ETSI TS 103 221-1 V1.13.1 (2022-12)

TECHNICAL SPECIFICATION

**Lawful Interception (LI);
Internal Network Interfaces;
Part 1: X1**

Reference

RTS/LI-00233-1

Keywords

interface, lawful interception

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 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

If you find a security vulnerability in the present document, please report it through our
Coordinated Vulnerability Disclosure Program:
https://www.etsi.org/standards/coordinated-vulnerability-disclosure

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or
other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

ETSI

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are 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 (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, 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.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Lawful Interception (LI).

The present document is part 1 of a multi-part deliverable covering the Internal Network Interfaces for Lawful Interception (LI), as identified below:

Part 1:   "X1";

Part 2:   "X2/X3".

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document defines an electronic interface for the exchange of information relating to the establishment and management of Lawful Interception. Typically, this interface would be used between a central LI administration function and the network internal interception points.

Typical reference models for LI define an interface between Law Enforcement Agencies (LEAs) and Communication Service Providers (CSPs), called the handover interface. They also define an internal network interface within the CSP domain between administration and mediation functions for lawful interception and network internal functions, which facilitates the interception of communication. This internal network interface typically consists of three sub-interfaces; administration (called X1), transmission of intercept related information (X2) and transmission of content of communication (X3). The present document specifies the administration interface X1.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

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

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI TS 133 107: "Universal Mobile Telecommunications System (UMTS); LTE; Digital cellular telecommunications system (Phase 2+) (GSM); 3G security; Lawful interception architecture and functions (3GPP TS 33.107)".

[2] IETF RFC 4122: "A Universally Unique Identifier (UUID) URN Namespace".

[3] W3C® Recommendation 28 October 2004: "XML Schema Part 2: Datatypes Second Edition".

[4] ETSI TS 103 280: "Lawful Interception (LI); Dictionary for common parameters".

[5] Recommendation ITU-T E.212: "The international identification plan for public networks and subscriptions".

[6] ETSI TS 123 003: "Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; 5G; Numbering, addressing and identification (3GPP TS 23.003)".

[7] IETF RFC 3261: "SIP: Session Initiation Protocol".

[8] IETF RFC 3966: "The tel URI for Telephone Numbers".

[9] IETF RFC 3508: "H.323 Uniform Resource Locator (URL) Scheme Registration".

[10] IETF RFC 7542: "The Network Access Identifier".

[11] IETF RFC 2865: "Remote Authentication Dial In User Service (RADIUS)".

[12] IETF RFC 2818: "HTTP over TLS".

[13] IETF RFC 7230: "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing".

[14]   IETF RFC 5246: "The Transport Layer Security (TLS) Protocol Version 1.2".

NOTE: Obsoleted by IETF RFC 8446.

[15]   Void.

[16]   IETF RFC 7525: "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)".

[17]   IETF RFC 6125: "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)".

[18]   IETF RFC 4519: "Lightweight Directory Access Protocol (LDAP): Schema for User Applications".

[19]   ETSI TS 103 221-2: "Lawful Interception (LI); Internal Network Interfaces; Part 2: X2/X3".

[20]   IETF RFC 8446: "The Transport Layer Security (TLS) Protocol Version 1.3".

[21]   IETF RFC 7540: "Hypertext Transfer Protocol Version 2 (HTTP/2)".

[22]   ETSI TS 133 127: "Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; 5G; Lawful Interception (LI) architecture and functions (3GPP TS 33.127)".

[23]   IETF RFC 6530: "Overview and Framework for Internationalized Email".

[24]   W3C® Recommendation 21 March 2017: "XML Path Language (XPath) 3.1".

[25]   IETF RFC 6920: "Naming Things with Hashes".

[26]   FIPS PUB 202: "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions".

[27]   IETF RFC 7042: "IANA Considerations and IETF Protocol and Documentation Usage for IEEE 802 Parameters".

2.2   Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1]   OWASP: "Transport Layer Protection Cheat Sheet".

NOTE: Available at https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Protection_Cheat_Sheet.html.

[i.2]   ETSI TR 103 308: "CYBER; Security baseline regarding LI and RD for NFV and related platforms".

[i.3]   ETSI GS NFV-SEC 009: "Network Functions Virtualisation (NFV); NFV Security; Report on use cases and technical approaches for multi-layer host administration".

[i.4]   ETSI GS NFV-SEC 012: "Network Functions Virtualisation (NFV) Release 3; Security; System architecture specification for execution of sensitive NFV components".

[i.5]   OWASP: "XML Security Cheat Sheet".

NOTE: Available at https://cheatsheetseries.owasp.org/cheatsheets/XML_Security_Cheat_Sheet.html.

[i.6]   GSMA RCC.07: "Rich Communication Suite - Advanced Communications Services and Client Specification".

3   Definition of terms, symbols and abbreviations

3.1   Terms

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

destination: point to which xIRI and/or xCC is delivered by the NE

Destination IDentifier (DID): identifier to uniquely identify a Destination internally to the X1 interface

Destination Set: collection of DIDs and their associated preference of use

Destination Set IDentifier (DSID): identifier to uniquely identify a Destination Set internally to the X1 interface

Network Element (NE): element or function performing the interception

   NOTE: Equivalent to the term Network Function (NF).

protocol error: error at the X1 protocol level (rather than any fault with ADMF or NE)

   NOTE: In the present document, the term "error" in general refers to a protocol error, whereas issues with systems not behaving correctly are called "faults".

task: continuous instance of interception at a single NE carried out against a set of target identifiers, identified by an X1 Identifier, starting from an activate command and ending with a deactivate command or terminating fault

terminating fault: fault signalled from NE to ADMF which terminates the specific Task

X1: LI interfaces internal to the CSP for management tasking

X2: LI interfaces internal to the CSP for xIRI delivery

X3: LI interfaces internal to the CSP for xCC delivery

X1 Identifier (XID): identifier to uniquely identify a Task internally to the X1 interface as well as across related X2 and X3 interfaces

   NOTE: The XID is also either associated to only one LIID or can be allowed to be associated to multiple LIIDs.

X1 Transaction ID: identifier used to identify a specific request/response pair

3.2   Symbols

Void.

3.3   Abbreviations

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

   ADMF   ADministration Function
   AVP   Attribute-Value Pair
   CC   Content of Communication
   CIDR   Classless Inter Domain Routing
   CSP   Communication Service Provider
   DID   Destination IDentifier
   DSID   Destination Set Identifier
   EUI   Extended Unique Identifier

FQDN   Full Qualified Domain Name
GTP-C   GPRS Tunnel Protocol (Control plane)
GTP-U   GPRS Tunnel Protocol (User plane)
HI   Handover Interface
HTTP   HyperText Transfer Protocol
HTTPS   HTTP over TLS
IMEI   International Mobile Equipment Identity
IMEISV   International Mobile Equipment Identity Software Version
IMPI   IP Multimedia Private Identity
IMPU   IP Multimedia PUblic identity
IMSI   International Mobile Station Identity
IP   Internet Protocol
IRI   Intercept Related Information
LEA   Law Enforcement Agency
LEMF   Law Enforcement Monitoring Facility
LI   Lawful Interception
LIID   Lawful Interception IDentifier
MAC   Media Access Control
MDF   Mediation and Delivery Function
MSISDN   Mobile Station International Subscriber Directory Number
NAI   Network Access Identifier
NAT   Network Address Translation
NE   Network Element

NOTE: The element or function performing the interception.

NF   Network Function
NFV   Network Functions Virtualisation
OID   Object ID
OWASP   Open Web Application Security Project
POI   Point Of Interception
QoS   Quality of Service
RADIUS   Remote Authentication Dial In User Service
RCS   Rich Communication Suite
RDN   Relative Distinguished Name
SGSN   Serving GPRS Support Node
SIP   Session Initiation Protocol
SIP-URI   Session Initiation Protocol Uniform Resource Identifier
SNMP   Simple Network Management Protocol
SUCI   SUbscription Concealed Identifier
TCP   Transmission Control Protocol
TEL-URI   Telephony Uniform Resource Identifier
TISPAN   Telecommunication and Internet converged Services and Protocols for Advanced Networking
TLS   Transport Layer Security
TPM   Trusted Platform Module
UDP   User Datagram Protocol
UID   Unique IDentifier
URI   Uniform Resource Identifier
UTF   UCS Transformation Formats
UUID   Universally Unique Identifier
xCC   X3 Content of Communications
XID   X1 Identifier
xIRI   X2 Intercept Related Information
XML   eXtended Markup Language
XSD   XML Schema Definition

4   Overview

4.1   Reference model

4.1.1   Overview

The X1 interface is based on communication between two entities; the controlling function (e.g. a CSP ADMinistration Function (ADMF)), and the controlled function (e.g. a Network Element or Network Function (the terms are equivalent) performing interception or mediation and delivery). The X1 reference model is shown in figure 1.
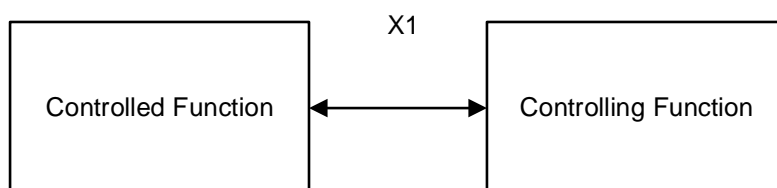
Figure 1: X1 reference model

While the present document uses the terms Network Element (NE), the term is intended to represent any given Network Function (NF) which is intended to be given information regarding interception or mediation and delivery. Similarly, the term "ADMF" is intended to represent any given network function that controls interception or mediation and delivery in other functions.

4.1.2   ADMF deployment model

Figure 2 shows a deployment model for X1 where a CSP ADMF uses X1 to provision a number of NEs to perform interception.
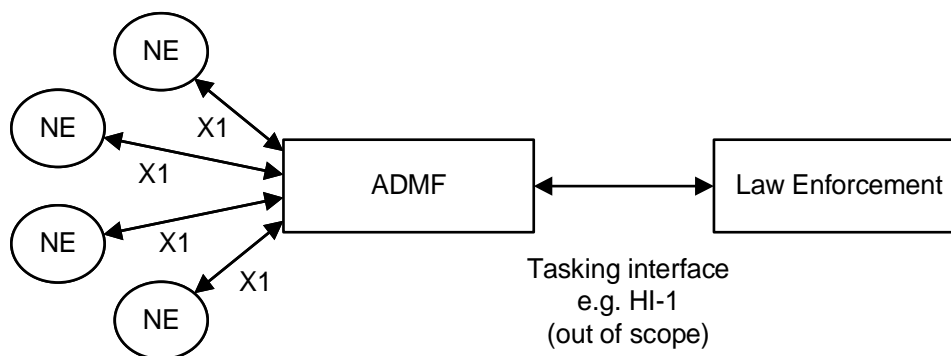
Figure 2: X1 reference model

Only one ADMF shall make changes by X1 to a given NE. This is called the ADMF which is "responsible" for that NE.

Onward delivery of information from the NE is called X2 (for xIRI) and X3 (for xCC). X2 and X3 are defined in ETSI TS 103 221-2 [19].

Some deployments may involve multiple ADMFs for redundancy or other purposes; where multiple ADMFs are required, the NE shall be implemented such that it presents itself as a separate NE to each ADMF.

ADMF and NE shall implement time synchronization where possible; in situations where it is not possible, the ADMF shall maintain knowledge of the timing offset between the ADMF and NE.

NOTE: The present document may be used in direct delivery scenarios, in which the NE delivers directly to the LEMF. Any consequences of using direct delivery are out of scope of the present document.

4.1.3   Triggering deployment model

Figure 3 shows another possible deployment model for X1, where the X1 protocol is used to trigger interception by one in a second network function. In this deployment model, the "Triggering Function" takes on the role of the ADMF in the previous deployment model, while the "Triggered Function" takes on the role of the NE.
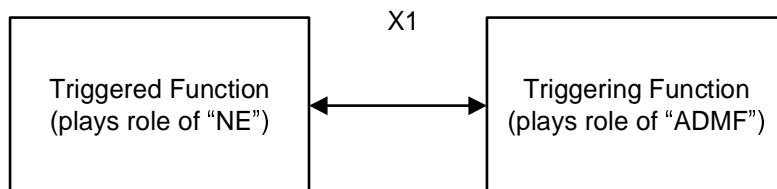
Figure 3: X1 deployment model for Triggering Functions

If this deployment model is used, then in the following clauses references to the ADMF should be interpreted as applying to the Triggering Function, while references to the NE should be interpreted as references to the Triggered Function.

4.1.4   Mediation and delivery function deployment model

Figure 4 shows another possible deployment model for X1, where the X1 protocol is used to manage a CSP mediation and delivery function. In this deployment model, the MDF takes on the role of the NE in the previous deployment model.
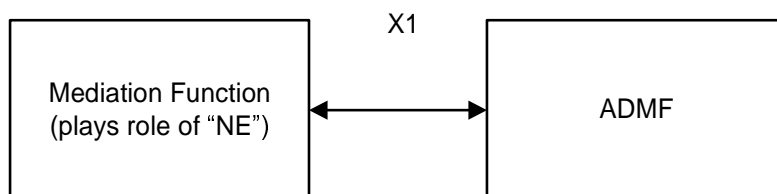
Figure 4: X1 deployment model for Mediation and Delivery Functions

If this deployment model is used, then in the following clauses references to the NE should be interpreted as applying to the MDF.

4.2   Reference model for X1: requesting and responding

X1 transactions consist of a request followed by a response.

Requests may be sent in either direction i.e. with the ADMF or NE initiating the request. The side initiating the request is called the "Requester"; this term is used when it is not specified whether it is the ADMF or NE making the request. The other side is called the "Responder".
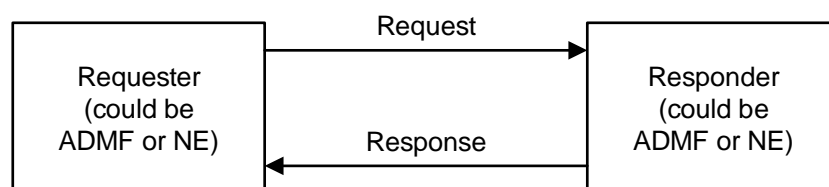
Figure 5: Showing generic terminology

It is likely that in most situations, the ADMF will initiate the message i.e. to distribute information or request status. However, it is possible that the NE will initiate the request in order to deliver fault reports, etc.

ADMF is Requester
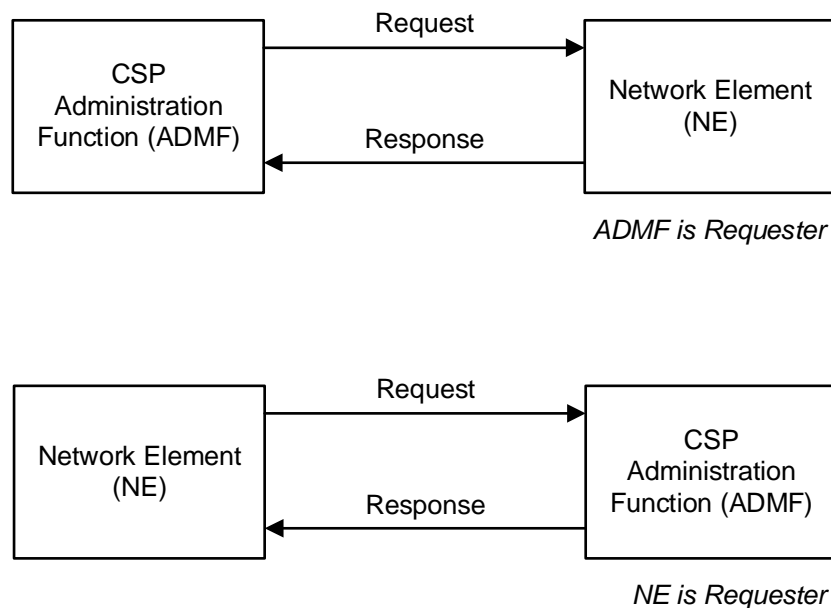
NE is Requester

Figure 6: Showing two situations with either ADMF or NE as the requester

4.3   Overview of security

Security is based on creating public/private keys for the ADMF and each NE for which it is responsible. All transactions over X1 are performed using the security procedures in clause 8, which provide assurance that communication only takes place between an NE and ADMF which have been populated with the relevant key material.

NE implementers are strongly discouraged from exposing additional interfaces for controlling the LI functionality of the NE other than by X1 e.g. via a local administrative interface at the NE. If such additional interfaces exist, any such action performed on the NE shall be captured on the NE audit/logging, and any consequences of such actions shall be able to be seen and controlled by the ADMF that is responsible for the NE i.e. the ADMF shall be able to use the X1 interface to stop or undo any changes made over a local administrative interface. There may be broader consequences that are not covered by the present document if an NE is tasked independently of the X1 interface (e.g. security concerns).

4.4   Relationship to other standards

The present document forms part of a family of internal interface documents covering all of X1, X2 and X3 which are handled as separate standards.

Some models of LI (e.g. 3GPP TS 33.107 [1] and 3GPP TS 33.127 [22]) define interfaces for the purposes described in clause 4.1, (e.g. X1_1, X1_2 and X1_3 defined by 3GPP TS 33.107 [1]; or LI_X1 defined by 3GPP TS 33.127 [22]). The present document is designed to fulfil the requirements for those interfaces.

4.5   Release management

This clause describes the release management requirements. The requirements are:

- The version of the present document is defined as <major>.<minor>.<patch>.

- The major version should be incremented when making a backwards incompatible change.

- The minor version should be incremented when adding backwards compatible functionality.

- The patch version should be incremented when fixing a backwards compatible bug.

Once a major version has been incremented, the previous major version will be supported for 2 years after publication of the new version. Change requests issued to a version that is no longer supported will need to be issued for the latest supported major version.

5   Basic concepts

5.1   The lifecycle of a Task

5.1.1   Start and end of a Task

A Task relates to a single target identifier, and goes from the point an ActivateTask Request is sent by the ADMF to the time a DeactivateTask Request is sent by the ADMF, a "terminating fault" occurs, or (for Tasks with the "ImplicitDeactivationAllowed" flag set) the NE determines that it has completed.

The present document does not define which situations are categorized as "terminating faults". Local recovery procedures should be followed before a Task is ended with a "terminating fault". In general, irrecoverable failures with an interception, or major security issues at an NE should be considered terminating faults, and certain outcomes with keepalives are also terminating faults (where defined in clause 6.6.2).

5.1.2   Identification of a Task

Each Task on X1 is uniquely identified by an X1 Identifier (XID) and it is handled independently of all others. The ADMF shall assign the XID as a version 4 UUID as per IETF RFC 4122 [2]. The ADMF is responsible for correlating the XID to any LI instance identifiers used to communicate with Law Enforcement. When used between the ADMF and the MDF, the entire LI system may support one of several possibilities:

1)   an XID may only map to a single LIID; or

2)   an XID may map to multiple LIIDs.

In the first case, each intercept is separately provisioned for a target ID at a given POI. In either case, the ADMF shall provide the XID to LIID(s) mapping to the MDF.

In addition, the XID is released once the Task has ended.

5.1.3   Destinations

Intercepted traffic is delivered by the NE to a Destination. Each Destination is uniquely identified by a Destination Identifier (DID), and is handled independently from details of the Task. DIDs can optionally be grouped with individual DID preference weightings as part of a Destination Set. Destination Sets specify an action, which defines how DIDs within the Destination Set are used; Destination Sets are uniquely identified by their Generic Object ID, which is referred to as the Destination Set Identifier (DSID) (see annex E).

Each Task is associated with one or more Destinations or Destination Sets. Prior to associating a Task with a given DID or DSID, it is required that a Destination with the DID, or Destination Set with the DSID has already been created (as described in clause 6.3 and annex E) but there is no requirement that a connection has been successfully established for that DID or DSID.

Checks regarding availability and status of downstream delivery of information are outside the scope of the present document.

5.1.4   Generic Objects

The NE may require supplementary information which is not described within the Task or Destination objects. Such information is contained within structures derived from a Generic Object, which may be managed via the messages defined in clause 6.8. Generic Object are defined in clause 6.8.1.2. This mechanism shall only be used for the implementation of objects that are defined and standardized in the present document.

5.2   The lifecycle of an X1 request/response

5.2.1   Identification of X1 request/response

Each request and response shall be identified by an X1TransactionID. The requester (may be ADMF or NE) shall assign an X1TransactionID as a version 4 UUID as per IETF RFC 4122 [2].

5.2.2   Responding to the request

The response shall be sent without undue delay and shall be sent within TIME1 of receiving the request. TIME1 shall be configurable and by default TIME1 shall be five seconds. TIME2, the time a requester waits for a response, shall be configurable, it shall be at least twice TIME1 and by default shall be fifteen seconds.

An error response shall be sent if the request is not compliant syntactically (it does not match the schema) or semantically (it is not compliant or consistent with the existing state of the NE e.g. activating an existing XID).

If the request is compliant, one of the following responses shall be sent:

- "OK - Acknowledged and Completed" response shall be sent if the request is fully understood, compliant and the request has been successfully completed. If the request was a request for information then all the information shall be delivered together as part of the "OK - Acknowledged and Completed" response. The NE and ADMF shall be designed so that information requested (status and Task information) is in a data store which is readily available without undue delay and within TIME1.

- If the action requested cannot be completed within TIME1, an "OK - Acknowledged" response shall be sent. A status report shall be sent by the NE as soon as the action is completed or if it is unsuccessful (see clause 6.5.2.2). This status report shall be sent as a new request/response pair, using the same XID or DID but the status report shall have its own X1TransactionID. The "OK - Acknowledged" response shall only be used for responding to requests which are Activating, Modifying or Deleting either Tasks or Destinations (those in clauses 6.2 and 6.3) and they shall not be used to respond to other request types.

5.2.3   Behaviour if a response is not received

If the requester has not received a response after TIME2 (as defined in clause 5.2.2), or if a status report on the completion of the whole request following an "OK - Acknowledge" has not been received in a timely fashion, the requester may assume that either the request or response failed to get through. For example, the requester may consider requesting the status of the XID in question to see whether the prior request has been actioned (e.g. ActivateTask, ModifyTask, DeactivateTask or DeactivateAllTasks) or the requester may re-send the original request (as a new request, with a new X1TransactionID).

5.3   Warnings and Faults

The present document uses the term "error" to mean a protocol error within the X1 protocol as defined in clause 6.7.

All other problems are categorized as warnings, alerts or faults:

- Warnings are one-off problems i.e. sent by the NE and then not referred to again over X1. Warnings shall not be used for issues which are affecting traffic (i.e. losing content or intercept-related information). For example, warnings may include resources being nearly exhausted but not yet traffic-affecting. Warnings should include that keys/certificates are about to expire.

- Alerts are one-off problems that might affect traffic (e.g. cleared database).

- Faults are problems which the NE will continue to be aware of and which the NE is trying to manage and/or rectify. Any issue which loses traffic is categorized as a fault.

Warnings and alerts are reported using issue-reporting messages (clause 6.5) but then are not included in any future Status-Getting messages (see clause 6.4). The NE shall log any warnings and alerts for audit reasons.

The NE shall remember which of the XIDs are in fault and whether the NE itself is in a fault situation. An issue report (see clause 6.5) is required at the start of the fault. The NE shall report faults when responding to the Status-Getting message defined in clause 6.4. The NE shall also indicate that a fault has been cleared (see clauses 6.5.2 and 6.5.3) unless otherwise configured.

6   Message Structure and Data Definitions

6.1   X1 Message details

X1 messages contain information as defined in table 1 (the information is Mandatory, Optional or Conditional as shown in the last column).

Table 1: Message details

Field	Description	Format	Mandatory (M), Optional (O) or Conditional (C)
ADMF Identifier	Identifies the ADMF uniquely to the NE. Required to match the details provided by the ADMF's X.509 certificate (see clause 8)	Token as per W3C® Recommendation [3], section 3.4.2. Definition and assignment of identifiers is a deployment issue	M
NE Identifier	Uniquely identifies the NE to the ADMF. Required to match the details provided by the NE's X.509 certificate (see clause 8)	Token as per W3C® Recommendation [3], section 3.4.2. Definition and assignment of identifiers is a deployment issue	M
MessageTimestamp	Timestamp indicating the time the message was sent	See ETSI TS 103 280 [4] Qualified Microsecond Date Time	M
Version	Version of the present document used for encoding the message	See clause 4.5	M
X1TransactionID	Used to correlate Request and Response. Shall be omitted for "TopLevelError" situations as defined below this table but otherwise is mandatory	An ID as defined in clause 5.2	C

In addition to the information in table 1, the X1 Request shall indicate the type of request being made (see clauses 6.2 to 6.6), and contain the appropriate request parameters for that type of request.

If the X1 Request could not be parsed, then the response shall be constructed with an ADMF and NE Identifier (extracting the identifier of the Requester from the X.509 certificate if necessary), MessageTimestamp and Version, and a "TopLevelError" flag but no other information.

If the request could be parsed then the response shall indicate the type of response being returned (see clauses 6.2 to 6.6) and contain the appropriate response parameters for that type of response.

A "RequestContainer" is used to contain one or more requests. All requests in a container are delivered at the same time, from the same Requester and to the same Responder. There is no implication about which order they are processed; for this reason, the ADMF should avoid sending ActivateTask and ModifyTask messages for the same XID in the same RequestContainer. A "ResponseContainer" is used to contain all the responses to the requests in the container. The ordering of these responses does not have a meaning. All responses are sent at the same time, from the same Responder and to the same Requester. The RequestContainer and ResponseContainer shall be used even if there is one request and one response.

For each "OK - Acknowledged" response received for the requests transported by a "RequestContainer", the requester should implement logic to assure the related status report is received and the transaction is completed or initiate a recovery procedure.

6.2 Message definitions: starting, modifying and stopping tasks

6.2.1 ActivateTask

6.2.1.1 Summary

DIRECTION: ADMF to NE.

USAGE: Used by the ADMF to add a new Task to an NE.

Table 2: ActivateTaskRequest

Field	Description	Format	M/C/O
TaskDetails	Target and interception details	See clause 6.2.1.2	M

Table 3: ActivateTaskResponse

Field	Description	Format	M/C/O
OK or Error	The general errors in clause 6.7 apply. Also, it is an error if the XID is already present at the NE	See clause 6.7	M

6.2.1.2 TaskDetails

The TaskDetails structure shall include the following.

Table 4: TaskDetails

Field	Description	Format	M/C/O
XID	Uniquely identifies the Task. There may be more than one different Task relating to the same target identifier (two distinct XIDs). The X1 interface supports delivery for this situation (i.e. it is not considered an error on the X1 interface).	UUIDv4 (see clause 5.1).	M
TargetIdentifiers	List of criteria which are used to identify the traffic to be intercepted. Where multiple criteria are present, all criteria are required to be matched. If an NE cannot target based on the criteria specified (e.g. due to an unsupported format or inappropriate combination of identifiers) the NE shall reject the request with an appropriate error. It is an implementation decision which identifiers and combinations of identifiers are supported.	Each TargetIdentifier given follows one of the formats given in table 5.	M
DeliveryType	Statement of whether to deliver X2 and/or X3. An MDF shall ignore the contents of the field, and use the DeliveryType value given in the relevant MediationDetails structure (see annex C).	Enumerated value - one of "X2Only", "X3Only" and "X2andX3".	M

Field	Description	Format	M/C/O
ListOfDIDs	Details of where to send the intercepted traffic. It is an implementation decision for the NE to determine how to duplicate traffic if multiple destinations and/or destination sets are specified, or if multiple destinations or destination sets are supported.	List of Destination Identifiers (DID) and/or List of Destination Set Identifiers (DSID) referencing the desired delivery destination records.	M
ListOfMediationDetails	Set of details for use by an NE that is performing mediation (i.e. a mediation and delivery function). This shall be included between the ADMF and the MDF. Multiple instances of this parameter may be included (e.g. when multiple LIIDs are associated with an XID).	See annex C.	C
CorrelationID	Correlation identifier to assign to intercepted material for this Task. Intended for use in triggering scenarios, and shall be ignored by non-mediation function NEs.	Unsigned integer.	O
ImplicitDeactivationAllowed	Indication that a Task may implicitly deactivate itself once the NE has determined that it has completed. On deactivation of the Task, the NE shall issue a ReportTaskIssue message with the appropriate TaskReportType (see clause 6.5.2).	Boolean.	O
ProductID	When provided, shall be used by the receiving entity to populate the X2/X3 XID header as per ETSI TS 103 221-2 [19], clause 5.2.7 instead of the XID of the Task. If not provided, the XID of the Task shall be used.	UUIDv4.	O
ListOfServiceTypes	Shall be included when explicitly identifying the CSP-provided service(s) to be reported for this task. Details of the use of this field are left to the relevant LI architecture.	One or more of the enumerated values of the ServiceType field as listed in table C.2.	C
TaskDetailsExtensions	One or more extension placeholders; each may be populated by a list of elements defined by external specifications.	See annex B.	O

If a Task has an invalid combination of DeliveryType and Destinations (e.g. "X2andX3" delivery specified, but only an X2 Destination given), then the NE shall reject the ActivateTaskRequest with an appropriate error.

If a Task has a ServiceType not supported by the NE, then the NE shall reject the ActivateTaskRequest with an appropriate error. If the expected services to which interception applies are the only services that an NE provides, then inclusion of ServiceType to the LI function in that NE is not necessary. If the ServiceType is not included, then interception applies to all services supported by the NE.

The list of permissible TargetIdentifier formats is given in table 5.

Table 5: TargetIdentifier Formats

Format Name	Description	Format
E164Number	E.164 Number in fully international format, written as decimal digits	Given in ETSI TS 103 280 [4] InternationalE164 format
IMSI	International Mobile Subscriber Identity, following the Recommendation ITU-T E.212 [5] numbering scheme, written as decimal digits	Given in ETSI TS 103 280 [4] IMSI format
IMEI	International Mobile station Equipment Identity, following the numbering plan defined in 3GPP TS 23.003 [6], written as decimal digits without the (Luhn) check digit	Given in ETSI TS 103 280 [4] IMEI format
MACAddress	A MAC address	Given in ETSI TS 103 280 [4] MACAddress format

Format Name	Description	Format
IPv4Address	An IPv4 address	Given in ETSI TS 103 280 [4] IPv4Address format
IPv6Address	IPv6 address	Given in ETSI TS 103 280 [4] IPv6Address format
IPv4CIDR	IPv4CIDR, written in dotted decimal notation followed by CIDR notation	Given in ETSI TS 103 280 [4] IPv4CIDR format
IPv6CIDR	IPv6CIDR written as eight groups of four hexadecimal digits separated by a colon, followed by CIDR notation	Given in ETSI TS 103 280 [4] IPv6CIDR format
TCPPort	TCP Port number, written in decimal notation	Given in ETSI TS 103 280 [4] TCPPort format
TCPPortRange	Range of TCP Ports, written as decimal numbers separated by a colon	Given in ETSI TS 103 280 [4] TCPPortRange format
UDPPort	UDP Port number, written in decimal notation	Given in ETSI TS 103 280 [4] UDPPort format
UDPPortRange	Range of UDP Ports, written as decimal numbers separated by a colon	Given in ETSI TS 103 280 [4] UDPPortRange format
EmailAddress	Email address	Given in ETSI TS 103 280 [4] EmailAddress format
InternationalizedEmailAddress	Email address following IETF RFC 6530 [23]	Given in ETSI TS 103 280 [4] InternationalizedEmailAddress format
SIP-URI	SIP-URI according to the SIP URI scheme given in IETF RFC 3261 [7]	Given in ETSI TS 103 280 [4] SIPURI format
TEL-URI	TEL-URI according to the TEL URI scheme (see IETF RFC 3966 [8]) Implementers should consider whether the value could be sent as an E.164 number (or one of the related types) instead	Given in ETSI TS 103 280 [4] TELURI format
H323-URI	H323 URI according to the H323 URI scheme (see IETF RFC 3508 [9])	Given in H323Uri format (see XSD schema)
IMPU	IP Multimedia Public Identity, as per 3GPP TS 23.003 [6]	Given in IMPU format (see XSD schema)
IMPI	IP Multimedia Private Identity, as per 3GPP TS 23.003 [6]	Given in IMPI format (see XSD schema)
NAI	Network Access Identifier following IETF RFC 7542 [10] format	Given in ETSI TS 103 280 [4] NAI format
RADIUS	Any Radius attribute that uniquely identifies the subscriber within the specific CSP (see note 1)	Given as binary octets containing RADIUS AVP following IETF RFC 2865 [11], clause 5 (see note 2)
GTPUTunnelId	GTP-U Tunnel Identifier	Given as a 32-bit integer
GTPCTunnelId	GTP-C Tunnel Identifier	Given as a 32-bit integer
CallPartyRole	Identifies the role of a party in a call. Intended for use in conjunction with e.g. E164Number	One of the values "Originating", "Terminating", "ForwardedTo"
NonLocalIdentifier	Identifies whether the identifier is local or non-local. Intended for use in conjunction with e.g. E164Number	One of the values "Local" or "NonLocal"
SUPIIMSI	Subscription Permanent Identifier in IMSI format	Given in ETSI TS 103 280 [4] SUPIIMSI format
SUPINAI	Subscription Permanent Identifier in NAI format	Given in ETSI TS 103 280 [4] SUPINAI format
SUCI	Subscription Concealed identifier	Given in ETSI TS 103 280 [4] SUCI format
PEIIMEI	Permanent Equipment Identifier in IMEI format	Given in ETSI TS 103 280 [4] PEIIMEI format
PEIIMEICheckDigit	Permanent Equipment Identifier in IMEICheckDigit format	Given in ETSI TS 103 280 [4] PEIIMEICheckDigit format
PEIIMEISV	Permanent Equipment Identifier in IMEISV format	Given in ETSI TS 103 280 [4] PEIIMEISV format
GPSIMSISDN	General Purpose Subscription Identifier in MSISDN format	Given in ETSI TS 103 280 [4] GPSIMSISDN format
GPSINAI	General Purpose Subscription Identifier in NAI format	Given in ETSI TS 103 280 [4] GPSINAI format
EUI64	64 bit Extended Unique Identifier following IETF RFC 7042 [27] format.	Given in ETSI TS 103 280 [4] EUI64 format
HashedIdentifier	Hashed target identifier	See annex D

Format Name	Description	Format
TargetIdentifierExtension	Identifier defined by an external specification	See annex B
NOTE 1: Future versions of the present document may need to consider temporary identifiers including pseudonyms or short-term identifiers which have been derived from the permanent identifiers.		
NOTE 2: Depending on NE implementation, this may not be exactly the same binary representation used to match traffic e.g. for case-insensitive matching.		

6.2.2 ModifyTask

DIRECTION: ADMF to NE.

USAGE: Used by the ADMF to modify an existing Task on the NE. All details for the Task shall be given (i.e. the modified details and the information that is unchanged) to totally replace the previous Task details.

Depending on the NE implementation, it may not be possible to modify some or all of the Task details. If the NE cannot modify one or more of the elements in the ModifyTaskRequest, it shall reject the entire ModifyTaskRequest with an appropriate error response.

The length of time an NE requires to make the changes requested in the ModifyTaskRequest message is an implementation detail, but the expectation is that changes are made without undue delay.

Table 6: ModifyTaskRequest

Field	Description	Format	M/C/O
Task details	Target and interception details (same as for ActivateTaskRequest)	See clause 6.2.1.2	M

Table 7: ModifyTaskResponse

Field	Description	Format	M/C/O
OK or Error	The general errors in clause 6.7 apply. Also, it is an error if the XID is not already present	See clause 6.7	M

6.2.3 DeactivateTask

DIRECTION: ADMF to NE.

USAGE: Used by the ADMF to deactivate (permanently stop and remove) a Task on the NE.

There is no concept of suspension or temporary deactivation. To stop a Task "temporarily", ADMFs shall deactivate the Task and then activate a new Task.

Table 8: DeactivateTaskRequest

Field	Description	Format	M/C/O
XID	See clause 5.1	See clause 5.1	M

Table 9: DeactivateTaskResponse

Field	Description	Format	M/C/O
OK or Error	The general errors in clause 6.7 apply. Also, it is an error if the XID is not already present at the NE	See clause 6.7	M

6.2.4   DeactivateAllTasks

DIRECTION: ADMF to NE.

USAGE: If enabled, the DeactiveAllTasks command shall perform a "DeactiveTask" command for all Tasks
on the NE.

Table 10: DeactivateAllTasksRequest

Field	Description	Format	M/C/O
There shall be no request parameters			

Table 11: DeactivateAllTasksResponse

Field	Description	Format	M/C/O
OK or Error	The general errors in clause 6.7 apply. See below regarding whether "DeactivateAllTasks" is enabled; if Disabled then DeactivateAllTasks always triggers an error response of type "DeactivateAllTasks message is not enabled"	See clause 6.7	M

The DeactiveAllTasks request shall be supported by all implementations of the present document. It should be agreed in advance as to whether the DeactivateAllTasks request is enabled or disabled. By default (if there has been no agreement in advance) then DeactivateAllTasks is enabled.

6.3   Message definitions: creating, modifying and removing Destinations

6.3.1   CreateDestination

6.3.1.1   Summary

DIRECTION: ADMF to NE.

USAGE: Used by the ADMF to add a new Destination to the NE.

Table 12: CreateDestinationRequest

Field	Description	Format	M/C/O
Destination details	Details of the new destination	See clause 6.3.1.2	M

Table 13: CreateDestinationResponse

Field	Description	Format	M/C/O
OK or Error	The general errors in clause 6.7 apply. Also, it is an error if the DID is already present at the NE	See clause 6.7	M

6.3.1.2 DestinationDetails

DestinationDetails relate to the delivery of information from the NE to a Destination.

The DestinationDetails structure is defined as follows.

Table 14: DestinationDetails

Field	Description	Format	M/C/O
DID	Destination Identifier which uniquely identifies the destination	UUIDv4 (see clause 5.1)	M
FriendlyName	A human-readable name associated with the delivery destination	Free-text string	O
DeliveryType	Statement of whether to deliver X2 and/or X3 to this destination	Enumerated value - one of "X2Only", "X3Only" and "X2andX3"	M
DeliveryAddress	One of the values from table 15 shall be included	As defined in table 15	M
DestinationDetails Extensions	One or more extension placeholders; each may be populated by a list of elements defined by external specifications	See annex B	O

The DeliveryAddress structure is defined as follows.

Table 15: DeliveryAddress

Field	Description	Format
IPAddressAndPort	This covers both IPv4 and IPv6 and contains a single IP Address and Port	IPAddressAndPort from ETSI TS 103 280 [4]
E164Number	E.164 destination	InternationalE164 (see ETSI TS 103 280 [4])
URI	URI destination (e.g. an FQDN or other form of URI)	anyURI (see W3C® Recommendation [3], section 3.2.17)
EmailAddress	Email address of the destination	EmailAddress (see ETSI TS 103 280 [4])

6.3.2 ModifyDestination

DIRECTION: ADMF to NE.

USAGE: Used by the ADMF to modify an existing Destination on the NE. All details for the Destination shall be given (i.e. the modified details and the information that is unchanged) to totally replace the previous Destination details.

Depending on the NE implementation, it may not be possible to modify some or all Destination details while the Destination is in use. If the NE cannot modify one or more of the elements in the ModifyDestinationRequest, it shall reject the entire ModifyDestinationRequest with an appropriate error response.

The length of time an NE requires to make the changes requested in the ModifyDestinationRequest message is an implementation detail, but the expectation is that changes are made without undue delay.

Table 16: ModifyDestinationRequest

Field	Description	Format	M/C/O
DestinationDetails	Updated details for the destination	See clause 6.3.1.2	M

Table 17: ModifyDestinationResponse

Field	Description	Format	M/C/O
OK or Error	The general errors in clause 6.7 apply. Also, it is an error if the DID is not present	See clause 6.7	M

6.3.3   RemoveDestination

DIRECTION: ADMF to NE.

USAGE: Used by the ADMF to remove a Destination from the NE.

A Destination may only be removed if it is not referenced by any Tasks. An NE shall respond with an appropriate error if the ADMF attempts to remove a Destination that is referenced by a Task.

Table 18: RemoveDestinationRequest

Field	Description	Format	M/C/O
DID	See clause 5.1	See clause 5.1	M

Table 19: RemoveDestinationResponse

Field	Description	Format	M/C/O
OK or Error	The general errors in clause 6.7 apply. Also, it is an error if the DID is not already present at the NE	See clause 6.7	M

6.3.4   RemoveAllDestinations

DIRECTION: ADMF to NE.

USAGE: To completely and permanently remove all Destinations on the NE.

Table 20: RemoveAllDestinationsRequest

Field	Description	Format	M/C/O
There shall be no message parameters			

Table 21: RemoveAllDestinationsResponse

Field	Description	Format	M/C/O
OK or Error	The general errors in clause 6.7 apply. See below regarding whether "RemoveAllDestinations" is enabled; if Disabled then RemoveAllDestinations always triggers an error response	See clause 6.7	M

The RemoveAllDestinations request shall be supported by all implementations of the present document.

It shall be agreed in advance as to whether the RemoveAllDestinations request is enabled or disabled. By default (if there has been no agreement in advance) then RemoveAllDestinations is enabled.

If RemoveAllDestinations is disabled, then a RemoveAllDestinations request shall always trigger an ErrorResponse indicating "RemoveAllDestinations request is not enabled".

If RemoveAllDestinations is enabled, then a RemoveAllDestinations request shall remove all Destinations on that NE, or it shall trigger an error for the general error conditions listed in clause 6.7. Since a RemoveDestination request can only be issued against destinations that are not in use, an NE shall respond with an error if the ADMF sends a RemoveAllDestinations request while any of the Destinations are referenced by Tasks.

6.4   Message details: getting information from NE

6.4.1   Overview

This clause defines messages for the ADMF to request status information from the NE. This is distinct from "Reporting Issues" where the NE pushes information to the ADMF (see clause 6.5).

The following requests and responses shall be supported:

- GetTaskDetails: to request details of a single Task.

- GetDestinationDetails: to request details of a single Destination.

- GetNEStatus: to request status of the NE itself.

- GetAllDetails: requests details of all Tasks, Destinations, Generic Objects and the status of the NE itself.

- GetAllTaskDetails: requests details of all Tasks.

- GetAllDestinationDetails: requests details of all Destinations.

- GetAllGenericObjectDetails: requests details of all Generic Objects.

- ListAllDetails: requests the XIDs of all Tasks, DIDs of all Destinations and Object IDs of all Generic Objects (i.e. not all the details).

6.4.2   GetTaskDetails

6.4.2.1   Summary

DIRECTION: ADMF to NE.

USAGE: Used by the ADMF to retrieve the details of a particular Task.

Table 22: GetTaskDetailsRequest

Field	Description	Format	M/C/O
XID	See clause 5.1	See clause 5.1	M

Table 23: GetTaskDetailsResponse

Field	Description	Format	M/C/O
TaskResponseDetails	The Task details are as per clause 6.2.1.2, additionally containing a TaskStatus structure as per clause 6.4.2.2, unless there is an error, in which case see clause 6.7. If the XID is not present, this is an error (the appropriate error code shall be used, see clause 6.7)	See clauses 6.2.1.2 and 6.4.2.2	M

6.4.2.2   TaskStatus

The TaskStatus contains information about a Task as collected internally by the NE.

Table 24: TaskStatus

Field	Description	Format	M/C/O
ProvisioningStatus	Indicates whether the Task has been provisioned ("complete"), has failed to provision ("failed") or whether it is awaiting provisioning ("awaitingProvisioning")	One of the values "awaitingProvisioning", "failed" or "complete"	M
ListOfFaults	List of all active faults on that Task. If there are no faults, the field shall be omitted	List of ErrorInformation structures (see clause 6.7)	C
TimeOfLastIntercept	Time of last traffic intercepted if any (omit if none seen so far or as provided beneath this table) This time may also be updated periodically (instead of per packet) if required due to performance reasons	See ETSI TS 103 280 [4], Qualified Microsecond Date Time	C
AmountOfX2Data	Data transmitted over X2 since the creation of the Task in bytes, summed across all Destinations. This field shall be included unless the exception beneath this table applies. If given, shall be correct at the time given in TimeOfLastIntercept	Integer	C
AmountOfX3Data	Data transmitted over X3 since the creation of the Task in bytes, summed across all Destinations. This field shall be included unless the exception beneath this table applies. If given, shall be correct at the time given in TimeOfLastIntercept	Integer	C
TimeOfLastModification	Time of the last modification to the Task (omit only if unmodified or as provided beneath this table)	See ETSI TS 103 280 [4], Qualified Microsecond Date Time	C
NumberOfModifications	Number of successful modifications since start. This field shall be included unless the exception beneath this table applies	Integer	C

For any of the following fields: TimeOfLastIntercept, AmountOfX2Data, AmountOfX3Data, TimeOfLastModification and NumberOfModifications, if the functionality needed to determine information for a field is not implemented by an NE, the field shall always be omitted.

6.4.3 GetDestinationDetails

6.4.3.1 Summary

DIRECTION: ADMF to NE.

USAGE: Used by the ADMF to retrieve the details of a particular Destination.

Table 25: GetDestinationRequest

Field	Description	Format	M/C/O
DID	See clause 5.1	See clause 5.1	M

Table 26: GetDestinationResponse

Field	Description	Format	M/C/O
DestinationResponseDetails	The destination details are as per table 14, additionally containing a DestinationStatus structure as per clause 6.4.3.2, unless there is an error, in which case see clause 6.7. If the DID is not present, this is an error (the appropriate error code shall be used, see clause 6.7)	See clauses 6.3.1.2 and 6.4.3.2	M

6.4.3.2 DestinationStatus

The DestinationStatus relates only to the status of the delivery Destination as seen by the NE.

Table 27: DestinationStatus

Field	Description	Format	M/C/O
DestinationStatus	Status of Destination. Indicating whether the destination is active and working, or whether there is a delivery fault and traffic being lost. It is possible in the DeliveryFault state that some traffic is still being delivered - the determining factor is that issues with delivery to this destination is causing some traffic to be lost	One of "ActiveAndWorking" or "DeliveryFaults"	M
ListOfFaults	List of all active faults on that Destination	List of ErrorInformation structures (see clause 6.7)	M

6.4.4 GetNEStatus

6.4.4.1 Summary

DIRECTION: ADMF to NE.

USAGE: Used by the ADMF to determine the status of the NE.

Table 28: GetNEStatusRequest

Field	Description	Format	M/C/O
There shall be no request parameters			

Table 29: GetNEStatusResponse

Field	Description	Format	M/C/O
NEStatusDetails	The NEStatusDetails for the NE. The NE Status shall be one of "OK" i.e. no NE faults, or "Faults" i.e. NE losing traffic (these are separate from delivery faults which are reported per XID). Additionally, a list of currently unresolved faults (list of ErrorInformation items) shall be included (previous warnings are not included here)	Enumerated NEStatus value - one of "OK" or "Faults". List of ErrorInformation structures (see clause 6.7)	M

6.4.5 GetAllDetails

6.4.5.1 Summary

DIRECTION: The GetAllDetails command goes from ADMF to NE.

USAGE: For the ADMF to determine the details of all Tasks, Destinations and the status of the NE itself.

Table 30: GetAllDetailsRequest

Field	Description	Format	M/C/O
There shall be no request parameters			

Table 31: GetAllDetailsResponse

Field	Description	Format	M/C/O
NEStatusDetails	The NEStatusDetails for the NE The NEStatus shall be one of "OK" i.e. no NE faults, or "Faults" i.e. NE losing traffic (these are separate from delivery faults which are reported per XID) Additionally, a list of currently unresolved faults (list of ErrorInformation items) shall be included (previous warnings are not included here)	Enumerated NEStatus value - one of "OK" or "Faults" List of ErrorInformation structures (see clause 6.7)	M
ListOfTaskResponseDetails	The response shall include TaskResponseDetails structures for all Tasks present on the NE. If there are no Tasks, an empty list shall be returned - this is not an error	See clauses 6.2.1.2 and 6.4.2.2	M
ListOfDestinationResponse Details	The response shall include DestinationResponseDetails structures for all destinations present on the NE. If there are no destinations, an empty list shall be returned - this is not an error	See clauses 6.3.1.2 and 6.4.3.2	M
ListOfGenericObjectDetails	The response shall include Generic Object details for every object present on the NE. If there are no such objects, an empty list shall be returned - this is not an error. May be omitted if Generic Objects are not supported by the NE	See clause 6.8	C

6.4.6   ListAllDetails

6.4.6.1   Summary

DIRECTION: ADMF to NE.

USAGE: Used by the ADMF to retrieve the list of all XIDs and DIDs (i.e. a list of identifiers) but no details.

Table 32: ListAllDetailsRequest

Field	Description	Format	M/C/O
There shall be no request parameters				

Table 33: ListAllDetailsResponse

Field	Description	Format	M/C/O
ListOfXIDs	A list of all XIDs on the NE. If there are none, then an empty list is returned; this is not an error	List of XIDs	M
ListOfDIDs	A list of all DIDs on the NE. If there are none, then an empty list is returned; this is not an error	List of DIDs	M
ListOfGenericObjectIDs	A list of all Generic Object IDs on the NE (see clause 6.8.1.3). If there are none, an empty list is returned - this is not an error. May be omitted if Generic Objects are not supported by the NE	List of objectIDs	C

6.4.7   GetAllTaskDetails

6.4.7.1   Summary

DIRECTION: The GetAllTaskDetails command goes from ADMF to NE.

USAGE: For the ADMF to determine the details of all Tasks.

Table 30a: GetAllTaskDetailsRequest

Field	Description	Format	M/C/O
There shall be no request parameters			

Table 31a: GetAllTaskDetailsResponse

Field	Description	Format	M/C/O
ListOfTaskResponseDetails	The response shall include TaskResponseDetails structures for all Tasks present on the NE. If there are no Tasks, an empty list shall be returned - this is not an error	See clauses 6.2.1.2 and 6.4.2.2	M

6.4.8   GetAllDestinationDetails

6.4.8.1   Summary

DIRECTION: The GetAllDestinationDetails command goes from ADMF to NE.

USAGE: For the ADMF to determine the details of all Destinations.

Table 30b: GetAllDestinationsDetailsRequest

Field	Description	Format	M/C/O
There shall be no request parameters			

Table 31b: GetAllDestinationsDetailsResponse9

Field	Description	Format	M/C/O
ListOfDestinationResponseDetails	The response shall include DestinationResponseDetails structures for all destinations present on the NE. If there are no destinations, an empty list shall be returned - this is not an error	See clauses 6.3.1.2 and 6.4.3.2	M

6.4.9   GetAllGenericObjectDetails

6.4.9.1   Summary

DIRECTION: The GetAllGenericObjectDetails command goes from ADMF to NE.

USAGE: For the ADMF to determine the details of all Generic Objects.

Table 30c: GetAllGenericObjectDetailsRequest

Field	Description	Format	M/C/O
ObjectType	If present, only the specific XSD type is required rather than the whole Generic Object Details	URIQualifiedName (as defined in XPath 3.1 [24] definition 117	O

Table 31c: GetAllGenericObjectDetailsResponse

Field	Description	Format	M/C/O
ListOfGenericObjectDetails	The response shall include Object details for every object present on the NE or details for object of the specified objectType (if present in the request). If there are no such objects, an empty list shall be returned - this is not an error. May be omitted if Generic Objects are not supported by the NE	See clause 6.8	C

6.5   Message details: reporting issues from the NE

6.5.1   Overview

This clause defines request types for the NE to report issues to the ADMF. It is distinct from "Getting Status", in which the ADMF retrieves information from the NE (see clause 6.4).

Issues may be:

- Relating to a particular XID (including delivery issues with that XID).

- Relating to a particular DID.

- Relating to the whole NE.

6.5.2   ReportTaskIssue on given XID

6.5.2.1   Summary

DIRECTION: NE to ADMF.

USAGE: The NE shall send a ReportTaskIssue request when it becomes aware of an issue (warning or fault) relating specifically to a particular XID. It shall also be used to follow up on an "OK - Acknowledged" response, to signal that a request has been completed (clause 5.2) successfully or unsuccessfully.

Faults and warnings are defined in clause 5.3; see also clause 5.1 about terminating and non-terminating faults.

If a non-terminating fault becomes terminating, the NE shall send another ReportTaskIssue.

If a non-terminating fault is cleared, the NE shall send another ReportTaskIssue indicating the fault is cleared.

Table 34: ReportTaskIssueRequest

Field	Description	Format	M/C/O
XID	See clause 5.1	See clause 5.1	M
TaskReportType	Type of Issue	See clause 6.5.2.2	M
TaskIssueErrorCode	Error code associated with the issue, if appropriate	See clause 6.7	O
TaskIssueDetails	Further description of issue if appropriate	Free text	O

Table 35: ReportTaskIssueResponse

Field	Description	Format	M/C/O
OK or Error	The general errors in clause 6.7 apply	See clause 6.7	M

It is possible that the ADMF is not aware of the XID which is referenced in the NE message. The ADMF shall not send an error back to the NE in this situation: it is for the ADMF to decide how to handle this (e.g. GetAllDetails or GetAllTaskDetails or Deactivate the XID in question are possible approaches).

6.5.2.2   Task report types

The TaskReportType shall be one of the following:

- All clear: non-terminating fault resolved.

- Warning: not traffic-affecting.

- Non-terminating fault (currently unable to collect traffic but not terminating).

- Terminating fault. The message is used by the NE to indicate that the Task has experiences a terminating fault and has been deactivated.

- Implicit Deactivation: A Task with the "ImplicitDeactivationAllowed" flag has been deactivated.

- Actioned: Request has been fully actioned and was successful (to follow up on "OK - Acknowledged" response from clause 5.2).

- Failed: Request has been fully actioned but was unsuccessful (to follow up on "OK - Acknowledged" response from clause 5.2). This is a terminating fault.

6.5.3   ReportDestinationIssue on given DID

6.5.3.1   Summary

DIRECTION: NE to ADMF.

USAGE: The NE shall send a ReportDestinationIssue request when it becomes aware of an issue (warning or fault) relating specifically to a particular DID. It shall also be used to follow up on an "OK - Acknowledged" response, to signal that a request has been completed (clause 5.2) successfully or unsuccessfully.

Faults and warnings are defined in clause 5.3; see also clause 5.1 about terminating and non-terminating faults.

If a non-terminating fault becomes terminating, the NE shall send another ReportDestinationIssue.

If a non-terminating fault is cleared, the NE shall send another ReportDestinationIssue indicating the fault is cleared.

Table 36: ReportDestinationIssueRequest

Field	Description	Format	M/C/O
DID	See clause 5.1	See clause 5.1	M
DestinationReportType	Type of Issue	Same as TaskReportType, see clause 6.5.2.2	M
DestinationIssueErrorCode	Error code for the issue, if appropriate	See clause 6.7	O
DestinationIssueDetails	Further description of issue if appropriate	Free text	O

Table 37: ReportDestinationIssueResponse

Field	Description	Format	M/C/O
OK or Error	The general errors in clause 6.7 apply	See clause 6.7	M

6.5.4   ReportNEIssue

DIRECTION: NE to ADMF.

USAGE: The NE shall send a ReportNEIssue request when it becomes aware of an issue (warning, alert or fault) relating to the whole NE.

NE issues can relate to:

- Any hardware or software issue on NE (storage nearly full, power issue).

- Current security issue on NE.

- Any issues with logging or audit material.

- Any report from manual changes to NE configuration.

- Any report of databases being cleared in the NE.

Table 38: ReportNEIssueRequest

Field	Description	Format	M/C/O
TypeOfNEIssue Message	Indicates the type of message (Warning, Fault Cleared, Fault Report, Alert)	One of the following: "Warning", "FaultCleared", "FaultReport", "Alert"	M
Description	Description of the issue being reported	Free text	M
IssueCode	Integer code indicating the distinct issue information if TypeOfNEIssueMessage is "Alert" and the error code is part of the issue codes section in table 46	Integer	C

Table 39: ReportNEIssueResponse

Field	Description	Format	M/C/O
OK or Error	The general errors in clause 6.7 apply	See clause 6.7	M

6.6   Message details: pings and keepalives

6.6.1   Ping

DIRECTION: Either direction.

USAGE: At any time from the ADMF or NE, to get a response over the X1 interface (does not test X2 or X3 or onward delivery).

Table 40: PingRequest

Field	Description	Format	M/C/O
There shall be no request parameters			

Table 41: PingResponse

Field	Description	Format	M/C/O
OK or Error	The OK response has no other content. The general errors in clause 6.7 apply.	See clause 6.7	M

6.6.2   Keepalive

DIRECTION: The Keepalive command goes from ADMF to NE.

USAGE: See below.

Table 42: KeepaliveRequest

Field	Description	Format	M/C/O
There shall be no request parameters			

Table 43: KeepaliveResponse

Field	Description	Format	M/C/O
OK or Error	The OK message has no other content. The general errors in clause 6.7 apply.	See clause 6.7	M

The Keepalive functionality shall be supported by NE and ADMF. It is for prior agreement to determine whether Keepalives are enabled or disabled. By default (with no prior agreement) they are enabled. It is intended as a means for the NE application to assert that the ADMF application is still operational, and remove all tasking information as a security measure if it is not.

If Keepalives are enabled, the ADMF shall send out a Keepalive message at least every TIME_P1 (by default TIME_P1 is 1 minute) if no other X1 request has been sent to the NE.

If Keepalives are enabled, the NE shall respond with an OK for each Keepalive; if the NE has not seen a Keepalive message for TIME_P2 (by default TIME_P2 is 1 hour) then the NE shall perform a DeactivateAllTasks command i.e. deactivate all XIDs on the NE. The NE implementation shall reset the timer whenever any X1 Request is received from the ADMF (including a Keepalive Request).

6.7   Protocol error details

If the Responder is unable to perform an action requested as part of a Request Message, then it shall respond to that Request Message with an Error Response.

An ErrorResponse is a response which has the information from clause 6.1, but the response body has an error code from the list below and a free text field for further information. It has the following structure.

Table 44: ErrorResponse

Field	Description	Format	M/C/O
RequestMessageType	Indicates the type of Request Message that the Error Response message is a response to	One of the following: "ActivateTask", "ModifyTask", "DeactivateTask", "DeactivateAllTasks", "GetTaskDetails", "CreateDestination", "ModifyDestination", "RemoveDestination", "RemoveAllDestinations", "GetDestinationDetails", "GetNEStatus", "GetAllDetails", "GetAllTaskDetails", "GetAllDestinationDetails", "GetAllGenericObjectDetails", "ListAllDetails", "ReportTaskIssue", "ReportDestinationIssue", "ReportNEIssue", "Ping", "Keepalive", "ExtendedRequestMessageType"	M
ErrorInformation	Error code and optional description for the error	ErrorInformation as defined in table 45	M
ExtensionInformation	Indicates the specification of the extension and the extended type of the Request Message that the Error Response message is a response to if the RequestMessageType is "ExtendedRequestMessageType"	ExtensionInformation as defined in table 46a	C

Table 45: ErrorInformation

Field	Description	Format	M/C/O
ErrorCode	Integer code indicating the type of error (see table 46)	Integer	M
ErrorDescription	Free text field giving further details of the error. Implementers are encouraged to avoid placing sensitive information (such as personally identifiably information or sensitive details of the network) in error messages.	UTF-8 string	M

The ErrorResponse is used only as a response to a request which could not be actioned or understood. It is different from reporting on the status of the Task which are called "faults" and "warnings" but not "protocol errors".

Table 46: Error codes

Error Code	Error Description	Suggested Information elements
General message errors		
1000	Generic error	Details of the error
1010	Syntax/schema error	Details of the schema or syntax error
1020	Unsupported version	Version supported by the issuing system
1030	ADMF Identifier does not match certificate details	None
1040	Unexpected ADMF Identifier	None
1050	NE Identifier does not match certificate details	None
1060	Unexpected NE Identifier	None
1070	Keepalive not supported	None
1080	Unsupported request	None
Identifier errors		
2010	XID already exists on NE	XID in question
2020	XID does not exist on NE	XID in question
2030	DID already exists on the NE	DID in question
2040	DID does not exist on the NE	DID in question
2050	GenericObjectID already exists on the NE	GenericObjectID in question
2060	GenericObjectID does not exist on the NE	GenericObjectID in question
ActivateTask/ModifyTask errors		
3000	Generic ActivateTask failure	Details of why the Task cannot be activated
3001	Generic ModifyTask failure	Details of why the Task cannot be modified
3010	Unsupported TargetIdentifier type	Details of the unsupported TargetIdentifier type
3020	Unsupported combination of TargetIdentifiers	Details of the unsupported combination
3030	Multiple destinations not supported	None
3040	Invalid combination of DeliveryType and Destinations specified	None
3050	Unsupported ServiceType	Details of the unsupported ServiceType
DeactivateTask failures		
4000	Generic DeactivateTask failure	Details of why the Task cannot be deactivated
DeactivateAllTasks failures		
5000	Generic DeactiveAllTasks failure	Details of why all Tasks cannot be removed
5010	DeactivateAllTasks not enabled	None
CreateDestination/ModifyDestination failures		
6000	Generic CreateDestination failure	Details of why the Destination cannot be created
6001	Generic ModifyDestination failure	Details of why the Destination cannot be modified
6020	Unsupported DeliveryAddress type	Details of the DeliveryAddress type requested
RemoveDestination failures		
7000	Generic RemoveDestination failure	Details of why the Destination cannot be removed
7010	Destination in use	Details of the Task(s) referencing the Destination if possible
RemoveAllDestinations failures		
8000	Generic RemoveAllDestinations failure	Details of why all Destinations cannot be removed
8010	Destinations in use	Details of which Destinations are in use, and (if possible) by which Tasks
8020	RemoveAllDestinations not enabled	None
Generic Object failures		
8500	Generic CreateObject failure	Details of why the Generic Object cannot be created
8510	Generic ModifyObject failure	Details of why the Generic Object cannot be modified
8530	Generic DeleteObject failure	Details of why the Generic Object cannot be deleted
8540	Generic DeleteAllObjects failure	Details of why all Generic Objects cannot be deleted
8550	DeleteAllObjects not enabled	None
Status/fault codes		
9000	Error cleared	Nature of the error which has now cleared
9010	Generic warning	Details of the warning
9020	Generic non-terminating fault	Details of the fault
9030	Terminating fault	Details of the fault
9040	Request actioned	X1TransactionID of the request now actioned
Issue codes		
10000	Database cleared	None

Implementers shall use the most specific error code available.

Table 46a: ExtensionInformation

Field	Description	Format	M/C/O
ExtensionSpecification	Value indicating the specification using the extension	One of the following: "TS133128"	M
ExtendedRequestMessageType	Free text field giving the name of the extended type of Request Message that the Error Response message is a response to, as defined in the related specification	UTF-8 string	M

6.8   Message definitions: managing general objects

6.8.1   CreateObject

6.8.1.1   Summary

DIRECTION: ADMF to NE.

USAGE: Used by the ADMF to add a new Generic Object to the NE.

Table 47: CreateObjectRequest

Field	Description	Format	M/C/O
createObject	Contains a structure derived from the GenericObject.XSD type.	See clause 6.8.1.2	M

Table 48: CreateObjectResponse

Field	Description	Format	M/C/O
OK or Error	The general errors in clause 6.7 apply.	See clause 6.7	M

Generic Objects provide a means of storing additional information at the NE beyond that described by Tasks or Destinations. If the NE already contains a Generic Object with the same objectID as the one supplied in the CreateObjectRequest, the NE shall reject the request with an appropriate error response. If the NE cannot store the supplied record e.g. because it does not support the supplied object type, it shall reject the CreateObjectRequest with an appropriate error response.

6.8.1.2   Generic Object Structure

All Generic Objects shall be descended from the abstract X1Object defined in the schema of the present document. The X1Object definition contains the following fields.

Table 49: X1Object

Field	Description	Format	M/C/O
objectID	Shall uniquely identify the Generic Object at the NE.	GenericObjectID (see clause 6.8.1.3)	M

6.8.1.3   GenericObjectID

A GenericObjectID uniquely identifies a given Generic Object. Derived Generic Object types may introduce further identifier fields, but the GenericObjectID shall be unique for that object at the NE, and shall be the identifier used in relevant Generic Object messages (see clause 6.8).

The GenericObjectID shall be given as a UUID.

6.8.2   ModifyObject

6.8.2.1   Summary

DIRECTION: ADMF to NE.

USAGE: Used by the ADMF to modify an existing Generic Object on the NE. All the details for the object shall be given (i.e. the modified details and the information that is unchanged) to totally replace the previous object details.

Table 50: ModifyObjectRequest

Field	Description	Format	M/C/O
modifyObject	Contains a structure derived from the GenericObject XSD type.	See clause 6.8.1.2	M

Table 51: ModifyObjectResponse

Field	Description	Format	M/C/O
OK or Error	The general errors in clause 6.7 apply.	See clause 6.7	M

Depending on the NE implementation, it may not be possible to modify some or all of the object details. If the NE cannot modify one or more of the elements in the modifyObject structure, it shall reject the entire ModifyObjectRequest with an appropriate error response.

6.8.3   DeleteObject

6.8.3.1   Summary

DIRECTION: ADMF to NE.

USAGE: Used by the ADMF to remove a Generic Object from the NE.

Table 52: DeleteObjectRequest

Field	Description	Format	M/C/O
objectID	Unique identifier for the object.	See clause 6.8.1.3	M

Table 53: DeleteObjectResponse

Field	Description	Format	M/C/O
OK or Error	The general errors in clause 6.7 apply.	See clause 6.7	M

6.8.4   GetObject

6.8.4.1   Summary

DIRECTION: ADMF to NE.

USAGE: Used by the ADMF to retrieve details of a particular Generic Object from the NE.

Table 54: GetObjectRequest

Field	Description	Format	M/C/O
objectID	Unique identifier for the object.	See clause 6.8.1.3	M

Table 55: GetObjectResponse

Field	Description	Format	M/C/O
genericObjectResponseDetails	Structure containing the object and its current status.	See table 56	M

Table 56: GenericObjectResponseDetails

Field	Description	Format	M/C/O
object	The details of the object identified by the objectID, unless there is an error, in which case see clause 6.7. If no object with the supplied objectID is present, this is an error.	See clause 6.8.1.2	M
status	Types derived from GenericObject may also derive status definitions from the GenericObjectStatus type. In this case, the relevant derived status shall be supplied here, otherwise the field shall be omitted.	Derived type	C

6.8.5   ListObjectsOfType

6.8.5.1   Summary

DIRECTION: ADMF to NE.

USAGE: Used by the ADMF to retrieve all identifiers (objectIDs) of objects stored at the NE that have a particular type.

Table 57: ListObjectsOfTypeRequest

Field	Description	Format	M/C/O
objectType	Shall be set to the name of the specific XSD type required.	URIQualifiedName (as defined in XPath 3.1 [24], definition 117)	M

Table 58: ListObjectsOfTypeResponse

Field	Description	Format	M/C/O
listOfObjects	List of objectIDs corresponding to those objects stored at the NE which are of the type specified in the query.	See clause 6.8.1.3	M

6.8.6   DeleteAllObjects

6.8.6.1   Summary

DIRECTION: ADMF to NE.

USAGE: If enabled, the DeleteAllObjects command shall perform a "DeleteObject" command for all Generic Objects on the NE.

Table 59: DeleteAllObjectsRequest

Field	Description	Format	M/C/O
There shall be no request parameters			

Table 60: DeleteAllObjectsResponse

Field	Description	Format	M/C/O
OK or Error	The general errors in clause 6.7 apply. See below regarding whether " DeleteAllObjects " is enabled; if Disabled then DeleteAllObjects always triggers an error response of type " DeleteAllObjects message is not enabled"	See clause 6.7	M

The DeleteAllObjects request shall be supported if the implementation supports Generic Objects. It should be agreed in advance as to whether the DeleteAllObjects request is enabled or disabled. By default (if there has been no agreement in advance) then DeleteAllObjects is enabled.

7   Transport and Encoding

7.1   Introduction

The present document defines a single profile for transport and encoding of X1 messages.

7.2   Profile A

7.2.1   Encoding

XML encoding shall be used. An XSD schema is provided in archive ts_10322101v011301p0.zip which accompanies the present document. In the event of a discrepancy between the XSD schema and the present document, the present document shall be considered authoritative.

The attached samples provide an informative example for implementations of the present document. The samples do not form part of the normative specification.

The attached tool "validate_examples.py" allows implementers to validate the XSD against the attached examples.

7.2.2   Transport layer

7.2.2.1   HTTPS and HTTP

HTTPS shall be used as per IETF RFC 2818 [12]. The details relating to HTTP are given in this clause and the details relating to TLS are specified in clause 8.2.

In this clause, the term HTTP is used (it is implicit that it is in fact HTTPS i.e. that the HTTP is used over TLS).

7.2.2.2   How HTTP is used

The ADMF and NE shall both run HTTP clients and servers:

- For messages where the ADMF is the requester, the ADMF shall use its HTTP client and the NE shall use its HTTP server.

- For messages where the NE is the requester, the NE shall use its HTTP client and the ADMF shall use its HTTP server.

Details in the request:

- Each "RequestContainer" shall be sent as a HTTP request. It shall be a "POST" message (regardless of which type of X1 request it is) and the message body shall contain the RequestContainer as described in clause 6.

Details in the response:

- Each "ResponseContainer" message shall be sent as a HTTP response.

- The response shall indicate HTTP level errors within the range of HTTP error codes. If the HTTP level transaction is successful, then the response shall be a 200 OK message, with the ResponseContainer contained within the message body.

- HTTP error codes shall only be used to indicate HTTP-level errors, and shall not be used to indicate errors with the X1 responses themselves. X1-level errors shall be indicated by correct use of the appropriate X1 ErrorResponse, encoded and returned as a HTTP 200 OK response.

7.2.2.3   Profile

The following profile shall be used:

HTTP version 1.1 or HTTP/2 shall be used. ADMF implementations shall support both.

Where used, HTTP version 1.1 shall be used as per IETF RFC 7230 [13] and related specifications.

NOTE: HTTP/1.1 defaults to the use of "persistent connections" (see IETF RFC 7230 [13], section 6.3). Implementers are encouraged to support the use of persistent connections.

Where used, HTTP/2 shall be used as per IETF RFC 7540 [21] and related specifications.

HTTP/1.1 Pipelining shall not be used.

A Requester may issue multiple HTTP requests in parallel over multiple HTTP connections or multiplexed HTTP/2 requests. However, such implementations should be aware that there is no guarantee of the order in which these requests are processed by the Responder. If such ordering is important to the Requester, it is responsible for ensuring the requests are sent out in the correct order, and for waiting for the response to each request before issuing the next one. Transfer Coding shall not be applied to the HTTP Request or Response (see IETF RFC 7230 [13], section 4).

By default, port 443 shall be used. If this is already in use, then the NE and ADMF shall be able to be configured with a port number, which shall be agreed prior to use of the standard.

By default, the ADMF shall send the HTTP requests with the path set to "/X1/NE" and the NE shall send the HTTP requests with the path set to "/X1/ADMF". An exception to the default shall only be made with strict agreement between NE and ADMF; however, implementers shall ensure that an X1 implementation can be configured with a different path if required.

8   Security

8.1   Overview

This clause details security measures to be implemented for the X1 interface. Other security aspects related to the NE (e.g. secure storage of information, access control) are out of scope of the present document.

8.2   Transport Security

8.2.1   Summary

TLS shall be used which provides authentication and authorization, integrity and confidentiality as well as replay protection between the TLS endpoints.

8.2.2   Profile

TLS shall be followed, using at least version 1.2 as defined in IETF RFC 5246 [14], supporting the recommendations given in IETF RFC 7525 [16].

New implementations should support TLS 1.3 as defined in IETF RFC 8446 [20].

8.2.3   Key generation, deployment and storage

Apart from requirements given in clauses 8.2.1, 8.2.2 and 8.2.4, aspects concerning the generation, distribution, storage and revocation of key material and certificates are out of scope of the present document. Implementations are encouraged to support best practice e.g. the guidance given in OWASP TLS Cheat Sheet [i.1], section 2.6.

> NOTE: It is assumed that the NE and ADMF are in a physically secure environment. For future uses (e.g. NFV), then this assumption would no longer be valid. Further details would then need to be added about the security of storage of key or certificate material e.g. TPM, Secure enclaves. See ETSI TR 103 308 [i.2], ETSI GS NFV-SEC 009 [i.3] and ETSI GS NFV-SEC 012 [i.4].

8.2.4   Authentication

Implementations shall perform mutual authentication using X.509 certificates following IETF RFC 6125 [17]. Implementations shall ensure that it is configurable which certificates are used.

X1 implementations shall check that the UID relative distinguished name (OID 0.9.2342.19200300.100.1.1) of the Subject field in the certificate (see IETF RFC 4519 [18], section 2.39) provided matches the Sender or Receiver ID (whichever is provided by the other party in the communication). If a Responder receives an X1 message where these values do not match, it shall respond with an X1 error message indicating that the Requester is not authorized. If the Requester receives a response where these values do not match, then it shall disregard the response and log an appropriate error message.

8.3   Additional security measures (beyond transport layer)

It will be important to follow general security best practice (e.g. use of firewalls and/or access lists to prevent denial-of-service attacks). This is out of scope of the present document. However, implementers are specifically encouraged to follow XML best practices outlined in the OWASP XML Security Cheat Sheet [i.5].

The present document does not recommend that message-layer encryption or message-level message authentication codes are used in addition to the provisions in this clause. Of course, there may be threat models in which additional encryption may be thought to be useful. The present document does not forbid adding message-layer encryption e.g. by encrypting the whole of the payloads of the request and response messages. The details of the changes needed to do this are outside the scope of the present document.

Annex A (normative):
Requirements

A.1   Basic requirements

A.1.1   Existing standards

The interface should use already existing mechanisms and standards if possible:

R1)   **Future proof:** Changes can be made and new features can be added. A version structure will allow for co-existence of different versions.

R2)   **Open structure:** The interface will have an open structure that will allow for extensions. Though it should be as strict as possible to make implementations as interoperable as possible. Extensions should not have any negative impact on security and other requirements.

R3)   **Security:** Authentication, integrity protection and confidentiality shall be supported from end to end.

R4)   **Authenticity:** The authenticity of a message can be checked in a standalone environment (e.g. no connection to an online server needed, root certificate can be enough).

R5)   **Legal framework:** The present document contains a technical specification which is independent of national legislation. It does not supersede national legislation or approved practices.

R6)   **Direct delivery:** Some network elements support direct delivery of IRI and CC without any additional mediation and delivery function. The interface should also support administration of these network elements.

R7)   **Core functionality:** It shall be possible to provision (create, modify and delete) interceptions including all necessary parameters (e.g. CC/IRI-destination) on network nodes. It shall be possible to retrieve details of a single or all interceptions provisioned on a network node.

R8)   **Administration:** It shall be possible to administrate LI relevant configuration on network nodes (e.g. update of security certificates).

A.2   Protocol & Architecture requirements

The following protocol and architecture requirements are listed:

R9)   **Node Scope:** The X1 architecture and protocol shall support administration of all nodes involved in capture and control of target intercept traffic including intercept nodes and mediation and delivery functions. This shall include both on-switch and off switch probe scenarios.

R10)   **Basic functionality:** The basic message exchange protocol shall be able to carry both generic LI parameters (e.g. those obtained from X1 E-warrant interface) and Interception Node manufacturer specific parameters.

R11)   **Extensible:** The basic message exchange protocol shall allow limited extensibility to support parameter not currently supported by the base protocol. This extensibility shall be limited to encourage future extension of the standardized basic functionality in future versions of the X1 standard.

R12)   **Flexibility:** The X1 architecture and message exchange technique shall be flexible to allow implementation in both existing and future national and international operator network architectures. As a minimum it shall be compatible with 3GPP, TISPAN/NTECH, NFV SEC, ETSI TC LI, ANSI and other international network architecture and handover standards.

R13)   **One-to-many:** The architecture and message protocol shall support both one-to-one and one-to-many LI end point configurations (i.e. it shall be possible to provision hundreds of end points simultaneously and efficiently).

R14) Backwards compatibility: The X1 architecture and protocol shall be backwards compatible with existing LI devices where possible. Specifically the standardized X1 shall not place significantly more performance or load impacts than existing proprietary approaches on LI nodes.

There is no specific requirement to retro-fit this X1 standards onto existing IP or legacy circuit switched nodes, although the standards does not prohibit such retrofitting where practical. Parallel running of X1 and legacy or proprietary interfaces shall be supported where practical. The X1 architecture shall permit different versions of X1 to be running on different components and (as far as is practical) the functionality from the older version shall still continue to work (though features introduced in the new versions shall cause errors to be sent).

R15) Lightweight: Many LI devices (e.g. Switches/Routers) currently use lightweight protocols such as SNMP, and have limited processing power and/or limited application layer intelligence. The protocol shall be designed to support such lightweight devices.

R16) Permanent and dynamic connections: The X1 architecture and message exchange technique shall support both permanent connection and dynamic link/connection scenarios.

R17) Direct delivery: Support situation where interception is delivered direct to LEMF without further CSP mediation. No need to explicitly draw this out but do allow enough information over X1 to support this situation.

R18) Delay: The X1 architecture and message exchange technique shall by design not introduce undue delay compared with existing proprietary X1 implementations.

R19) Dynamic Triggering and HI1: The X1 architecture and message exchange technique shall be compatible and interoperable with both ETSI TC LI HI1 and Dynamic Triggering standards.

A.3   Security requirements

R20) Authentication: The X1 architecture and message exchange technique shall provide both authentication of physical end points and authentication of the software application receiving the message.

NOTE: Requirement is limited to authenticating the LI function identity and not authenticating the software version or integrity.

R21) Authorization: The X1 architecture and message exchange technique shall provide both authorization of physical end points and authorization of the software application receiving the message.

R22) Accounting and audit: The X1 architecture and message exchange technique shall include sufficient information to enable Accounting & Auditing functions in the ADMF and NE.

R23) Integrity protection: The X1 message exchange technique shall provide integrity protection for all messages exchanged between nodes in the X1 architecture. Use of Integrity protection shall be mandatory.

R24) Confidentiality protection: The X1 message exchange technique shall provide confidentiality protection for all messages exchanged between nodes in the X1 architecture.

R25) Replay protection: The X1 message exchange technique shall provide replay protection for all messages exchanged between nodes in the X1 architecture.

R26) Standalone interface: The X1 architecture and message exchange technique shall be designed as a standalone physically dedicated LI interface. The design and selection of the protocol shall where possible ensure vulnerabilities in non-LI interfaces on the same node shall not impact LI interfaces and security.

R27) Hardened Protocol: The X1 message exchange technique shall use a harden protocol containing minimal options or extensions which are not specifically required by X1.

R28) Minimum Security Level: The X1 architecture and message exchange techniques shall provide a minimum level of security (including cypher suites and key length), which shall be supported by all nodes. At least two algorithms shall be specified. The protocol and algorithms shall be resistant to bid down attack.

R29) **Underlying Infrastructure Trust:** The X1 architecture and message exchange techniques shall assume by default that the underlying network communication links and infrastructure are untrusted.

R30) **Firewall and NAT Transversal:** The X1 message exchange technique shall be compatible with existing operator firewall and NAT transversal architectures. The message exchange technique shall not require unrestricted opening of common ports (e.g. port 80 or 21). The message exchange technique shall not prohibit the development of future X1 aware firewall filtering to provide rejection of malicious X1 message at operator security gateways.

R31) **Certificate and Key Management:** The X1 architecture and message exchange techniques shall include (where applicable) Certificate and Key Management mechanisms. In addition mechanisms for Certificate/Key revocation shall be provided.

R32) **Single Node Compromise:** The X1 architecture and message exchange techniques shall ensure that a vulnerability or weak implementation in one node does not adversely affect other nodes. Specifically it shall not be possible to attack one interception node by using recovered plan text or other security parameters from a vulnerable one.

R33) **Node Administration:** The X1 architecture and message exchange techniques shall ensure by design that within node implementations, non-LI super-users can be prevented from making LI related parameters changes without authority from and knowledge of the LI administrator.

R34) **Encryption of target information:** It shall be possible to use encrypted target information only by use of encrypted targets and encryption keys. In case of encrypted information it shall be possible to change encrypted target information and encryption keys periodically without interruption of any active interception.

A.4   Other requirements

A.4.1   Performance statistics (for further study)

Performance requirements include:

- In general or per LI measure.

- Activity: Amount of intercepted traffic? Maximum and average bandwidth? Minutes of intercepted voice? Count of intercepted messages? Time of last activity?

- Maximum number of parallel interceptions (e.g. in busy hours).

- Maximum number of parallel intercepted accounts/connections with same target identifier (e.g. in case of IMEI duplicates).

The performance requirements are derived from measures of the amount and rate of Lawful Interception. Clearly this will vary but some guidelines are as follows:

- Considerations of the bandwidth of intercepted traffic are in general not relevant to X1 (except perhaps for a NE to report that bandwidth is exceeding certain parameters).

- Number of targets on cover at any given time:

  - This number is usually very small compared to the total number of users and for the purposes of the present document will be considered as tens or hundreds at most.

- Are there situations where a single target on cover causes a lot of X1 messages. Consider the following ways this could happen:

  - Can a single target cause a large number of target identifiers to be tasked (consider roaming)?

  - Can one have a large number of HI1 messages for each target identifier (frequent changing of parameters)?

Annex B (normative):
Use of extensions

B.1   Overview

The present document defines a number of extension points, including in the TaskDetails structure (see clause 6.2.1.2), and TargetIdentifier format (see table 5). This clause defines how extensions are to be used in table 4 and table 5.

B.2   Extension definitions

Where a feature or information element already exists in the present document, it shall be used in preference to any extended field. Extensions shall not be drafted as an alternative or re-formatting of functionality or information that already exists within the present document.

An extension shall be a structure (e.g. a complexType in XSD) defined in a separate schema, and shall contain at a minimum the following elements.

Table B.1: Extension fields

Field	Description	Format	M/C/O
Owner	Human-readable indication of the entity responsible for the definition and maintenance of the extension	UTF-8 string	M

The extensions shall be defined in a namespace belonging to the entity responsible for drafting and maintaining the extension. It shall not be defined in the namespace of the present document.

Annex C (normative):
Using Task Object at Mediation and Delivery Functions

C.1 Overview

An ADMF may use X1 messages to provision a mediation and delivery function instead of a point of interception, following the deployment model given in clause 4.1.4. This annex describes how the usage and meaning of the messages defined in clause 6 differ when used for this purpose. Unless otherwise specified, the messages are used as for any other NE.

C.2 TaskDetails

C.2.1 General

The TaskDetails structure used in the ActivateTask and ModifyTask messages are used as for an NE with the differences described in the following clauses.

C.2.2 MediationDetails structure

The MediationDetails structure provides additional details for a Task, specific to Mediation and Delivery Functions. Multiple instances of the MediationDetails structure may be used to indicate that multiple LIIDs are associated with the task.

When a ModifyTask message is received by the MDF from the ADMF, the MDF shall, upon successful processing and execution of the ModifyTask message, ensure that:

1)   only the LIIDs included in the ModifyTask message (via a MediationDetails structure) remain active; and

2)   any LIIDs that were associated with the task identified in the ModifyTask message, but were not identified in the ModifyTask message, shall be deactivated (i.e. those intercepts shall cease).

To clarify the above, suppose that TaskID A had LIID 4 and LIID 5 associated with it and interception was active on both LIID 4 and LIID 5. If a ModifyTask message is received and successfully processed by the MDF with a single MediationDetails structure that includes LIID 4, then the interception on LIID 4 will remain active while the interception on LIID 5 will cease.

Table C.1: Mediation Details structure

Field	Description	Format	M/C/O
LIID	Lawful Interception Identifier associated with the Task	LIID as defined in ETSI TS 103 280 [4]	M
DeliveryType	Statement of whether to deliver HI2 and/or HI3 for this LIID	Enumerated value - one of "HI2Only", "HI3Only" or "HI2andHI3"	M
StartTime	Start time associated with the activation of interception or mediation for the Task (which may be in the future)	Timestamp	O
EndTime	End time associated with the deactivation of interception or mediation for the Task	Timestamp	O
ListOfDIDs	Details of where to send the intercepted traffic for this LIID. Shall be included if deviation from the taskDetails ListofDIDs is necessary. If included, the details shall be used instead of any delivery destinations specified in the ListOfDIDs field in the TaskDetails structure	List of Destination Identifiers (DID) referencing the desired delivery destination records	C

Field	Description	Format	M/C/O
MediationDetailsExtension	One or more extension placeholders; each may be populated by a list of elements defined by external specifications	See annex B	O
ServiceScopingOptions	Shall be included to Identify the service(s) and associated service-related delivery settings for this LIID if there is no default setting in the MDF for ServiceScoping or to override the default setting in the MDF for ServiceScoping. May include more than one instance of this parameter to allow for different combinations of sub-parameters associated with a single LIID. This parameter is defined in more detail in table C.2	Sequence containing - one or more instances of the ServiceScoping options listed in table C.2	C

Table C.2: Service Scoping structure

Field	Description	Format	M/C/O
ServiceType	Shall be included to Identify the service(s) to be reported for this LIID per the description beneath the table. The values given in this field indicate the services to which the other options in the Service Scoping structure shall apply	One or more of the following enumerated values: • "voice" • "data" • "messaging" • "pushToTalk" • "LALS" • "RCS""	C
LocationType	Shall be included to Identify whether and under what conditions to deliver location information for this LIID per the description beneath the table	Choice of: • "doNotReport" • One or more of the following enumerated values: – "reportBeginningAndEnd" – "reportUponChange" – "reportLALS"	C
SuspendOnOutboundInternationalRoaming	Shall be included to Identify whether to suspend interception or not (i.e. continue interception) if the target undergoes outbound international roaming and per the description beneath the table	Boolean	C
ReportPostDialledDigits	Shall be included to identify whether reportPostdialleddigits are to be reported for this LIID for the ServiceType of "voice" per the description beneath the table. Not applicable to other service types	Boolean	C

For ServiceType, LocationType, SuspendOnOutboundInternationalRoaming, or ReportPostDialledDigits, If there is no default setting in the MDF or to override the default setting in the MDF, the corresponding parameter shall be included.

NOTE: RCS is defined as Rich Communication Services (see GSMA RCC.07 [i.6]).

Annex D (normative):
Hashed Identifiers

D.1 Overview

Hashed identifiers provide an alternative to providing plain-text target identifiers over X1. This is intended to provide a measure of additional security against disclosure of such target identifiers. However, it should be noted that this technique does not provide protection against:

- An attacker in possession of hash information from verifying whether a specific given identifier matches a given hash or salt.

- An attacker in possession of complete hash information (including salt) from recovering identifiers that have a small set of possible values (e.g. MSISDN numbers in a particular country) by brute force attack.

Instead, this technique is intended to provide a simple extra layer of protection against e.g. accidental disclosure via a user interface.

D.2 Hashed Identifier Usage

D.2.1 Overview

An ADMF wishing to provision an NE with a hashed identifier uses the following procedure:

1) The ADMF populates a Hash Context object with the operator's chosen hash algorithm identifier and ra random salt value (see clause D.2.2).

2) The ADMF issues a CreateObject request containing the Hash Context object to the NE (see clause 6.8.1).

3) The ADMF calculates the hash digest of the required plain-text identifier using the details from the Hash Context (see clause D.2.3.2).

4) The ADMF populates a HashedIdentifier structure with the digest, along with an indication of the target identifier type and the identifier of the Hash Context object containing the salt (see clause D.2.3).

5) The ADMF issues an ActivateTask request containing the HashedIdentifier to the NE (see clause 6.2.1).

The NE can now inspect each candidate identity and create a hash digest using the information in the Hash Context. If the digest matches the one in the HashedIdentifier structure, the NE can consider the target identity to have matched.

Hashed Identifiers may only be used for target identifier types which derive from simple types such as xs:token, and which specify a single unambiguous value as a target identifier. Hashed Identifiers may not be used for:

- target identifier types which are complex types due to potential ambiguities in forming a canonical binary representation (see clause D.2.3);

- target identifier types which do not describe a single unambiguous value (such as tcpPortRange) since it is impossible to determine whether a given identifier matches the target identifier by comparing hashes.

However, a given Task may contain both hashed and non-hashed target identifiers (e.g. a hashed IPv4 address along with a plain-text tcpPortRange) in its targetIdentifiers list.

D.2.2   Hash Context

A Hash Context is derived from a Generic Object (see clause 6.8.1.1) and consists of the following elements:

Table D.1: Hash Context structure

Field	Description	Format	M/C/O
hashAlgorithm	Gives the object identifier of the hash context containing the relevant configuration details used to calculate the hash digest.	Hash name string matching one of those defined in the IANA Named Information Hash Algorithm Registry [25].	M
salt	Salt to be used when calculating the hash digest value (see clause D.2.3). Shall be at least 8 octets long.	XML hexbinary representation of the salt value.	M

The choice of hash algorithm is made by the operator and enforced by the ADMF. ADMFs and NEs supporting hashed identifiers shall support the use of the following hash algorithms:

- sha-256 with 256-bit value length as defined in IETF RFC 6920 [25].

- sha-512 with 512-bit value length as defined in IETF RFC 6920 [25].

- sha3-512 with 512-bit value length as defined in FIPS PUB 202 [26].

Additional algorithms may be supported in both the ADMF and NE.

If the ADMF requests the creation of a Hash Context object with an unsupported hash algorithm or an insufficiently long salt, the NE shall reject the request with an appropriate error.

A Hash Context and its associated salt may be used by multiple HashedIdentifier instances (see clause D.2.3) to reduce the processing burden at the NE, at the cost of reducing the number of salts that an attacker would need to deal with if attempting to exhaustively search for the original target identifier.

D.2.3   HashedIdentifier

D.2.3.1   Structure

A HashedIdentifier consists of the following elements.

Table D.2: HashedIdentifier structure

Field	Description	Format	M/C/O
hashContextID	Gives the object identifier of the hash context containing the relevant configuration details used to calculate the hash digest.	X1ObjectID (see clause D.2.2)	M
targetIdentityType	Name of the equivalent plain-text target identity element from the TargetIdentifier type. Only simple types are supported.	String	M
hashDigest	Digest of the target identifier and salt.	XML hexbinary representation of the binary digest	M

D.2.3.2   Hashing procedure

It is essential that both the ADMF and NE calculate the hashDigest value in the same way. The hashDigest value shall be calculated according to the following procedure at both NE and ADMF:

1) Ensure that any plain-text target identity used to calculate a hashDigest is first correctly normalized into the format defined by the relevant TargetIdentifier format (see table 5).

2)   For values where value comparisons are case-insensitive, transform the plain-text identity to lower-case. In cases where parts of the value are case-insensitive and others are not (e.g. SIP URI) then only the case-insensitive parts shall be lower-cased.

3)   Obtain a binary representation of the plain-text target identity:

-    For simple types derived from xs:token or xs:string, the binary representation shall be the octets giving the UTF-8 encoding of the plain-text string.

-    For simple types derived from xs:hexbinary, the binary representation shall be the octets represented by the hexbinary notation.

-    For simple types derived from xs:integer and which represent unsigned numbers, the binary representation shall be the octets of the binary representation of that number given in network byte order (i.e. big endian).

4)   Concatenate the octets of the salt value from the associated Hash Context to the end of the binary representation of the identity.

5)   Take the hash of the concatenated result using the hash algorithm identified by the associated Hash Context.

D.3   Worked examples

D.3.1   Worked example 1

D.3.1.1   Initial information

Table D.3: Initial information

Information element	Value
Chosen hashing algorithm	sha-256
Plain-text target identity type	InternationalE164
Plain-text target identity value	"447700900000"

D.3.1.2   Construction of the Hash Context

The ADMF chooses a salt value of 0x4241792fc4d3d097, and allocates a random UUID for an object identifier. The ADMF now has enough information to populate a Hash Context object.

```
<X1Object xsi:type="HashContext">
  <x1ObjectID>e3d62e2b-d211-433d-b0f9-488ed89ba7c0</x1ObjectID>
  <hashAlgorithm>sha-256</hashAlgorithm>
  <salt>4241792fc4d3d097</salt>
</X1Object>
```

The ADMF may now issue a CreateObjectRequest message to the NE with this Hash Object (see clause 6.8.1).

D.3.1.3   Binary representation of the target identity

In this case, the identity is of type InternationalE164. This is a type derived from xs:token, so the binary representation is the bytes of the utf-8 representation of the string.

```
binary_representation = 0x343437373030393030303030
```

D.3.1.4   Concatenation with the salt

The salt is taken from the Hash Context that the ADMF wishes to use, and is appended to the end of the binary representation.

```
concatenated_value = binary_representation || salt
     = 0x343437373030393030303030304241792fc4d3d097
```

D.3.1.5   Calculation of the hash digest

The ADMF can now calculate the hash digest of the concatenated value, using the hash algorithm identified in the Hash Context object (in this case, sha-256).

```
hashDigest = sha-256(concatenated_value)
    = sha-256(0x343437373030393030303030304241792fc4d3d097)
= 0xddbe522009b5b32f1b84c82c06dedc0d24ba373d4ae244790fd071076b4536c0
```

D.3.1.6   Construction of the HashedIdentifier

The ADMF now has all the information it needs to create a HashedIdentifier target identity for use in an ActivateTask message towards the NE.

```
<hashedIdentifier>
  <hashContextId>e3d62e2b-d211-433d-b0f9-488ed89ba7c0</hashContextId>
  <targetIdentityType>InternationalE164</targetIdentityType>
  <hashDigest>ddbe522009b5b32f1b84c82c06dedc0d24ba373d4ae244790fd071076b4536c0</hashDigest>
</hashedIdentifier>
```

Annex E (normative):
Destination Sets

E.1   Overview

When intercepted traffic is to be delivered by the NE to a Destination which belongs to a group of related Destinations DIDs can be grouped together under a single DSID (see clause 5.1.3).

When Destination Sets are used each Task is associated with one or more Destination Sets. Prior to associating a Task with a given DSID, it is required that a Destination Set with the DSID has already been created as described in clause E.2 but there is no requirement that a connection has been successfully established for that DSID.

Checks regarding availability and status of downstream delivery of information are outside the scope of the present document.

E.2   Destination Set Usage

E.2.1   Overview

All Generic Object Methods are applicable to DestinationSetDetails Objects.

An ADMF wishing to use a DSID within a provisioning request towards an NE uses the following procedure:

- The ADMF populates a DestinationSetDetails object with the identifiers and values as described in clause E.2.2.

- The ADMF issues a CreateObject request, containing the DestinationSetDetails object, to the NE (see clause 6.8.1).

- The ADMF issues an ActivateTask request containing the DestinationSetDetails Generic Object ID(s), also referred to as the DSID(s), to be used within the ListofDIDs field (see clause 6.2.1).

It is required that a Destination with the DID has already been created (as described in clause 6.3) before it can be included within a Destination Set Details object, although there is no requirement that a connection has been successfully established for that DID.

Depending on the NE implementation, it may not be possible to modify some elements or all of a Destination Set details while the Destination Set is in use. If the NE cannot modify one or more of the elements in the ModifyObject request, it shall reject the entire ModifyObject request with an appropriate error response.

The length of time an NE requires to make the changes requested in the ModifyObject request message is an implementation detail, but the expectation is that changes are made without undue delay.

A Destination Set may only be removed if it is not referenced by any Tasks. An NE shall respond with an appropriate error if the ADMF attempts to remove a Destination Set that is referenced by a Task.

E.2.2 DestinationSetDetails Object

A DestinationSetDetails object consists of the following elements.

Table E.1: DestinationSetDetails structure

Field	Description	Format	M/C/O
FriendlyName	A human-readable name associated with the Destination Set.	Free-text string.	O
ListOfAssociatedDIDs	One or more ListOfIncludedDID.	As defined in Table E.2	M
DestinationSetDetailsExtensions	One or more extension placeholders; each may be populated by a list of elements defined by external specifications.	See annex B.	O
DestinationSetType	Shall be included to identify how IRI and/or CC should be distributed across the DIDs within the Destination Set.	Enumerated value - one of "Redundant" or "Duplicate".	M

The ListOfIncludedDID structure is defined as follows.

Table E.2: ListOfAssociatedDIDs

Field	Description	Format	M/C/O
DID	See clause 5.1.	See clause 5.1.	M
Preference	An integer representing the DIDs preference of use within the Destination Set. Where available for use by the NE, DID or DIDs with the lowest preference values will be used in prior to those with higher preference values.	Integer.	C

Where the DestinationSetType included within the DestinationSetDetails is "Redundant" the POI will use the specified DIDs as a set of redundant end points, it is mandatory for the "Preference" to be defined for each DID within a Destination Set where the DestinationSetType is "Redundant". Preference defines the DIDs order of use with the smallest integer indicating the most preferred DID(s). Should the most preferred DID(s) become unavailable the next preferred and available DID(s) shall be used.

It is an implementation decision for the NE to determine whether to duplicate traffic if two or more DIDs with the same Preference value are referenced within the same DestinationSetDetails object.

Where the DestinationSetType included within the DestinationSetDetails is "Duplicate", the NE will send copies of intercepted traffic to all DIDs within the set, preference shall not to be included where the DestintionSetDetails is of type "Duplicate".

Annex F (informative): Change history

TC LI Approval Date	Version	Remarks
October 2017	1.1.1	First publication XSD schema is provided in TS_103_221_01_v010101.xsd contained in archive ts_10322101v010101p0.zip.
February 2018	1.2.1	Included Change Request: TS103221-1CR001r1 (cat F) Warning and Faults Reporting This CR was approved by TC LI#47 (5-7 February 2018, New Delhi) No changes in XML Schema.
June 2018	1.3.1	Included Change Request: TS103221-1CR002r2 (cat F) X1 response/request lifecycle This CR was approved by TC LI#48 (26-28 June 2018, Bergen) No changes in XML Schema.
February 2019	1.4.1	Included Change Request: TS103221-1CR003r3 (cat B) Support for 5G This CR was approved by TC LI#50 (5-7 February 2019, Dubai) XSD schema is provided in TS_103_221_01_v010401.xsd contained in archive ts_10322101v010401p0.zip.
July 2019	1.5.1	Included Change Requests: CR004r1 (cat F) Permitting multiple extensions in X1 CR005r6 (cat C) Mediation Details Update CR006r1 (cat F) Task Details Update CR007r1 (cat F) Clarify XID to LIID Relationship CR008r1 (cat F) DeliveryAddress Updates CR009r1 (cat F) TaskStatus Updates CR010 (cat C) Corrections after implementation These CRs were approved by TC LI#51 (11-13 June 2019, Texel) XSD schema is provided in TS_103_221_01_v010501.xsd contained in archive ts_10322101v010501p0.zip.
October 2019	1.6.1	Included Change Requests: CR012 (cat B) Use of HTTP/2 CR013 (cat B) Addition of Product ID CR014 (cat C) Making the requirements annex informative CR015 (cat B) Update for TLS 1.3 CR016 (cat D) Alignment to 3GPP terminology These CRs were approved by TC LI#52 (15-17 October 2019, Turin) XSD schema is provided in TS_103_221_01_v010601.xsd contained in archive ts_10322101v010601p0.zip.
July 2020	1.7.1	Included Change Request: CR017 (cat F) Clarifications on use of delayed Acknowledgements for Destinations This CR was approved by TC LI#54-e (17-25 June 2020) No changes in XML Schema. XSD schema is provided in TS_103_221_01_v010601.xsd contained in archive ts_10322101v010701p0.zip.

Table: Status of the present document: ETSI TS 103 221-1 Internal Network Interfaces; Part 1: X1

Status of the present document: ETSI TS 103 221-1 **Internal Network Interfaces; Part 1: X1**		
TC LI Approval Date	**Version**	**Remarks**
February 2021	1.8.1	Included Change Requests: CR019r2 (cat F) MessageTimestamp clarification CR020r2 (cat F) X1 HTTP path clarification CR021r2 (cat F) Updating the version field CR022r3 (cat F) Clarifying UID RDN These CRs were approved by TC LI#56-e (15-19 February 2021) XSD schema is provided in TS_103_221_01_v010801.xsd contained in archive ts_10322101v010801p0.zip.
June 2021	1.9.1	Included Change Requests: CR025r1 (cat B) Addition of InternationalizedEmailAddress type CR026r1 (cat B) New services in service scope structure of table C.2 These CRs were approved by TC LI#57-e (21-25 June 2021) XSD schema is provided in TS_103_221_01_v010901.xsd contained in archive ts_10322101v010901p0.zip.
October 2021	1.10.1	Included Change Requests: CR027r2 (cat B) Generic object mechanism CR028r2 (cat B) Hashed Identifiers CR029r2 (cat B) Addition of EUI-64 CR030r2 (cat B) Addition of Service Type to Task Details These CRs were approved by TC LI#58-e (18-22 October 2021) XSD schema is provided in TS_103_221_01_v011001.xsd contained in archive ts_10322101v011001p0.zip.
February 2022	1.11.1	Included Change Requests: CR031r2 (cat B) Clarifications to Generic Object sections CR032r3 (cat C) Destination Identifier Set Object These CRs were approved by TC LI#59-e (14-19 February 2022) XSD schema is provided in TS_103_221_01_v011101.xsd contained in archive ts_10322101v011101p0.zip.
July 2022	1.12.1	Included Change Requests: CR034r4 (cat C) Adding NE alert when database is cleared CR035r1 (cat C) ErrorResponse for X1 RequestMessageType extensions These CRs were approved by TC LI#60 (28-30 June 2022, Paris) XSD schema is provided in TS_103_221_01_v011201.xsd contained in archive ts_10322101v011201p0.zip.
November 2022	1.13.1	Included Change Requests: CR036r2 (cat C) Enhancing getting details from NE CR037r2 (cat B) ServiceScopingOptions Alignment These CRs were approved by TC LI#61 (22-24 September 2022, Malmö) XSD schema is provided in TS_103_221_01_v011301.xsd contained in archive ts_10322101v011301p0.zip which accompanies the present document.

History

Document history		
V1.1.1	October 2017	Publication
V1.2.1	March 2018	Publication
V1.3.1	September 2018	Publication
V1.4.1	April 2019	Publication
V1.5.1	July 2019	Publication
V1.6.1	December 2019	Publication
V1.7.1	August 2020	Publication
V1.8.1	April 2021	Publication
V1.9.1	July 2021	Publication
V1.10.1	December 2021	Publication
V1.11.1	March 2022	Publication
V1.12.1	August 2022	Publication
V1.13.1	December 2022	Publication