sequence minOccurs="0" maxOccurs="unbounded">
        <xsd:any processContents="lax"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:element name="Extension" type="tsl:ExtensionType"/>
<xsd:complexType name="ExtensionType">
    <xsd:complexContent>
        <xsd:extension base="tsl:AnyType">
        <xsd:attribute name="Critical" type="xsd:boolean" use="required" />
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ExtensionsListType">
    <xsd:sequence>
        <xsd:element ref="tsl:Extension" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
```

AnyType type allows for any kind of content. ExtensionType is derived from AnyType by extension. Its Critical attribute indicates whether this element is critical or not. The ExtensionListType is an unbounded list of Extension elements.

Processing of Critical attribute MUST be as the one defined by RFC 3280 [17] for the critical field of extensions of X.509 v3 certificates. Applications MUST reject the TSL if they encounter a critical extension that they do not recognize. However, they MAY ignore a non-critical extension that they do not recognize.

B.1.2.6 Types for URIs

The present document defines new derived types from `xsd:anyURI`. Their schema definition is shown below.

```
<xsd:simpleType name="NonEmptyURIType">
    <xsd:restriction base="xsd:anyURI">
        <xsd:minLength value="1"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="NonEmptyMultiLangURIType">
    <xsd:complexContent>
        <xsd:extension base="tsl:NonEmptyURIType">
            <xsd:attribute ref="xml:lang" use="required"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="NonEmptyMultiLangURIListType">
    <xsd:sequence>
        <xsd:element name="URI" type="tsl:NonEmptyMultiLangURIType" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="NonEmptyURIListType">
    <xsd:sequence>
        <xsd:element name="URI" type="tsl:NonEmptyURIType" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
```

An element of `NonEmptyURIType` type contains a non empty URI value.

An element of `NonEmptyMultiLangURIType` contains a non empty URI value pointing to a resource written in the language that is signalled by the `xml:lang` attribute.

An element of `NonEmptyMultiLangURIListType` contains a sequence of non empty URI values pointing to a resource written in the language that is signalled by the `xml:lang` attribute.

An element of `NonEmptyURIListType` contains a sequence of non empty URI values.

B.1.3 The `TrustServiceStatusList` element

The `TrustserviceStatusList` element is the root element of an XML TSL. An implementation must generate *laxly schema valid* [XML-schema] `TrustserviceStatusList` elements as specified by the following schema.

```
<xsd:element name="TrustServiceStatusList" type="tsl:TrustStatusListType"/>
<xsd:complexType name="TrustStatusListType">
        <xsd:sequence>
            <xsd:element    ref="tsl:SchemeInformation"/>
            <xsd:element    ref="tsl:TrustServiceProviderList" minOccurs="0"/>
            <xsd:element    ref="ds:Signature"/>
        </xsd:sequence>
        <xsd:attribute name="TSLTag" type="tsl:TSLTagType" use="required"/>
        <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

The optional attribute Id may be used to make a reference to the TrustServiceStatusList element.

B.1.3.1 The `TSLTag` attribute

This REQUIRED attribute shall facilitate the identification of the TSL as such, when electronic searches are conducted across the Internet. It will be a string with a fixed value. Its schema definition follows.

```
<xsd:simpleType name="TSLTagType">
<xsd:restriction base="xsd:anyURI">
    <xsd:enumeration
    value="http://uri.etsi.org/02231/TSLTag"/>
</xsd:restriction>
</xsd:simpleType>
```

B.2 The SchemeInformation element

The SchemeInformation element is a container structure for all the elements giving detailed information about the scheme.

```
<xsd:element name="SchemeInformation" type="tsl:TSLSchemeInformationType"/>
<xsd:complexType name="TSLSchemeInformationType">
    <xsd:sequence>
        <xsd:element name="TSLVersionIdentifier" type="xsd:integer" fixed="2"/>
        <xsd:element name="TSLSequenceNumber" type="xsd:positiveInteger"/>
        <xsd:element name="TSLType" type="tsl:NonEmptyURIType"/>
        <xsd:element name="SchemeOperatorName" type="tsl:InternationalNamesType"/>
        <xsd:element name="SchemeOperatorAddress" type="tsl:AddressType"/>
        <xsd:element name="SchemeName" type="tsl:InternationalNamesType"/>
        <xsd:element name="SchemeInformationURI" type="tsl:NonEmptyMultiLangURIListType"/>
        <xsd:element name="tsl:StatusDeterminationApproach"
          type="tsl: NonEmptyURIType"/>
        <xsd:element name="SchemeTypeCommunityRules"
            type="tsl:NonEmptyURIListType" minOccurs="0"/>
        <xsd:element ref="tsl:SchemeTerritory" minOccurs="0"/>
        <xsd:element ref="tsl:PolicyOrLegalNotice" minOccurs="0"/>
        <xsd:element name="HistoricalInformationPeriod" type="xsd:nonNegativeInteger"/>
        <xsd:element ref="tsl:PointersToOtherTSL"  minOccurs="0"/>
        <xsd:element name="ListIssueDateTime" type="xsd:dateTime"/>
        <xsd:element ref="tsl:NextUpdate"/>
        <xsd:element name="SchemeExtensions" type="tsl:ExtensionsListType" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
```

B.2.1 The TSLVersionIdentifier element

This REQUIRED element specifies the version of the TSL format. In this version of the TSL it must have the value "2".

B.2.2 The TSLSequenceNumber element

This REQUIRED element specifies the sequence number of the TSL. At the first release of the TSL, the value of the sequence number shall be "1". The value shall be increased by "1" at each subsequent release of the TSL.

B.2.3 The TSLType element

This REQUIRED element specifies the type of the TSL. Its values are URIs as those listed in clause D.2 or another ones registered and described by the scheme operator or another entity.

B.2.4 The SchemeOperatorName element

This REQUIRED element specifies the name(s) under which the scheme operator does business or is given its mandate.

B.2.5 The SchemeOperatorAddress element

This REQUIRED element contains the address of the scheme operator.

B.2.6 The SchemeName element

This REQUIRED element specifies the name(s) under which the scheme operates.

B.2.7 The SchemeInformationURI element

This REQUIRED element contains the URIs where users can obtain scheme-specific information.

B.2.8 The StatusDeterminationApproach element

This REQUIRED element specifies the status determination approach (see clause 5.3.8). Its value may be one of the URIs listed in clause D.2 or any other URI value registered and described by the scheme operator or another entity.

B.2.9 The SchemeTypeCommunityRules element

This OPTIONAL element contains a sequence of registered URIs, used as unique identifier when it is required to indicate one or more sets of rules/policies under which the TSL has been issued.

B.2.10 The SchemeTerritory element

This OPTIONAL element specifies the country in which the scheme is established. See clause 5.3.10 for a discussion of its contents. Its schema definition follows.

```
<xsd:element name="SchemeTerritory" type="tsl:SchemeTerritoryType"/>
<xsd:simpleType name="SchemeTerritoryType">
    <xsd:restriction base="xsd:string">
        <xsd:length value="2"/>
    </xsd:restriction>
</xsd:simpleType>
```

B.2.11 The PolicyOrLegalNotice element

This OPTIONAL element MAY be used to specify the scheme's policy or provide a notice concerning the legal status of the scheme or legal requirements met by the scheme for the jurisdiction in which the scheme is established and/or any constraints and conditions under which the TSL is maintained and offered. It can be provided in multiple languages. This string is either recognized as a registered URI or represents the textual form of the legal notice. Its schema definition follows.

```
<xsd:element name="PolicyOrLegalNotice" type="tsl:PolicyOrLegalnoticeType"/>
<xsd:complexType name="PolicyOrLegalnoticeType">
    <xsd:choice>
        <xsd:element name="TSLPolicy" type="tsl:NonEmptyMultiLangURIType"
            maxOccurs="unbounded"/>
        <xsd:element name="TSLLegalNotice" type="tsl:MultiLangStringType"
            maxOccurs="unbounded"/>
</xsd:choice>
</xsd:complexType>
```

B.2.12 The HistoricalInformationPeriod element

This REQUIRED element contains the duration over which historical information in this TSL is provided (see clause 5.3.12).

B.2.13 The PointersToOtherTSL element

This OPTIONAL element specifies URIs where users can obtain other TSLs. The OtherTSLPointersType specifies a list of OtherTSLPointer elements, each holding a URI pointing to the TSL and additional information about that TSL, which is implementation-specific.

```
<xsd:element name="PointersToOtherTSL" type="OtherTSLPointersType"/>
<xsd:complexType name="OtherTSLPointersType">
    <xsd:sequence>
        <xsd:element ref="OtherTSLPointer" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:element name="OtherTSLPointer" type="tsl:OtherTSLPointerType"/>
<xsd:complexType name="OtherTSLPointerType">
    <xsd:sequence>
        <xsd:element name="TSLLocation" type="tsl:NonEmptyURIType"/>
        <xsd:element ref="tsl:AdditionalInformation"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:element name="AdditionalInformation" type="tsl:AdditionalInformationType"/>
<xsd:complexType name="AdditionalInformationType">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="TextualInformation" type="tsl:MultiLangStringType"/>
        <xsd:element name="OtherInformation" type="tsl:AnyType"/>
    </xsd:choice>
</xsd:complexType>
```

The AdditionalInformation element may contain a textual information within the TextualInformation element or any other type of information qualifying the pointed TSL, within the element OtherInformation.

B.2.14 The ListIssueDateTime element

This REQUIRED element specifies the date and time of the issuance of the TSL.

B.2.15 The NextUpdate element

This REQUIRED element specifies the latest date and time by which the TSL will be next issued. Its schema definition follows.

```
<xsd:element name="NextUpdate" type="tsl:NextUpdateType"/>
<xsd:complexType name="NextUpdateType">
    <xsd:sequence>
        <xsd:element name="dateTime" type="xsd:dateTime" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
```

The NextUpdate element MAY be an empty element. This MUST occur when a scheme manager issues its last TSL before ceasing its activities. An empty NextUpdate element indicates that this will be the last issuance of a TSL by the scheme manager.

B.2.16 The SchemeExtensions element

This OPTIONAL element allows for the inclusion of additional information on a scheme. The specific content of such additional information is left open.

B.2.17 The TrustServiceProviderList element

This element contains all the information related to all the TSPs recognized by the scheme. It is a list of TrustServiceProvider elements, each one containing all the information related to one TSP. If present it SHALL contain at least one TrustServiceProvider element. For each TSP, the list of services offered is REQUIRED and at least on service MUST be listed. Their schema definitions follow.

```
<xsd:element name="TrustServiceProviderList" type="tsl:TrustServiceProviderListType"/>
<xsd:complexType name="TrustServiceProviderListType">
    <xsd:sequence>
        <xsd:element ref="tsl:TrustServiceProvider" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:element name="TrustServiceProvider" type="tsl:TSPType"/>
<xsd:complexType name="TSPType">
    <xsd:sequence>
        <xsd:element ref="tsl:TSPInformation"/>
        <xsd:element ref="tsl:TSPServices"/>
    </xsd:sequence>
</xsd:complexType>
```

B.3 The TSPInformation element

The TSPInformation element has the following structure.

```
<xsd:element name="TSPInformation" type="tsl:TSPInformationType"/>
<xsd:complexType name="TSPInformationType">
    <xsd:sequence>
        <xsd:element name="TSPName" type="tsl:InternationalNamesType"/>
        <xsd:element name="TSPTradeName" type="tsl:InternationalNamesType"
                minOccurs="0"/>
        <xsd:element name="TSPAddress" type="tsl:AddressType"/>
        <xsd:element name="TSPInformationURI"
                      type="tsl:NonEmptyMultiLangURIListType"/>
        <xsd:element name="TSPInformationExtensions" type="tsl:ExtensionsListType"
              minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
```

B.3.1 The TSPName element

This REQUIRED element contains the name of the TSP.

B.3.2 The TSPTradeName element

This OPTIONAL element contains alternative trading names of the TSP.

B.3.3 The TSPAddress element

This REQUIRED element contains the address of the TSP.

B.3.4 The TSPInformationURI element

This REQUIRED element contains a pointer to a web page holding service-specific information.

B.3.5 The TSPInformationExtensions element

This OPTIONAL element allows for the inclusion of additional information on a Trust Services Provider. The specific content of such additional information is left open.

B.3.6 The TSPServices element

This element contains information of a list of Trust Services the TSP offers. It is a sequence of TSPService elements, whose contents are described with detail in clause B.4.

```
<xsd:element name="TSPServices" type="tsl:TSPServicesListType"/>
<xsd:complexType name="TSPServicesListType">
    <xsd:sequence>
        <xsd:element ref="tsl:TSPService"maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:element name="TSPService" type="tsl:TSPServiceType"/>
<xsd:complexType name="TSPServiceType">
    <xsd:sequence>
        <xsd:element ref="tsl:ServiceInformation"/>
        <xsd:element ref="tsl:ServiceHistory" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
```

B.4 The ServiceInformation element

The ServiceInformation element is a container element holding information about a specific service.

```
<xsd:element name="ServiceInformation" type="tsl:TSPServiceInformationType"/>
<xsd:complexType name="tsl:TSPServiceInformationType">
    <xsd:sequence>
        <xsd:element ref="tsl:ServiceTypeIdentifier"/>
        <xsd:element name="ServiceName" type="tsl:InternationalNamesType"/>
        <xsd:element ref="tsl:ServiceDigitalIdentity"/>
        <xsd:element ref="tsl:ServiceStatus"/>
        <xsd:element name="StatusStartingTime" type="xsd:dateTime"/>
        <xsd:element name="SchemeServiceDefinitionURI"
            type="tsl:NonEmptyMultiLangURIListType" minOccurs="0"/>
        <xsd:element ref="tsl:ServiceSupplyPoints" minOccurs="0"/>
        <xsd:element name="TSPServiceDefinitionURI"
            type="tsl:NonEmptyMultiLangURIListType" minOccurs="0"/>
        <xsd:element name="ServiceInformationExtensions"
            type="tsl:ExtensionsListType" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
```

B.4.1 The ServiceTypeIdentifier element

This REQUIRED element specifies the identifier of the service type. Its value may be one of the URIs listed in clause D.2 or any other URI value registered and described by the scheme operator or another entity.

```
<xsd:element name="ServiceTypeIdentifier" type="tsl: NonEmptyURIType"/>
```

B.4.2 The ServiceName element

This REQUIRED element specifies the name under which the service is provided.

B.4.3 The ServiceDigitalIdentity element

This is a REQUIRED field. This element MAY be empty or contain a number of several elements. Each element contains alternative information for identifying the same service. When identification is based on a public key they borrow their contents from XML-Signature [34] specification. In these cases implementations MAY use one or several of the following three representations for a key:

1) A ds:Keyvalue element.

2) The X509SKI element.

3) The `X509Certificate` element.

Implementations MAY also use a Distinguished Name (DN).

Applications MUST implement the `X509Certificate`,the `X509SKI` and X509SubjectName elements exactly as specified in XML-Signature [34] when they use them. Element:`X509SubjectName` will contain a Distinguished Name encoded as established by XML-Signature [34] in its clause 4.4.4.

The `X509SKI` element MAY be used only if there exists a X.509 certificate whose subject is the service to be identified. In this case, the content of this element MUST be the same as the content of the `SubjectKeyIdentifier` extension of the aforementioned certificate.

The number of elements required for identifying a service depends on the TSP policy as well as of the relying party, and any further consideration on this topic are beyond the scope of the present document.

In addition, implementations MAY use other values for element making use of the `Other` element, whose contents are left open.

```
<xsd:element name="ServiceDigitalIdentity" type="tsl:DigitalIdentityListType"/>
<xsd:complexType name="DigitalIdentityListType">
    <xsd:sequence>
        <xsd:element name="DigitalId" type="tsl:DigitalIdentityType" minOccurs="0"
        maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="DigitalIdentityType">
    <xsd:choice>
        <xsd:element name="X509Certificate" type="xsd:base64Binary"/>
        <xsd:element name="X509SubjectName" type="xsd:string"/>
        <xsd:element ref="ds:KeyValue" />
        <xsd:element name="X509SKI" type="xsd:base64Binary"/>
        <xsd:element name="Other" type="tsl:AnyType"/>
    </xsd:choice>
</xsd:complexType>
```

B.4.4 The `ServiceStatus` element

This REQUIRED element specifies the identifier of the status of the service. See clause 5.5.4 for an explanation of its contents. Its schema definition follows. Its value may be one of the URIs listed in clause D.2.

```
<xsd:element name="ServiceStatus" type="tsl:NonEmptyURIType"/>
```

B.4.5 The `StatusStartingTime` element

This REQUIRED element specifies the date and time on which the current status became effective.

B.4.6 The `SchemeServiceDefinitionURI` element

This OPTIONAL element specifies the URI where users can obtain service-specific information provided by the scheme operator.

B.4.7 The `ServiceSupplyPoints` element

This element contains a sequence of `ServiceSupplyPoint` elements, each one being a non-empty URI that points to the place where users and relying parties may gain access to the service.

```
<xsd:element name="ServiceSupplyPoints" type="tsl:ServiceSupplyPointsType"/>
<xsd:complexType name="ServiceSupplyPointsType">
    <xsd:sequence maxOccurs="unbounded">
        <xsd:element name="ServiceSupplyPoint" type="tsl:NonEmptyURIType"/>
    </xsd:sequence>
</xsd:complexType>
```

B.4.8 The TSPServiceDefinitionURI element

This OPTIONAL field specifies the URI where users can obtain service-specific information provided by the TSP.

B.4.9 The ServiceInformationExtensions element

This OPTIONAL element allows for the inclusion of additional information on a service. The specific content of such additional information is left open.

B.4.10 The ServiceHistory element

This OPTIONAL field provides any historical status information.

```
<xsd:element name="ServiceHistory" type="tsl:ServiceHistoryType"/>
```

B.5 The ServiceHistory type

This element is a sequence of ServiceHistoryInstance elements. Each one has a content as specified in clause 5.6 and equivalent to the information contained in clause 5.5 with the addition of the ServiceInformationExtensions element. For XML, the relevant fields have been specified in clauses B.4.1 through B.4.5 (representing clauses 5.6.1 through 5.6.6 as well as clauses 5.5.1 through 5.5.5 inclusive, and clause 5.5.9). The ServiceInformationExtensions element is already specified in clause B.4.9.

This element MAY be present or absent. If present it MAY be empty, for signalling that so far no history has been yet built. Its schema definition follows.

```
<xsd:element name="ServiceHistory" type="tsl:ServiceHistoryType"/>
<xsd:complexType name="ServiceHistoryType">
    <xsd:sequence>
        <xsd:element ref="tsl:ServiceHistoryInstance" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:element name="ServiceHistoryInstance" type="tsl:ServiceHistoryInstanceType"/>
<xsd:complexType name="ServiceHistoryInstanceType">
    <xsd:sequence>
        <xsd:element ref="tsl:ServiceTypeIdentifier"/>
        <xsd:element name="ServiceName" type="tsl:InternationalNamesType"/>
        <xsd:element ref="tsl:ServiceDigitalIdentity"/>
        <xsd:element ref="tsl:ServiceStatus"/>
        <xsd:element name="StatusStartingTime" type="xsd:dateTime"/>
        <xsd:element name="ServiceInformationExtensions" type="tsl:ExtensionsListType"
        minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
```

B.6 The Signature element

The present document allows the use of XML-Signature [34] based signatures for signing a TSL: this includes use of TS 101 903 [35] signatures (see clause F.3 for further discussion). The TSL-structure contains a ds:Signature element that represents an enveloped signature-type. The present document mandates the following constraints to any XML-Signature [34]-based signature applied to a TSL:

1) It MUST be an enveloped signature.

2) Its ds:SignedInfo element MUST contain a ds:Reference element with the URI attribute set to a value referencing the TrustedStatusList element enveloping the signature itself. This ds:Reference element MUST satisfy the following requirements:

    a) It MUST contain only one ds:Transforms element.

b) This `ds:Transforms` element MUST contain two `ds:Transform` elements. The first one will be one whose `Algorithm` attribute indicates the enveloped transformation with the value: "http://www.w3.org/2000/09/xmldsig#enveloped-signature". The second one will be one whose Algorithm attribute instructs to perform the exclusive canonicalization "http://www.w3.org/2001/10/xml-exc-c14n#"

3) ds:CanonicalizationMethod MUST be "http://www.w3.org/2001/10/xml-exc-c14n#".

4) It MAY have other `ds:Reference` elements.

Rules 2 and 3 ensure that the enveloping `TrustServiceStatusList` element is actually signed as mandated by the processing model in clause 4.3.3.3 of XML-Signature [34] (with reference to same-document URI references). They also ensure that if relative referencing mechanisms are used in the `ds:Reference` element, the `TrustServiceStatusList` may be safely inserted within other xml documents.

Rule 4 allows, among other things, for inclusion of signed properties in the signature, like the ones standardized in TS 101 903 [35].

B.6.1 The scheme identification

As stated in clause 5.7.2, in a signed TSL the signature MUST also cover the scheme identification. This requirement may be fulfilled by standard mechanisms provided by both XML-Signature [34] and TS 101 903 [35].

When a plain XML-Signature [34] signature is generated, one of the following elements MUST be present within the `ds:KeyInfo`'s child element, `ds:X509Data`: a `ds:X509Certificate` element containing an X.509 certificate ITU-T Recommendation X.509 [26], a `ds:X509SKI` element containing the `SubjectKeyIdentifier` extension, or an XML element containing a public key as established within XML-Signature [34] (for RSA and DSA public keys) or the corresponding specification (as new XML formats for carrying public key information are defined, such as that in RFC 4050 [20] for Elliptic Curve Algorithm public keys).

B.6.1.1 The scheme operator identifier in XAdES signatures

TS 101 903 [35] defines the `xades:SigningCertificate` as a signed property that contains an identifier of the signer's certificate and its digest. This is therefore a effective way of securing the scheme operator identifier (see clause F.3 for further discussion).

Even when the `xades:SigningCertificate` property is present, the current document does not prevent the inclusion of any of the three elements mentioned in the previous section within the `ds:KeyInfo`'s child element `ds:X509Data`.

Should a `ds:X509Certificate` containing the signer's certificate be present within a XAdES signature as a child of a `ds:X509Data` within `ds:KeyInfo`, its serial number and issuer identifier MUST match the serial number and issuer identifier present in the `xades:SigningCertificate` signed property.

Should the child of ds:X509Data element be a `ds:X509SKI` or an element encapsulating a public key, its contents MUST be consistent with the contents of the `xades:SigningCertificate` signed property, if present.

B.6.2 Algorithm and parameters

The algorithms, their parameters and formats supported by the present document are those supported by XML-Signature [34]. Further algorithms, parameters and their format MAY be specified elsewhere, e.g. as for the Elliptic Curve Signature Algorithm (ECDSA) in RFC 4050 [20].

Annex C (normative): ASN.1 and XML files

C.1 Electronic attachment

This document has an associated electronic document "ts_102231v020101p0.zip" that contains the ASN.1 module and XML and LDAP schemas that are integral parts of this specification and further described below.

CAVEAT: In the event that any part of the module and/or schemas within this electronic attachment are in conflict with the text of either annexes A or B, then those Annexes shall prevail as the authoritative sources.

C.2 ASN.1 module

The ASN.1 module is held in file "ts_102231v020101asn.asn". For the purpose of integrity checking, the hash values of this file are:

MD-5: 83e59341f28ea57a81df0489e274996f

SHA-1: 2ef04fc7ff4d6ff662df365ccf4bdc23eaafc870

C.3 XML schema

This XML schema is held in file "ts_102231v020101xsd.xsd". For the purpose of integrity checking, the hash values of this file are:

MD-5: db97ae004d2b2fa7ccc8a818d9a7a5bc

SHA-1: 638f9a9256a955b96b5cca213b75b3fa3586e43e

C.4 LDAP schema

This XML schema is held in file "ts_102231v020101sch.schema". For the purpose of integrity checking, the hash values of this file are:

MD-5: c31056379e840c614f35ce89078dbca9

SHA-1: 10bb3d8a7809167d8c2a3cd380da7fbb62bcafb4

Annex D (normative): Registered Uniform Resource Identifiers

This annex specifies those uniform resource identifiers (URIs) which have been registered in connection with the present document. Those with the radix (base) "http://uri.etsi.org/02231/……" are registered and declared by their presence in the present document, for specific usage within this document: those with the radix "http://uri.etsi.org/TrstSvc/……" are registered by ETSI as a Common Domain (ref. http://portal.etsi.org/ptcc/xml.asp#Common%20Domain) on behalf of the TC ESI because they have a wider applicability and usage and are listed here for the convenience of users of the present document.

Where URI's registered on behalf of the TC ESI are used within the specifications of TSL fields (clause 5) it is generally the case that users can register other URIs for their own purposes and extend the range of that field, although it is strongly RECOMMENDED that the scheme operator makes a clear declaration of the meaning of that URI. Refer to clause 5.2 and onwards.

In the following tables the following layout is used for each URI declaration:

The URI is given as an unbroken string	Related TSL field (if any)
The meaning of the URI is given, indented to emphasise its relationship to the preceding URI.	

Where more than one URI relates to a specific TSL field the second column will extend across all URI declarations (row-pairs) which apply.

D.1 URIs registered within the present document

The following URIs are hereby declared and registered under the present document's assigned radix:

http://uri.etsi.org/02231/v2.1.1	N/a
This issue of the ETSI Technical Specification 102 231and its related parts.	

http://uri.etsi.org/02231/TSLtag	TSL tag
A data structure which conforms to the TSL specification published in ETSI Technical Specification 102 231, in any of its historical issues or this one.	

http://uri.etsi.org/02231/v2#	N/a
The XML namespace identifier relating to the TSL version specified in this issue of ETSI Technical Specification 102 231.	

http://uri.etsi.org/02231/TDPContainer	N/a
A qualifier for web pages that contain one or more TDPs which can be used as a value of the attribute "profile" for the "head" element of the web page.	

D.2 ETSI Common Domain URIs

The following URIs have been declared and registered by ETSI under the Technical Committee Electronic Signatures Infrastructure's (TC ESI) assigned radix:

http://uri.etsi.org/TrstSvc/TSLtype/generic	TSL type
A TSL of trust services which are approved or recognized by the scheme operator owning the TSL through a process of direct oversight (whether voluntary or regulatory).	
http://uri.etsi.org/TrstSvc/TSLtype/schemes	
A TSL of other assessment schemes which are independently responsible for the approval or recognition by a community of trust services through a process of direct oversight (whether voluntary or regulatory).	

http://uri.etsi.org/TrstSvc/TSLtype/StatusDetn/active	Status determination approach (see note)
Services listed have their status determined after assessment by or on behalf of the scheme operator against the scheme's criteria (active approval/recognition).	
http://uri.etsi.org/TrstSvc/TSLtype/StatusDetn/passive	
Services listed have been nominated by their provider or are known to be operating in the marketplace, but have not undergone assessment by or on behalf of the scheme operator for initial approval (passive approval/recognition).	
http://uri.etsi.org/TrstSvc/TSLtype/StatusDetn/delinquent	
Services listed have been deemed to be non-compliant with scheme criteria.	
http://uri.etsi.org/TrstSvc/TSLtype/StatusDetn/null	
No predetermined criteria.	
NOTE: In the case of meanings "active" and "passive", a scheme could include in the TSL both services and schemes whose current status is approved/ recognized (either actively or passively, but each indicating a positive assertion) and those which have failed to meet the criteria. In the case of meaning "delinquent", the TSL would list only those services which had explicitly failed to fulfil the criteria of the scheme (i.e. had exhibited delinquency). It is therefore unlikely that such a status determination approach would include other schemes, although this could be determined by the scheme operator's rules.	

http://uri.etsi.org/TrstSvc/Svctype/CA/PKC	Service type identifier
A Certification authority issuing public key certificates.	
http://uri.etsi.org/TrstSvc/Svctype/CA/QC	
A Certification authority issuing Qualified Certificates.	
http://uri.etsi.org/TrstSvc/Svctype/TSA	
A Time stamping authority.	
http://uri.etsi.org/TrstSvc/Svctype/Certstatus/OCSP	
A Certificate status provider operating an OCSP-server.	
http://uri.etsi.org/TrstSvc/Svctype/Certstatus/CRL	
A Certificate status provider operating a CRL.	
http://uri.etsi.org/TrstSvc/Svctype/RA	
A Registration authority.	
http://uri.etsi.org/TrstSvc/Svctype/IdV	
An Identity verification service.	
http://uri.etsi.org/TrstSvc/Svctype/CGen	
A Certificate generation service which responds to requests for certificate generation from an authenticated source of identity information.	
http://uri.etsi.org/TrstSvc/Svctype/ACA	
An Attribute certification authority.	
http://uri.etsi.org/TrstSvc/Svctype/Archiv	
An Archival service.	
http://uri.etsi.org/TrstSvc/Svctype/KEscrow	
A Key escrow service.	
http://uri.etsi.org/TrstSvc/Svctype/PPwd	
Issuer of PIN- or password-based identity credentials.	
http://uri.etsi.org/TrstSvc/Svctype/ECsupervision	
An assessment scheme which is a system of supervision as defined in, and which complies with all applicable requirements of Directive 1999/93/EC [1].	
http://uri.etsi.org/TrstSvc/Svctype/ECvoluntary	
An assessment scheme which is a voluntary approval [accreditation] scheme as defined in, and which complies with all applicable requirements of Directive 1999/93/EC [1].	
http://uri.etsi.org/TrstSvc/Svctype/unspecified	
A trust service of an unspecified type.	

http://uri.etsi.org/TrstSvc/Svcstatus/inaccord	Service current status
The subject service is in accordance with the scheme's specific status determination criteria (*only for use in positive approval schemes*).	
http://uri.etsi.org/TrstSvc/Svcstatus/expired	
The subject service is no longer overseen by the scheme, e.g. due to non-renewal or withdrawal by the TSP, or cessation of the service **or** the scheme's operations.	
http://uri.etsi.org/TrstSvc/Svcstatus/suspended	
The subject service's status is temporarily uncertain whilst checks are made by the scheme operator (typically e.g. while a revocation request is being investigated or if action is required to resolve a deficiency in the service fulfilling the scheme's criteria.	
http://uri.etsi.org/TrstSvc/Svcstatus/revoked	
The subject service's approved status has been revoked because it is no longer in accordance with the scheme's specific status determination criteria (*only for use in positive approval schemes*).	
http://uri.etsi.org/TrstSvc/Svcstatus/notinaccord	
The subject service is ***not*** in accordance with the scheme's specific status determination criteria (*only for use in negative approval schemes*).	

http://uri.etsi.org/TrstSvc/schemerules/Dir-1999-93-EC/supervision	Scheme type/community/rules (at the primary level)
An assessment scheme which is a system of supervision as defined in, and which complies with all applicable requirements of Directive 1999/93/EC [1].	
http://uri.etsi.org/TrstSvc/schemerules/Dir-1999-93-EC/volapproval	
An assessment scheme which is a voluntary approval [accreditation] scheme as defined in, and which complies with all applicable requirements of Directive 1999/93/EC [1].	

http://uri.etsi.org/TrstSvc/schemerules/Dir-1999-93-EC/supervision/"EUMS"	Scheme type/community/rules (at the secondary level)
An assessment scheme which is a system of supervision as defined in, and which complies with all applicable requirements of Directive 1999/93/EC [1], and which is established in the EU Member State "EUMS", where "EUMS" is replaced by the applicable country name as a two-character code in accordance with ISO 3166-1 [21] Alpha-2 code.	
http://uri.etsi.org/TrstSvc/schemerules/Dir-1999-93-EC/volapproval/"EUMS"	
An assessment scheme which is a voluntary approval [accreditation] scheme as defined in, and which complies with all applicable requirements of Directive 1999/93/EC [1], and which is established in the EU Member State "EUMS", where "EUMS" is replaced by the applicable country name as a two-character code in accordance with ISO 3166-1 [21] Alpha-2 code.	

D.3 Registering additional URIs

Any organization operating a scheme might choose to create its own URIs for its own specific purposes or request ETSI to assign a registered URI root under the ETSI Identified Organization Domain, and then define its own URIs. It might be appropriate to register certain of those URIs where they complement URIs required by or which might be used in the context of the publication of a TSL. The following examples suggest how additional URIs could be created, including showing a second level of rules, after using the applicable Optional URI as shown above:

Potential URI **Meaning**	**Related TSL field (if any)**
http://uri.etsi.org/"registered_org"/schemerules/Dir-1999-93-EC/volapproval/"schemename"	Scheme type/community/rules (at the secondary level)
This could mean an assessment scheme called "schemename" being operated by "registered_org", where "registered_org" is replaced by the name of the scheme operator and "schemename" is replaced by the actual scheme name, which is a voluntary approval scheme as defined in and which complies with all applicable requirements of Directive 1999/93/EC [1] and which is established in an EU Member State (note that because voluntary schemes are not restricted to a single EU Member State's territory, there may be no need to indicate the State in which the scheme is established - this would be a matter of choice for the registering scheme).	
http://"scheme_op_URI_root"/.../schemerules/Dir-1999-93-EC/volapproval/"schemename"	
This URI would be registered under a different root, e.g. the scheme operator's, distinguished by "scheme_op_URI_root", or it could be another organization which maintains a registry of URIs. This URI could mean an assessment scheme called "schemename" being operated by "scheme_op" where "scheme_op" is replaced by the name of the scheme operator and "schemename" is replaced by the actual scheme name, which is a voluntary approval scheme as defined in and which complies with all applicable requirements of Directive 1999/93/EC [1] and which is established in an EU Member State (see note).	
NOTE: Because voluntary schemes are not restricted to a single EU Member State's territory, there may be no need to indicate the State in which the scheme is established - this would be a matter of choice for the registering scheme.	

Annex E (normative): Implementation notes for multilingual support

E.1 Multilingual character string

The string contained within a multilingual character string SHALL fulfil the requirements of annex N of ISO 10646 [23] subject to the following restrictions:

1) the content SHALL be a string of characters from the Universal Character Set (UCS) as defined by ISO 10646 [23];

2) the content MUST be UTF-8 encoded;

3) the content MUST NOT include any signature to identity the UCS (see annex H of ISO 10646 [23]);

4) control functions (ISO/IEC 6429 - see Bibliography), escape sequences (ISO/IEC 2022 - see Bibliography) and control sequences or strings MUST NOT be used; therefore control characters such as TAB, CR, LF MUST NOT be present;

5) private-use characters (see clause 10 of ISO 10646 [23]) from the private use zone (code points E000 to F8FF) in the Basic Multilingual Plane (BMP) and from the private-use Planes 0F and 10 in Group 00, SHALL NOT be used;

6) Tag Characters (see Annex T of ISO 10646 [23]) MUST NOT to be used: therefore the characters from the TAGS (3001) collection MUST not be used (see Annex A of ISO 10646 [23] for the list of defined collections);

7) the content SHALL be plain text without any mark-up elements or tags from languages as SGML, HTML, XML, XHTML, RTF, TeX and others;

8) it is RECOMMENTED that the content follows the semantic rules defined by UNICODE version 4.00 for the corresponding characters;

9) combining characters SHOULD NOT be used if the content can be expressed without them; if there is the need to use combining characters but it is possible not to use the ones listed in Annex B.1 of ISO 10646 [23], then that latter set MUST NOT be used (this helps to keep as low as possible the required implementation level (as defined by clause 14 of ISO 10646 [23]) for parsing applications.

E.2 Multilingual pointer

If the content pointed by the multilingual pointer is plain text, it SHALL meet the following requirements that express the conformity to ISO 10646 [23] according to the annex N of ISO 10646 [23] and add further restrictions:

1) the pointed content SHALL be a string of characters from the Universal Character Set (UCS) as defined by ISO 10646 [23];

2) the pointed-to content MUST be UTF-8 encoded;

3) the pointed-to content MAY include the signature for UTF-8 (see annex H of ISO 10646 [23]) to identify the UCS;

4) control functions (ISO/IEC 6429 - see Bibliography), escape sequences (ISO/IEC 2022 - see Bibliography) and control sequences or strings MAY be used;

5) private-use characters (see clause 10 of ISO 10646 [23]) from the private use zone (code points E000 to F8FF) in the Basic Multilingual Plane (BMP) and from the private-use Planes 0F and 10 in Group 00, SHALL NOT be used;

6) Tag Characters (see annex T of ISO 10646 [23]) MUST NOT to be used: therefore the characters from the TAGS (3001) collection MUST not be used (see annex A of ISO 10646 [23] for the list of defined collections);

7) if the pointed-to content is expressed by means of mark-up languages as SGML, HTML, XML, XHTML then:

   a) the requirements described in W3C Technical Report #20 [33] are RECOMMENDED;

   b) a language indication MAY be present according to the mechanisms listed in W3C Technical Report #20 [33].

8) it is RECOMMENDED that the pointed-to content follows the semantic rules defined by UNICODE version 4.00 for the corresponding characters;

9) combining characters SHOULD NOT be used if the pointed-to content can be expressed without them; if there is the need to use combining characters but it is possible not to use the ones listed in annex B.1 of ISO 10646 [23], then that latter set MUST NOT be used (this helps to keep as low as possible the required implementation level (as defined by clause 14 of ISO 10646 [23]) for parsing applications).

E.3 Overall requirements

For the XML implementation of a TSL, it is RECOMMENDED that the requirements of W3C Technical Report #20 [33] be met.

For interoperability purposes, all applications parsing TSLs MUST be able to store and manage all characters defined by ISO 10646 [23]. This way the digital signature applied to the TSL can be always verified, whatever UCS characters are used within the TSL. However the parsing application may not be able to correctly present all characters.

NOTE: Developers of TSL parsing applications are advised that if their application does not support some of these characters, the application SHOULD give notice to the user about possible incorrect representation of the content of multilingual fields; the precise behaviour of the application while presenting unsupported characters is left to developers.

Annex F (informative): TSL Signing considerations

Although this annex is informative implementers are strongly recommended to satisfy the guidance which it provides, if not immediately, then as soon as suitable applications are available.

F.1 Signing application maturity

The present document requires that, when signing a TSL, the signer's certificate is bound into the signature. The most reasonable means to accomplish this is by using the SigningCertificate signed attribute (or property) available in TS 101 733 [2] or TS 101 903 [35] signatures.

However, at the time of publication of The present document, TSL Implementers face the situation that they may not have access to an implementation of TS 101 733 [2] or TS 101 903 [35]. Only a single CAdES implementation was known to be available, and those implementations of TS 101 903 [35] were not up to date with the most recently published XAdES version. These and other reasons suggest that there is not a maturity in the availability of suitable digital signature implementations and therefore The present document allows for alternatives.

F.2 CMS/ESS and CAdES

The present document supports two options to accomplish binding the certificate into the signature:

1) Basic CMS signatures with the addition of an ESS feature.

   For CMS-Signatures RFC 3852 [37] (see clause A.6), using the SigningCertificate signed attribute defined in RFC 2634 [13] fulfils the requirement of signing the signing identifier together with the TSL. This attribute is one of the two possible options for the implementation of this requirement for a TS 101 733 [2] signature; a CMS signature that contains this attribute with the profile specified in clause A.6.2.1 is also a -signature compliant with TS 101 733 [2] (a CAdES-BES).

2) TS 101 733 [2] signatures that are CMS signatures using advanced security features.

   As an alternative the present document allows for using the OtherCertificate signed attribute (see clause A.6.2.2) defined in CAdES.

Applications supporting TSLs are recommended to implement option 1 with immediate effect. Option 2 should be used only in contexts where is known that all parties use CAdES compliant applications., even if supported by the present document.

Instead, in contexts where none of or few parsing applications compliant with TS 101 733 [2] are used, it is recommended to generate only basic signatures compliant with CMS and ESS (i.e. option 1). Since these basic signatures are also compliant with TS 101 733 [2], applications supporting TS 101 733 [2] would be able to completely parse and verify these "basic signatures".

In the case of contexts where applications compliant to both basic CMS/ESS signatures and TS 101 733 [2] are used, if a TSL is signed by using the advanced features provided by TS 101 733 [2], the implementations that support only CMS/ESS but not the advanced features of TS 101 733 [2] will be still able to verify the TS 101 733 [2] signature calculated over the TSL and the TS 101 733 [2] signed attributes, but probably they wouldn't be able to understand any of the attributes present other than those supported by CMS/ESS. Therefore the CMS/ESS implementation won't be able to exploit/check the advanced security services provided by TS 101 733 [2], but the possibility to use the basic service (i.e. verify the signature over the TSL) will be always retained.

F.3 XML

Using XML, applications not supporting TS 101 903 [35] are advised to put the signing certificate into the KeyInfo element and add a reference to this into the signature. This is the standard XML-Signature [34] way to have an element included within the signature. Such applications are encouraged to ensure they will not refuse a TSL whose TS 101 903 [35] signature contains elements unknown to the application.

If an implementation supports TS 101 903 [35] signatures, it is recommended that the xades:SigningCertificate element is included in xades:SignedSignatureProperties. Adding the reference to ds:KeyInfo is not necessary and in fact is discouraged, although, as acknowledged in Annex B, ds:KeyInfo itself may be present. Such implementations should be flexible enough to accept TSLs signed without TS 101 903 [35].

If an implementation supports TS 101 903 [35] signatures, it is recommended that the SigningCertificate element is included in SignedSignatureProperties. Adding the reference to KeyInfo is not necessary and in fact is discouraged. Such implementations should be flexible enough to accept TSLs signed without TS 101 903 [35].

Annex G (informative): Management and Policy considerations

The TSL is a mechanism which is supporting of electronic transactions but not essential for them. There remains a variety of different models on which schemes may operate and a variance in how information from TSLs may be interpreted. Because of this lesser degree of dependence upon the TSL, the need to keep up to date information within a TSL is less urgent than that for, e.g. a CRL.

Scheme operators should publish their specific criteria for the provision of revisions to TSL information. These revisions will fall into the following categories.

G.1 Change of scheme administrative information

This category includes any changes to information concerning the scheme and which is embedded within the TSL. Such changes could include, *inter alia*, change of scheme addresses, revisions to acceptance criteria, scheme policy. When these change the TSL should be re-issued.

If there are material changes to information directly referenced through the TSL but the reference itself does not change then there will be no need to amend the TSL.

Any changes in this category should not affect the status information concerning any trust services mentioned within the TSL.

If the changes were the result of a change of ownership of the entity operating the scheme then the scheme could continue to operate without change or the scheme could cease operations and re-establish itself as a new scheme. It would be for the operators to determine how they wanted to handle this and how they would deal with the handling of services recognized under the scheme.

G.2 Change of TSP administrative information

This category includes any changes to the information pertaining to a TSP and/or its service(s) which is/are referenced within the TSL. Such changes could include, *inter alia*, change of TSP addresses, location of specific information referenced by a URI. When any of these change the TSL should be re-issued without any change to the status information pertaining to services operated by the TSP concerned.

When any administrative change occurs the TSL should be re-issued with the previous "Service information" (see clause 6.4) becoming the most recent "History information" (see clause 6.5) and a new "Service information" entry being updated to reflect the new administrative information (without any change to the status itself).

A change to the "Service digital identity" (see clause 6.4.3) should be considered as a change to the service status - see clause C.6.3.

G.3 Trust-service identification

Whenever a scheme operator adds trust service to a TSL, it is important to users of the TSL to be able to unambiguously identify that service's status definition. While name and address may be highly relevant and therefore very important, the digital identity-field is the only option that can provide secure identification of the trust service and tokens which it supplies. The service digital identity-field does not, however, prescribe a specific format for this identifier, since the TSL is intended to be applicable to services based on technologies other than PKI.

For PKI-applications, applications also have choices as to how to present the digital identifier. For creating or parsing TSLs, applications should support three formats for the service digital identity:

1) one of the two methods defined in clause 4.2.1.2 of RFC 3280 [17], on how to calculate subject key identifiers for CA certificates;

2) X.509-certificates;

3) Public key.

G.4 Change of trust-service status

These changes are those directly affecting the inclusion, exclusion or reported status of any trust service within the TSL (and possibly also information concerning their provider) and whether the information is current or historical (e.g. the introduction of a new TSP and service; the revocation of a service).

When any such change occurs the TSL should be re-issued with the previous current status becoming the most recent historical status and current status being amended to reflect the situation.

Where a service changes its "Service digital identity" (see clause 6.4.3), e.g. as a result of a take-over or a re-branding or a renewal of associated digital data for security reasons, the situation should be handled effectively as if the service using the old identity had ceased to operate and the service using the new identity had come into being.

The service which is effectively stopping should have its "Service current status" (see clause 6.4.4) revised to meaning 2 (ceased operations) and the previous status information placed into the "History information" (see clause 6.5) of the TSL. This should then be retained for the published retention period (since there may be requirements to check on services rendered during it period of activity - no ceased service's "Historical information" should be discarded.

The service under the new digital identity should be given its own new entry, which at this initial stage would have no "History information" which required recording.

G.5 Amendment response times

Changes to any TSL information should be provided in a timely fashion, which as a minimum should be the following (the response times taking account of the format of the information's presentation):

a) Within four working hours of a decision to implement a change in status.

b) Where each TSL revision is disseminated electronically to those parties who are obliged by the scheme operator to maintain copy of the TSL for their own clients, a four working hour response should be met. Such parties would typically be TSPs whose services are listed in the TSL, and should themselves undertake to post the revised TSL within the same response criteria.

G.6 On-going verification of authenticity

The frequency at which information within a TSL will change is likely to be low. This could give a determined hacker sufficient time to replicate and replace all instances of a TSL, *IF* they were able to replace all examples of the TSL itself and a surrogate PKC for the TSL operator. This should be protected against by the scheme operator itself making frequent verification of its own TSL and all authorized and recognized replications of it. In addition, the regular re-issuing of the TSL, even when there is no change to any statuses within it, will also ensure that, at the least, the signature value changes periodically. This clause has already discussed some security measures which would reduce significantly the likelihood of this being achievable.

G.7 Upon a scheme's cessation of operations

Owing to the dependence which users may place upon the TSL, schemes which operate a TSL should have in place appropriate mechanisms for any cessation of their operations, be it temporary or permanent. The normative parts of the present document provide for a "Next update" date and time. This field makes explicit provision for a scheme to indicate that it is no longer functioning, by setting this field to null.

Notwithstanding that technical provision which allows a final TSL to be published "in perpetuity", scheme operators need also to consider additional actions to ensure a controlled cessation of their operations. As a minimum, the scheme should revoke the keys used for signing and verification of its TSL and make a public announcement of its cessation of operations, indicating (if known) whether this is temporary or permanent.

If time permits and circumstances warrant, a new TSL must be issued (ref. Next update) which relegates all status records to the history components as of a specific date after which the scheme no longer accepted responsibility for status determination and produces an archive for long-term reference. In addition to the specific provisions of the "Next update" field discussed above, it is required by the normative part ofn the present document that in such a circumstance the field "Service current status" is set to indicate "Expired".

Whilst the issues of the long-term validity of this archived TSL may be something for consideration it is beyond the scope of the present document to deal with them in depth. Suffice to say that, where there is a decision or obligation to hold available the final TSL status for an extended period, appropriate measures (already widely known and discussed in this field) should be taken to protect signatures against the decay of the strength of crypto algorithms.

G.8 User reference to TSL

When and how often a user/relying party should reference to a TSL for status information is not an issue within the scope of the present document. Such a decision lies with the user and should be a determination made according to a variety of factors reflecting their own circumstances, *inter alia*, the degree of reliance they place in a TSL status indication, how often they deal with the other party, the nature of the business relationship and the value of the business or the transaction in question. These are factors only they can determine after conducting their own risk analysis. They may have such infrequent recourse to a TSL that they will always check for any TSL records of status.

Scheme operator's could assist in this by offering additional services to notify when a new TSL is issued, or to guarantee frequent re-issue of a TSL at a frequency which may mean numerous re-issue without change of any services' status. However, the mechanisms proposed for having multiple copies of TSLs existing contemporaneously are designed to cater for the low rate of information change already discussed, and these may not be suitable for frequent TSL re-issue.

G.9 Reliance upon hard-copy TSL information

Whilst it is a requirement that scheme operators make available information which is "human-readable in printable, hard-copy form" there is no requirement, nor expectation, that hard copy should be provided in a manner which can be authenticated by any printable means. Users should expect that authenticated information presented on-screen by an application accessing a TSL will faithfully reproduce that information when it is printed and should take the trouble to cross-check the information with that on-screen where they have any doubts.

Scheme operators might choose to make paper copy available by surface post if that seems desirable.

G.10 TSL size

The present document provides a number of fields in which the scheme operator may choose to provide actual natural language text in preference to a URI or other reference to a source of information. Clearly the inclusion of large quantities of text will have a direct influence of down-load and parsing times, this especially so if e.g. it relates to the descriptions of services, and the scheme has a large number of trust services listed. It is therefore recommended that implementers take advantage of the opportunity to use URIs and limit embedded text as much as is reasonably, accounting for the overall size of the TSL and the available bandwidth and storage capacities of the typical user of their TSL. Referencing other documents also allows advantage to be taken of more sophisticated presentation options which formats such as PDF and other proprietary formats enable.

Annex H (informative): Locating and Authenticating a TSL

H.1 Introduction

This annex offers guidance on how to locate and authenticate TSLs. It does not try to cover all possible scenarios, but focuses on those that are likely to occur. It is based on the following assumptions:

- A relying party intends to authenticate a trust service token (TrST, e.g. a certificate) that has been received from some counter-party (see note).

NOTE: Whilst the relying party may have the desire to authenticate the TrST, the TSL cannot generally be relied upon to provide more than a secondary source of trust. In some circumstances it may be possible to derive from the TrST, information which provides a digital identity for its issuer, and that issuer may then be located within a TSL, there are many assumptions about trust which must be satisfied before a true authentication can be claimed by this process. One should therefore expect that, in general, further steps need be taken to authenticate the TrST.

- The relying party has at least reasons to assume there exists a scheme which the TrST-issuing trust service is part of.
- The relying party has at least reasons to assume the scheme is using a TSL for publishing the status of the services overseen by that scheme.

No further assumptions are made. It may be straightforward to retrieve the TSL or the relying party has to do a thorough search on the internet. Trusting the TSL-issuer is a question of policy and not dealt with at all.

Although this annex is written very much in terms of the relying party searching for and within a TSL which lists general trust services, the principles described may apply equally to the location and authentication of TSLs which list other assessment schemes (i.e. "Schemes" TSLs).

H.2 Locating a TSL

Locating a TSL can either be easy, if the trust service token provides a direct link or any other hint on where the TSL can be retrieved from. If no such information is available. The relying party may use certain strategies to find a suitable TSL. Both models are discussed in the clauses that follow.

H.2.1 TSL location models

We can consider three models by which TSL location information can be provided. They are: Bound, Linked, and De-coupled. Each is explained and their comparative merits considered below.

H.2.1.1 Bound information

In this model, information about a TSL (or possibly more than one) is intimately bound into the TrST. In other words, the TSP advertises the fact that its service fulfils the criteria of the indicated scheme. The user initiating the communication (i.e. the sender) need not be aware of the inclusion of this information.

Such a solution is easy in terms of the need to locate a TSL - the work is done - but it is "dirty" in that it renders the token a victim of the continued fulfilment of the scheme's criteria, and indeed the stability of the scheme itself. In the event that the status of the trust service changes, or the scheme's PKC itself is revoked, or the scheme substantially changes its criteria, or even ceases to exist in its recognized state, the TrST would most probably need to be revoked. This has the implication that a TSP issuing large volumes of tokens would have to revoke and re-issue them in the case of any of these failures originating largely outside its control (of course it may well be that this change in its status is the result of some action (or inaction) on the part of the TSP itself).

In the case of "black list" principle TSLs, it is manifestly unlikely that a TSP will bind in information of a negative nature, and so here the Bound model most probably does not apply. By the same token, even schemes applying positive criteria may find TSPs unwilling to bind in a pointer to information which may put them in a bad light if, for example, they have suffered a degradation in their approval status.

The bound model therefore suffers from its sensitivity to changes from a number of other sources and from circumstances where the TSP may feel jeopardized by inclusion of a reference to its present status. Nevertheless, if used this model obviates the need to search for a TSL (although there may be other TSLs not referenced which might have useful information about the trust service).

The TSL Distribution Point (see clause 6.3) is one of the prime mechanisms to locate a TSL relevant for validating a TrST. This mechanism may be used in all three models.

H.2.1.2 Linked information

In this model, information about any relevant TSL(s) is included within the transaction but not in a way which binds it intimately to the service token. The TSL location could be included by an application, possibly configured by either the user or their service provider; the user may not need to know about it, but transparency may not always be so clear as with the Bound model. The Linked model has the obvious advantage that status information is provided separately from the TrST and hence could change without having any impact on the TrST (although according to the nature of the scheme, this may not always be so).

Most of the arguments about the willingness of TSPs to include this information apply as they do to the Bound model. However, it is clearly less sensitive to status changes and also makes it unnecessary to search for TSL information, with the same caveat that there may be other TSLs not referenced which might have useful information about the trust service.

H.2.1.3 De-coupled information

In the De-coupled model there is no TSL location information provided with the transaction - it is up to the relying party to find it herself. This has the distinct advantage of there being no dependency on the TSP to provide the information, no need for the sender to have any knowledge of this information either.

This model carries a potential penalty: the relying party's system has to search for the TSL, and the search may have no initial clues as to where to look.

H.2.2 Searching for a TSL

It becomes necessary to search for a TSL particularly in the case of the De-coupled model, but it may also be necessary where the information provided through the Bound and Linked cases is inadequate for some reason. Note that a search may also be appropriate simply when an interested party seeks information about a particular TSP and/or its services but does not know where to find an associated TSL.

Searching can be broken down into four potential stages which can be regarded as offering decremental ease of searching. These are described below, starting with the simplest.

H.2.2.1 Same-scheme searching

In this case the relying party is able to use the TSL belonging to any scheme(s) within which fall any trust services with whom she herself has a relationship (and presumably, therefore, in which he has some assurance) - we will use the term "relying-party's scheme/TSL" as a convenience, although strictly speaking there is no direct relationship between the relying party as a subscriber to a service and any scheme under which that service operates. Such an approach would work where the counter-party's trust service is overseen by the same, or one of the, relying party's schemes. Each of the TSLs associated with those schemes could be searched for the presence of status information relating to the counter-party's trust service.

H.2.2.2 Known scheme searching

In this case there are three possible options, each dependent upon the relying party being a subscriber to at least one trust service which is within a TSL-issuing scheme, i.e. that there is a "relying-party scheme" as explained above. These options may exist in any combination.

In the first case, if the relying-party's scheme operates under, or within a federation or community of schemes all supervised by, a Root Key Authority (RKA) then it may be possible to derive from that RKA the location of other schemes which provide TSLs and which could be assumed to have the same degree of assurance as the relying-party's scheme.

In the second case, the relying-party's TSL could contain within it a pointer or pointers to other TSLs (see clause 6.2.12) which the relying-party's scheme operator feels worthy of some degree of recognition, or the scheme operator may publish a "Schemes" TSL to which the relying party could refer. (see TSL type). How one scheme operator determines that another TSL is sufficiently reliable to merit inclusion in their own is not defined by the present document. The scheme operator would be expected to make publicly accessible their policy for doing so, whether by using "Pointers" to other TSLs' or by publishing a "Schemes" TSL.

In the third case, the relying party may have built up their own list of TSLs or have access to an alternative "Schemes" TSL which they regard as reliable and could search any of those.

Thus by any combination of the above options, the relying party could have identified TSLs within which they could search for the presence of status information relating to the counter-party's TSP.

If none of the options in this and the preceding part are successful, then a "blind" search may be conducted, as described below.

H.2.2.3 "Blind" (unknown) scheme searching

If a relying party has absolutely no information about a scheme issuing TSLs relevant for authenticating a TrST, maybe even no information that such a scheme or a TSL exist, the fallback-strategy described in this clause may be successful.

The concept follows the model human users would apply in similar cases: they would use any internet-search engine. TSLs compliant with the present document will use the TSL tag value specified for that field. Thus, finding that tag value in the appropriate field of a data structure should identify it as a TSL. Further qualification and confidence can be drawn by parsing and matching other fields, such as the issuer distinguished name. If the issuers of the TSL follow the recommendations given in the present document, we expect the results of any web search to provide a direct link to a TSL in most cases. This expectation may be thwarted though by sort-of denial of service attacks, e.g. by publishing fake pages that would also show up as hits, but indeed lead to junk information only. It is considered unlikely that such attacks will be interesting enough to execute.

To be able to find a TSL using a search engine, the following assumptions and requirements are relevant:

- A TSL is unlikely to be found directly, so long as search engines do not index unspecific XML or DER-encoded data - at the time of publication of the present document only HTML, PDF and similar formats are indexed. To enable search-engines to find a TSL, an HTML-page is needed that contains a) a searchable string and b) a link to the TSL. By specifying a simple structure for such a page, and simple criteria to make that page "findable", applications will have a straightforward way to locate the TSL.

When a TSL is located by any of these means, its further parsing must be performed taking into consideration which type of TSL it is (TSL type).

H.2.2.3.1 Structure of the HTML-Page.

A scheme issuing a TSL is RECOMMENDED to publish a web-page defined by using either:

a) HTML 4.01 [31] or XHTML 1.0 [29] with **strict** DTD; or

b) XHTML 1.1 [30].

Later versions of XHTML MAY be used as and when they become available and widely accepted. The web page should be compliant with the following structure.

HTML version information.

It is RECOMMENDED to use the following declarations:

for HTML 4.01:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
```

for XHTML 1.0:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

for XHTML 1.1:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
    "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

for future versions of XHTML the declaration should be taken from their specifications.

A document head consisting of:

- The HEAD element using the profile-URI http://uri.etsi.org/02231/TDPContainer which clearly identifies that HTML-document as being a TSL-container.
- A TITLE element with the content "*Trust-service Status List Distribution Points Container*".
- A META element with the name "*contains*" and the content "*XML*" resp. "*DER*" or "*XML,DER*" if the page contains the XML resp. the DER version of the TSL, or both.
- Other META element, such as the element with the name *keywords*, are also possible.

```
<head profile="http://uri.etsi.org/02231/TDPContainer">
<title>Trust-service Status List Distribution Points Container</title>
<meta name="contains" content="XML,DER">
<meta name="keywords" content="TSL,Trust Status List,TDP">
</head>
```

The body-section contains a paragraph with the string suitable for searching this page, followed by several paragraphs, each of which contains exactly one anchor (A) element. The href attribute contains a URI pointing to a TSL. The content of the element must start with the string TSLLink and specify the type of TSL pointed to by adding XML or DER to the string. This is followed by a colon and the name of the scheme to which the TSL relates. This name should be exactly the same as the field Scheme name. If this field contains names in multiple languages, one, some or all of those names can be selected.

```
<body>
<p>This page contains links to objects of type http://uri.etsi.org/02231/TSLtag; the CMS
EncapsulatedContentInfo is identified by the 0.4.0.2231.1.0 / itu-t(0) identified-organization(4)
etsi(0) tsl-specification (2231) identifiers (1) tsl-info(0); the XML object is identified by
(http://uri.etsi.org/02231/v2#, TrustServiceStatusList)</p>
<p>
<a href="URI">TSLLink+[XML|DER]:SchemeName</a>
</p>
</body>
</html>
```

H.2.2.3.2 Example

The following example provides links to two formats of a TSL from the scheme "SomeScheme":

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head profile="http://uri.etsi.org/02231/TDPContainer">
<meta name="contains" content="XML,DER">
<meta name="keywords" content="TSL,Trust Status List,TDP">
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Trust-service Status List Distribution Points </title>
</head>

<body>
<p>This page contains links to objects of type http://uri.etsi.org/02231/TSLtag; the CMS
EncapsulatedContentInfo is identified by the 0.4.0.2231.1.0 / itu-t(0) identified-organization(4)
etsi(0) tsl-specification (2231) identifiers (1) tsl-info(0); the XML object is identified by
(http://uri.etsi.org/02231/v2#, TrustServiceStatusList)</p>
<p>
<a href="http://somescheme.org/tsl/xml/current">TSLLink+XML:SomeScheme</a>
</p>
<p>
<a href="http://somescheme.org/tsl/xml/current">TSLLink+DER:SomeScheme</a>
</p>
</body>
</html>
```

H.3 Authenticating a TSL

It is assumed that each scheme provides its users with the means to authenticate the TSLs it publishes, which may be performed by a TSL:

1) Ensure that the validity period of the TSL has not expired (see clause 5.3.15).

Starting with the scheme operator digital identity reference found within the TSL, retrieve the public key to be used to verify the signaturenumber of different mechanisms. This, therefore, is implementation specific, and it is recommended that scheme operators specify in their policy how to authenticate their TSLs, or provide users with the means to authenticate them. For example, a scheme could:

1) provide a trusted channel (e.g. TLS) to download the TSL from a secured site;

2) publish in a reliable source (e.g. an official bulletin) the digest of the scheme's public key corresponding to the private key used to sign the TSL.

For TSLs located after a "blind search" the means applicable to the authentication of such TSLs may not be immediately apparent to the relying party, and may require human intervention to make it possible.

The continued validity of the TSL should also be verified, by ensuring that the validity period of the TSL has not expired (see clause 5.3.15).

If either of these checks fails, the TSL authentication should be considered to have failed.

NOTE: The decision to trust an authenticated TSL is covered in clause H.4.

H.4 Trusting a TSL

A TSL is a signed electronic document. To verify the signature, relying parties need to be able to access the applicable public key. Since the scheme issuing the TSL is effectively positioned "above" the TSPs approved by that scheme, the authenticity of the public key cannot be certified by any TSP inside or outside the scheme. Providing the scheme's public key is therefore a problem very similar to providing the public key of a CA service and any details are out of scope for the present document. Nevertheless, self-signed keys established by well-known entities may prove to be a suitable solution. It is imperative that the key used for signing the TSL has a public-key certificate published (refer also to clause 6.2).

Widespread replication of a TSL may also be constructive in reducing traffic volumes accessing a single source, where the TSL is large.

After successful authentication of the TSL, the relying party needs to decide if it can trust the TSL. The process to be followed by any user that wants to use a TSL is very similar to the steps that need to be taken when deciding about trust in a certification authority. If public key certificates are used in this process, the relying parties' software should be able to distinguish between certificates trusted for issuing certificates and certificates trusted for issuing TSLs.

Having identified, located and authenticated a TSL, the user could then carry out any further steps to establish trust in the scheme/TSL as required by their own policy. Consequently the user decides whether or not to trust the scheme and the TSLs it operates, and the extent of that trust. Only if these further checks are positive is the information within the TSL relied upon.

The user can then take steps to ensure that on future searches this TSL is automatically accepted as being reliable. A typical procedure might therefore look like the following:

1) User imports the TSL's public key certificate or public key into the software;

2) User sets the status of the imported certificate or public key to something like "*trusted for issuing TSLs"*;

3) User subsequently uses the certificate or public key to verify TSLs maintained by the specified scheme.

It is assumed that the user is able to establish for themselves sufficient trust in the certificate or public key in question by verifying themselves a publicized hash of the certificate or the public key itself, available from some reputable source, e.g. published in an official journal.

The procedure described above can be performed by each user, but will in many cases be carried out on the level of an organization according to their own policy. In this case, the software environment of each user's machine would typically be pre-configured by the system administration or by the security officer. In time it is likely and certainly possible that such certificates or public keys could also be pre-installed in browsers, so enabling personal users to gain advantage from this approach.

In the case of compromise of the scheme's private key, the user must be informed in the same manner as in the case of a key compromise of a TSP's self-certified key. Such key compromise will get broad attention, since there will only be a limited number of schemes operational, they will be widely known, and furthermore their certificates (and therefore notification of their certificates' revocation) will be widely available, ensuring that such events will not remain unnoticed.

A scheme operator may also provide mechanisms compatible with the standard way of handling revocation information: add a CRL distribution point extension into the self-signed certificate and provide a CRL at that point. A compliant client implementation could then also automatically check that CRL to detect any revocation.

H.5 Replicating TSLs

TSLs will be relatively few in number, with only moderate numbers of service statuses described within them and furthermore, since it is unlikely that services will come and go with great rapidity (in terms of internet-speed), they will have a low frequency of information change. For this reason, low-complexity approaches to the publication of TSLs and to control over their authenticity are adopted. A scheme either can build upon the safety in numbers concept (i.e. multiple copies of each TSL) rather than developing more stringent management processes (e.g. specific access controls rather than general publication) or can alternatively adopt the standard central repository approach that is well known and understood from normal certification authority services.

The *safety in numbers*-concept builds on the premise that it is sufficiently difficult to insert multiple forged copies of a TSL into multiple repositories of a number of different organizations. Applications which want to validate a certain TSL therefore can retrieve copies from such repositories and compare them. Whether they only accept a TSL when all copies are equal or takes a majority vote is a policy question and out of scope of the present document.

H.6 Security issues

The security of this approach relies upon there being a reasonable number of TSPs and services, on the web sites of which shall be published the TSL and the related scheme's PKC, to ensure that complete replacement of these sources is a complex and difficult task. However, some specific considerations need to be made.

Where the number of services covered by any one scheme is small the low number of replications increases the vulnerability of the system. This should be overcome by encouraging the publication of the TSL and related PKC on other sites, such as those of government and industry bodies, and co-operating schemes.

Additionally, the public key corresponding to the scheme operator's signing key should be bound into a certificate by each participating TSP, and these certificates published as widely as is the list and the scheme operator's self-signed certificate. Thus, the level of complexity required of any agent intending to corrupt the TSL is increased quite significantly.

Although the idea of a harmonized TSL is to bring all scheme representations up to a consistent level of robustness, early implementations which exercise the "opt-out" implementation of a TSL may find themselves unable to publish their TSL a sufficient number of times. Taking for example a scheme operating only on a "black list" principle, it could be naïve to expect to find willing those TSPs whose services have been indicated as being in default according to the scheme's criteria - there is absolutely no incentive for them to display their own failure! A solution to this could be for such schemes to actually include within their list all TSPs falling within the scope of the scheme and making a distinct separation between those schemes who continue to operate in conformance with the "failure" criteria as well as those who fall into the "black list" zone. This could readily be accomplished by using the appropriate "status" indicators in the standard.

Additionally, some schemes may find comfort in existing within an hierarchical trust model, the wider implications of which could compensate for a small number of published copies of their TSL.

This trust decision process may be a manual one where a person assesses TSP-related information, or an automated one. It is beyond the scope of the present document to consider the complexities of how subjective manual decisions based upon TSL-derived information can be reached, whether published as a web page or printed on paper. This clause therefore focuses on the automated case only, where a signed TSL is handled by some piece of software which needs to make an automated decision.

H.7 Implications for authentication of Trust Service Tokens

Although a relying party searching a TSL for a status indication relating to the issue of some TrST it possesses, may have the desire to authenticate the TrST, the TSL generally provides only a secondary source of trust. In some circumstances it may be possible to derive from the TrST information which provides a digital identity for its issuer (e.g. a Time Stamping Token that includes the TSA's PKI certificate), and that issuer may then be located within a TSL, there are many assumptions about trust which must be satisfied before a true authentication can be claimed by this process. One should therefore expect that, in general, further steps need be taken to authenticate the TrST.

For sufficient confidence to exist such that a TrST can be considered to be a source of primary trust (i.e. to provide sufficient confidence to the relying party that the TrST is valid and issued according to certain criteria such that the relying party can depend upon the token and the transaction for which it stands) a number of factors must be considered, amongst which might be:

- When the TSL is of type "Generic", the strict relationship between the scheme issuing the TSL and the included service must be understood, in terms of the processes and criteria which are vouched-for.
- When the TSL is of type "Schemes", the relationship between the scheme issuing the TSL and those other schemes to which it refers must be understood, in terms of the processes and criteria which are vouched-for by those schemes, and in turn by those schemes and the services they list.
- Legal implications, such as the standing of the schemes concerned, and potentially whether authentication by reference to the TSL listing would be sufficient for legal evidentiary purposes (e.g. as opposed to parsing a certificate chain to a root certificate, as may be required in some jurisdictions).

With a sufficiently rigorous definition and understanding of a scheme's operation, its management processes and the criteria which it applied to determine the status of services which it listed (or other schemes which it listed, as appropriate), perhaps coupled with appropriate understanding of the liability implications, a scheme could, within a well-define community, be a source of primary trust, and therefore a source of authentication for trust services.

Annex I (informative): General TSL usage

I.1 Introduction

This annex serves to describe some general scenarios in which TSLs can be used, including how they can be located. It is not the intention to exhaustively detail all possible cases of use, nor does it assume any specific types of trust service, although it does discuss some key distinctions which are recognized by the type of TSL being used. The annex assumes familiarity with annex H, which describes how TSLs can be located and authenticated.

The present document describes two types of TSL, "Generic" and "of Schemes". This annex first considers TSLs of the "Generic" type, and then considers the alternative "of Schemes' type.

I.2 Generic TSL usage

The TSL was originally envisaged as a means to provide status information on electronic trust services falling within the scope of a scheme's oversight, whether by regulatory power or by voluntary acceptance. Such services evolved principally from those required to support Public-Key Infrastructures (PKI), although other electronic services not directly related to PKI but still providing trust through their functions were also anticipated, and the TSL structure as defined allows for these and is extensible to account for new electronic trust services as they arise.

Some examples of how a TSL can be used are given below. They do not go into great depth, but they do show the range of possible application of a TSL and the flexible nature iof the present document.

I.2.1 EC Supervisory System "D"

In this exemplar case "SupervisorStateD" (an EU Member State which is subject to Directive 1999/93/EC [1]) operates a "system of supervision" (see clause 3.3 of Directive 1999/93/EC [1]). The supervisory body approves QC-issuers only and signs the Cas' signing keys with its key. A trust path extends back to the supervisor's certificate, and relying parties are expected to parse that chain before determining whether they should rely upon the certificate in the context presented.

In this case, a TSL probably adds little direct benefit in the parsing of the certificate chain. The TSL does however provide useful publishable information for subscribers wishing to subscribe to a qualified certificate-issuing service in that Member State - the TSL will show which providers are available, where they are located, how they may be contacted, how good is their track record, and how long they been operating (assuming that the scheme has existed since the first such services began operating and that all such services were obliged to be within the scheme from that date). One can expect that the national body responsible for the scheme and publishing the TSL is well-known - it is a part of the government!

NOTE: In this case the service provider's legal right to operate is probably based on whether the provider's signing certificate remains valid (therefore legal) or has been revoked by the root authority (therefore illegal).

I.2.2 EC Supervisory System "G"

In another scenario, "SupervisorStateG" (another EU Member State which is subject to Directive 1999/93/EC [1]) operates a "system of supervision", the rules of which take a passive view of the fulfilment of the requirements upon supervisory systems. This system lists those providers of services related to QCs, and possibly any other trust services, by requiring them to notify the supervisory body of their services. The supervisory body monitors the market place and revokes inclusion on the list for those providers which are found not be fulfilling the requirements of Directive 1999/93/EC [1].

If this scheme does not publish a TSL, that might be inconvenient for relying parties. If the scheme publishes a TSL, a parsing application can LOCATE the appropriate, according to the techniques described in annex H. Essential criteria for locating any TSL entry which relates to this service would be:

- we rely upon the TSL bearing the requisite TSL tag;
- we rely upon the TSL relating to an EU supervisory system established in Member State "G";
- we know the service digital identity (from the TrST, i.e. the QC);
- we know the service type to be a QC issuer (so search for that service in a TSL which has those qualifying attributes).

Check the status of the Trust Service at the present time and derive a secondary trust factor according to whether the status is good or not. Legal right to operate is less obvious, although trading laws might be applied if not compliant with regulations.

I.2.3 Trust service status as legal evidence

In this case we imagine that "Consumer-alpha" denies that they ever entered a contract with "OrganizationX". "OrganizationX" holds "Alpha"s e-signature on a contract, and believes it to be supported by a QC and therefore having the legal status and value which that provides. However, "OrganizationXs" company policy is not to verify the certificates on contracts below one thousand euros. Now "OrganizationX" needs to prove its case - its legal representatives refer to the contract, find the date it was executed, then LOCATE a TSL which has oversight of the issuer (the issues around locating a TSL have now been amply discussed), and look for a record of the status of the certificate issuer on the date on which the contract was effected. There are a number of possible outcomes:

- No record in any TSL - no supporting evidence available; TSL with no history, or no history for the date in question - as previous outcome;
- History present for the required date - status good (i.e. was operating as a valid issuer of QCs at the time of issuing the certificate on which the contract signature is based - supporting evidence available:
  - status bad, may not be a QC; and
  - no obviously positive evidence to support "OrganizationXs" case.

I.2.4 Checking for anomalous status before accepting a credential

A voluntary approval scheme, "Trustscheme", is registered in one country but is an industry scheme set up for the good of many players within a larger community extending across national (and therefore legislative) boundaries. Approval by "Trustscheme" does not confer or deny any legal rights. It shows that the service is (or is not) being operated according to defined practices and criteria which are freely publicized, and that the services claiming compliance with those criteria are regularly audited. Finding such status information within the scheme's TSL will provide a secondary level of trust to a relying party. A parser could flag a bad status for checking prior to a transaction being enabled (similar to the way in which a browser may warn about a certificate it does not recognize when accessing web resources - a little window pops up and says "certificate not recognized - what do you want to do?" (not being bothered, most users will click "Accept" - but its their or their employer's choice!)). Such flags could be based upon final value or other criteria an automated process could apply - e.g. only if from a particular country, a particular organization, etc.

I.2.5 Cross-certification status confirmation

Should a national government wish to establish a National PKI Bridge CA (NBCA), which enables a community of Cas (in the all-inclusive term of them being either separate service components or all-in registrars, issuers, status publishers, etc.) to inter-operate against equivalent policy requirements. NBCA publishes a TSL listing all those services which have been certified according to the NBCA Policy Authority. Whenever any member of the NBCA community receives some TrST it first looks in the known TSL_{NBCA} which tells it how to react. Assuming the issuer of the TrST is shown having a good status at the time of issuing the TrST and at the current time, then the TrST is given due recognition, i.e. treated according to the agreed cross-certification rules. If the issuer/service provider cannot be found, some other process must be invoked (alert for human action, apply some other automated process, which may involve searching elsewhere), but cross-certification cannot be assumed. The textually-published TSL serves to assist subscribers and other users as to which organizations are cross-certified.

I.3 TSLs used to list other schemes

In the first version of the present document the field Pointers to other TSLs was provided. This allowed a scheme operator to provide pointers to other TSLs about which it knew, and according to whatever selection process it chose to apply (i.e. the specification imposed no specific selection criteria, even implicitly).

A specific development in the potential application of a TSL has been to make reference to other Scheme Operators and their TSLs, should those Scheme Operators issue them. From release *2.1.1* of the present document, there has been the capability to include another trust assessment scheme as a recognized "electronic trust service". The use of the TSL structure in such a case does not vary although the scheme-operator is at liberty to establish and publish their own rules for how their TSL is managed (i.e. the rule-set which applies to it).

This is based upon the principal of including another scheme operator's services as a type of trust service. This is logically consistent with the approach taken by the TSL specification: define the service, define the rules for inclusion of any specific service, apply those rules and list qualifying services accordingly. Those rules may be as rigid or as flexible as the scheme operator chooses, and need not be the same as those used by any other assessment scheme which is included.

By this means one scheme operator can be included within another's TSL. It is worth noting that the *referenced* scheme need not necessarily provide its own TSL - that would be a decision factor left to the owner of the scheme which is referenced.

I.3.1 Hierarchical relationships

In this clause, the term "hierarchy" is not intended to imply that any control exists between a scheme and other assessment schemes which it may include with its TSL. It may be that controls ***do*** exist between them, but here there is no presumption or reliance of that being the case.

Where a TSL "of Schemes" refers to other assessment schemes (the referenced schemes) the operators of those referenced schemes should be regarded as TSPs. The actual schemes which they operate should be regarded as trust services. The same rules which apply to the treatment of conventional trust services and their providers apply here. This approaches enables a common TSL format and accommodates an organization operating more than one scheme and publishing a TSL for each.

The following table indicates how key fields within a TSL "of Schemes" should be derive their content from fields within a referenced TSL (which could be of any type recognized by the present document or by the scheme operator which publishes the TSL "of Schemes").

"TSL of Schemes" field	Source field in the referenced TSL
TSP name	Scheme operator name
TSP address	Scheme operator address
Service name	Scheme name
Scheme service definition URI	Scheme information URI
Service digital identity	Scheme identification

Further to the above, the Service Supply Points of a "Schemes" TSL field may be used to provide the URI at which any TSLs (i.e. the TrST) issued by the listed schemes can be found (noting that an assessment scheme may issue a TSL by choice, not by any normative requirement of the present document). As indicated in clause H.7, the content of the field "Service digital identity" of a "Schemes" TSL may also be used to authenticate the TSL pointed to by these URIs. Therefore a "Schemes" TSL may be used to locate and/or authenticate TSLs issued by other schemes, if all schemes so-referenced can be relied upon to apply the same rules and field usage (e.g. by adhering to a commonly-agreed TSL profile).

One can consider a number of potential reasons for wishing to establish a TSL "of Schemes" (ToSch)- the following clauses offer a brief number of cases where a TSL can be used in this way. As they progress they illustrate use cases where the degree of certitude as to the meanings and processes in each case is greater.

I.3.2 A collection of TSLs

The previous Annex acknowledged the need sometimes to search for TSLs, which could be a laborious and time-consuming process if it has to be performed frequently (in practice this shouldn't be the case, but circumstances may vary). A beneficent entity might set up a web-crawling application to continuously crawl the internet and locate TSLs. Each time it did so it could perform checks on the TSL (identified because it had a verifiable "TSL tag") to see whether it had previously been located, and if not then the new TSL could be highlighted in order that the beneficent entity could research details of the scheme concerned, which could then be added to the TSL "of Schemes" the entity maintained. Depending on the checks it performed, and possibly filtering and rejection rules it applied, the resultant "Schemes" TSL could range from having a completely unqualified selection of other schemes, to having those schemes categorized or even selected for inclusion against defined criteria.

Such a TSL might be used by third parties who would more quickly locate other TSLs and could then apply their own specific queries to determine the TSL type and whether the service of interest was recorded. Note that in this web-crawling scenario, an un-filtered TSL "of Schemes" may include other TSLs "of Schemes", which users would need to recognize in order to correctly handle them.

I.3.3 Schemes applying common rules

Within a well-defined community, e.g. the EU or EFTA, there are a number of sovereign states working within a common legislative framework. Different states may (and generally do) implement framework legislation in different ways, but within the scope of the framework. A "RegionalBridge" might address this need.

In the Europe Union it could be used as follows. All Member States are required to establish systems for ensuring that issuers of QCs are in compliance with Directive 1999/93/EC [1]. Each country has a supervisory system: one might observe that they vary and some schemes publish a TSL, not all do. There is no obvious (i.e. consistent, normalized) way to locate these schemes, or any TSL they may operate - different ministries are involved, some schemes are outsourced to an industry body and no standardized naming conventions are recognized.

A central body might sponsor a simple scheme to merely list all supervisory schemes of the participating states. This could also be extended to include also voluntary schemes - it would be for the central scheme operator to define within their TSL how they did this. The provision now within the TSL specification for scheme operators to use registered URIs would facilitate the distinction between supervisory systems and voluntary schemes (see Service type identifier, in the context of "Scheme" TSLs). In the absence of a central body to support such a TSL, any other national body may provide such a function which might become widely recognized as a reliable reference source.

To facilitate the development of this kind of TSL "of Schemes" it is RECOMMENDED that the URI registered in accordance with clause D.3 actually points to the scheme concerned.

A similar use case exists for defined industry sectors, e.g. aerospace / defence / automotive / etc.

I.3.4 Schemes trusted by a vendor community

In a commercial use case, one might suppose that a large software company, "Megatuff", wants to add to its browsers a capability to add secondary trust to any certificates used in web sites and related services but has a problem in knowing where such trust may be found. It implements a scheme which publishes a TSL listing only other schemes which provide a degree of secondary trust which. "Megatuff" defines some basic requirements that these schemes must fullfil and then adds to its TSL all those which meet those requirements. Where regional considerations dictate, a hierarchy might be created: TSLglobal, which points to TSLregionA, TSLregionB, etc. Thus a set two-level hierarchy of TSLs "of Schemes" is created, perhaps locally managed against common policy.

I.3.5 Industrial trading consortium

In the final use case, we consider a Trans-Oceanic Consortium (TOC) which wants to establish some common rules for the identity proofing and credential-issuing of participants within a collaborative industry network. National criteria apply and must be fulfilled by industry located in that region. Assuming that participants within the consortium are required to use credentials issued by a service provider who's service has been assessed for compliance with the common rules, the TOC has two possible approaches to help consortium members check the status of their own and their counter-parties' services:

a) establish a "Generic" TSL, which individually lists each suitable service provider. In this case oversight may be difficult, since the TOC would need to effectively operate an assessment process of its own (even if outsourced);

b) establish a "Schemes" TSL, which referred to schemes which might be nationally established or which were industry / sector-based (see previous regional case).

The above use cases cover a broad spectrum of potential application of the TSL in both its types as defined within the present document. Adoption of the present document by assessment schemes will resolve the specifics and provide practical lessons.

Annex J (informative): TSL manual/auto field usage

The following table lists all fields defined for the TSL and indicates whether the field contents should be made available to users when presenting the TSL in a human-readable form (column 2) or whether the field is considered to be essential for effective automatic parsing (column 3), noting that all fields will be accessible through an automated process.

Although this annex is informative implementers are strongly recommended to satisfy the guidance which it provides, in order to provide users with information about TSLs in a consistent manner.

Field name	Human-readable?	Machine-processable?
Identification Tag		
TSL tag		✓
Scheme information		
TSL version identifier		✓
TSL sequence number		✓
TSL type	✓	✓
Scheme operator name	✓	
Scheme operator address	✓	
Scheme name	✓	
Scheme information URI	✓	✓
Status determination approach	✓	✓
Scheme type/community/rules	✓	✓
Scheme territory	✓	✓
TSL policy/legal notice	✓	✓
Historical information period	✓	✓
Pointers to other TSLs	✓	✓
List issue date and time	✓	✓
Next update		✓
Scheme extensions	where recognized and meaningful	where recognized
TSP information		
TSP name	✓	
TSP trade name	✓	
TSP address	✓	
TSP information URI	✓	✓
TSP information extensions	where recognized and meaningful	where recognized
Service information		
Service type identifier	✓	✓
Service name	✓	
Service digital identity	✓	✓
Service current status	✓	✓
Current status starting date and time	✓	✓
Scheme service definition URI	✓	✓
Service supply points	✓	✓
TSP service definition URI	✓	✓
Service information extensions	where recognized and meaningful	where recognized
Historical service information		
Service type identifier	✓	✓
Service name	✓	
Service digital identity	✓	✓
Service previous status	✓	✓
Previous status starting date and time	✓	✓
Service information extensions	where recognized and meaningful	where recognized
TSL signature information		
Scheme identification		✓
Textual certificate details, time and date of signing	✓	✓
Cryptographic data		✓

Annex K (informative): Bibliography

- ISO/IEC 6429: "Information technology -- Control functions for coded character sets".
- ISO/IEC 2022: "Information technology - Character code structure and extension techniques ".

History

Document history		
V1.1.1	October 2003	Publication
V2.1.1	March 2006	Publication