**3GPP SA3#79-LI-e-b *S3i200705***

**eMeeting, 10-12 November 2020**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.1* | | | | | | | | |
| **CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  |  | **CR** |  | **rev** | **1** | **Current version:** |  |  |
|  | | | | | | | | |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* | | | | | | | | |
|  | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME |  | Radio Access Network |  | Core Network | **X** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** | Identifier Association | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | SA3-LI (BT Plc, Public Safety Canada, PIDS, OTD, Ministère de l'Economie et des Finances, BfV, LKA Niedersachen, NTAC, OFCOM(CH), BKA, AGD, Telefónica S.A., Rogers Communications Canada) | | | | | | | | | |
| ***Source to TSG:*** | SA3 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | LI16 | | | | |  | ***Date:*** | | | 2020/11/10 |
|  |  | | | |  | |  | | |  |
| ***Category:*** | B |  | | | | | ***Release:*** | | | Release 16 |
|  | *Use one of the following categories:* ***F*** *(correction)* ***A*** *(mirror corresponding to a change in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)*  Detailed explanations of the above categories can be found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | | | | | | | | *Use one of the following releases: Rel-8 (Release 8) Rel-9 (Release 9) Rel-10 (Release 10) Rel-11 (Release 11) … Rel-15 (Release 15) Rel-16 (Release 16) Rel-17 (Release 17) Rel-18 (Release 18)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | Features such as SUPI privacy cause challenges for LEAs in being able correctly identify and target individual UEs for LI purposes. It also presents challenges for CSPs to be able to meet LI requirements in terms of removing encryption or transcoding applied by the CSP. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | This CR provides a set of real-time identifier assocation reporting and querying capabilities which allow LEAs to target individual UEs when SUPI privacy or other temporary identifier mechanisms are in use by the AMF. The CR allows privacy mechanism to remain enabled while allowing the CSP to meet LI related regulatory obligations. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | LEAs are not able to obtain necessary permanent to temporary and temporary to permanent identifier assocations for LI purposes. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 2, 3.3, 4.1, 4.2, 4.3, 5.2.7 (New), 5.7 (New), 5.8 (New), 5.9 (New), 6.2.2.1, 6.2.2.2.1, 6.2.2.2.2, 6.2.2.2.5, 6.2.2.2.7 (New), 6.2.2.3, 6.2.2A (New), 6.3.2.2, 6.3.2.2.1 (New), 6.3.2.2.2 (New), 6.3.2.3, 7.6 (New), Annex A, Annex C, Annex E (New), Annex F (New). | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **Y** | **N** |  | | | |  | | |
| ***Other specs*** | |  | **X** | Other core specifications | | | | TS/TR ... CR ... | | |
| ***affected:*** | |  | **X** | Test specifications | | | | TS/TR ... CR ... | | |
| ***(show related CRs)*** | |  | **X** | O&M Specifications | | | | TS/TR ... CR ... | | |
|  | |  | | | | | | | | |
| ***Other comments:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR’s revision history:*** | |  | | | | | | | | |

----------------------START OF CHANGES---------------------

# 2 References

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

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 23.501: "System Architecture for the 5G System".

[3] 3GPP TS 33.126: "Lawful Interception Requirements".

[4] 3GPP TS 23.502: "Procedures for the 5G System; Stage 2".

[5] 3GPP TS 33.127: "Lawful Interception (LI) Architecture and Functions".

[6] ETSI TS 103 120: " Lawful Interception (LI); Interface for warrant information".

[7] ETSI TS 103 221-1: "Lawful Interception (LI); Internal Network Interfaces; Part 1: X1".

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

[9] ETSI TS 102 232-1: "Lawful Interception (LI); Handover Interface and Service-Specific Details (SSD) for IP delivery; Part 1: Handover specification for IP delivery".

[10] ETSI TS 102 232-7: "Lawful Interception (LI); Handover Interface and Service-Specific Details (SSD) for IP delivery; Part 7: Service-specific details for Mobile Services".

[11] 3GPP TS 33.501: "Security Architecture and Procedures for the 5G System".

[12] 3GPP TS 33.108: "3G security; Handover interface for Lawful Interception (LI)".

[13] 3GPP TS 24.501: "Non-Access-Stratum (NAS) protocol for 5G System (5GS)".

[14] 3GPP TS 24.007: "Mobile radio interface signalling layer 3; General Aspects".

[15] 3GPP TS 29.244: "Interface between the Control Plane and the User Plane nodes".

[16] 3GPP TS 29.502: "5G System; Session Management Services; Stage 3".

[17] 3GPP TS 29.571: "5G System; Common Data Types for Service Based Interfaces; Stage 3".

[18] 3GPP TS 23.040: "Technical realization of the Short Message Service (SMS)".

[19] 3GPP TS 23.003: "Numbering, addressing and identification ".

[20] OMA-TS-MLP-V3\_5-20181211-C: "Open Mobile Alliance; Mobile Location Protocol, Candidate Version 3.5", <https://www.openmobilealliance.org/release/MLS/V1_4-20181211-C/OMA-TS-MLP-V3_5-20181211-C.pdf>.

[21] 3GPP TS 29.540: "5G System; SMS Services; Stage 3".

[22] 3GPP TS 29.518: "5G System; Access and Mobility Management Services; Stage 3".

[23] 3GPP TS 38.413: "NG Application Protocol (NGAP)".

[24] 3GPP TS 29.572: "Location Management Services; Stage 3".

[25] 3GPP TS 29.503: "5G System; Unified Data Management Services".

[26] IETF RFC 815: "IP DATAGRAM REASSEMBLY ALGORITHMS".

[27] IETF RFC 2460: "Internet Protocol, Version 6 (IPv6) Specification".

[28] IETF RFC 793: "TRANSMISSION CONTROL PROTOCOL".

[29] IETF RFC 768: "User Datagram Protocol".

[30] IETF RFC 4340: "Datagram Congestion Control Protocol (DCCP)".

[31] IETF RFC 4960: "Stream Control Transmission Protocol".

[32] IANA (www.iana.org): Assigned Internet Protocol Numbers, "Protocol Numbers".

[33] IETF RFC 6437: "IPv6 Flow Label Specification".

[34] IETF RFC 791: "Internet Protocol".

[35] Open Geospatial Consortium OGC 05-010: "URNs of definitions in ogc namespace".

[36] 3GPP TS 33.107: "3G security; Lawful interception architecture and functions".

[37] 3GPP TS 37.340: "Evolved Universal Radio Access (E-UTRA) and NR-Multi-connectivity; Stage 2".

[38] 3GPP TS 36.413: "S1 Application Protocol (S1AP)".

[39] OMA-TS-MMS\_ENC-V1\_3-20110913-A: "Multimedia Messaging Service Encapsulation Protocol".

[40] 3GPP TS 23.140: "Multimedia Messaging Protocol. Functional Description. Stage 2".

[41] 3GPP TS 38.415: "NG-RAN; PDU Session User Plane Protocol".

[XA] 3GPP TS 29.509: "5G System; Authentication Server Services; Stage 3".

----------------------NEXT CHANGES---------------------

## 3.3 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

ADMF LI Administration Function

CC Content of Communication

CSP Communication Service Provider

CUPS Control and User Plane Separation

ICF Identifier Caching Function

IEF Identifier Event Function

IQF Identifier Query Function

IRI Intercept Related Information

LALS Lawful Access Location Services

LEA Law Enforcement Agency

LEMF Law Enforcement Monitoring Facility

LI Lawful Interception

LICF Lawful Interception Control Function

LI\_HI1 LI\_Handover Interface 1

LI\_HI2 LI\_Handover Interface 2

LI\_HI3 LI\_Handover Interface 3

LI\_HI4 LI\_Handover Interface 4

LI\_HIQR Lawful Interception Handover Interface Query Response

LIPF Lawful Interception Provisioning Function

LIR Location Immediate Request

LI\_SI Lawful Interception System Information Interface

LI\_X1 Lawful Interception Internal Interface 1

LI\_X2 Lawful Interception Internal Interface 2

LI\_X3 Lawful Interception Internal Interface 3

LI\_XEM1 Lawful Interception Internal Interface Event Management Interface 1

LI\_XER Lawful Interception Internal Interface Event Record

LI\_XQR Lawful Interception Internal Interface Query Response

LTF Location Triggering Function

MDF Mediation and Delivery Function

MDF2 Mediation and Delivery Function 2

MDF3 Mediation and Delivery Function 3

MM Multimedia Message

MMS Multimedia Message Service

NPLI Network Provided Location Information

O&M Operations and Management

POI Point Of Interception

SIRF System Information Retrieval Function

SOI Start Of Interception

TF Triggering Function

xCC LI\_X3 Communications Content.

xIRI LI\_X2 Intercept Related Information

----------------------NEXT CHANGES---------------------

## 4.1 Introduction

The present document provides details of the internal and external interfaces required for a network operator, access provider and/or service provider to provide the necessary information to a Law Enforcement Agency (LEA) required to meet LI requirements. LI requirements for 3GPP networks and services are given in TS 33.126 [3].

The high-level architecture that defines the necessary interfaces is specified in TS 33.127 [5]. The generic high-level architecture is as follows:



Figure 4.1-1: High-level architecture diagram with key point-to-point LI interfaces

The specification of the interfaces is split into two parts:

- Internal interfaces used between an operator’s network functions are described in clause 4.2.

- External interfaces used in communicating with a LEA are described in clause 4.3.

## 4.2 Basic principles for internal interfaces

This clause lists the internal interfaces shown in clause 4.1, indicates the protocol used to realise each interface, and gives a reference to the relevant clauses of the present document that specify how the protocol is to be used for the given interface.

Table 4.2-1: Internal interfaces and related protocols

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Description | Protocol used to realise interface | Usage |
| LI\_SI | Used to provide system information to the LIPF from the SIRF. | Out of scope of the present document. |  |
| LI\_X1 | Used to configure and audit Directly-provisioned POIs, TFs and MDFs. | ETSI TS 103 221-1 [7]. | See clause 5.2.2 |
| LI\_X1 (Management) | Used to audit Triggered POIs. | ETSI TS 103 221-1 [7]. | See clause 5.2.3 |
| LI\_X2 | Used to pass xIRI from IRI-POIs to the MDF2. | ETSI TS 103 221-2 [8]. | See clause 5.3.2 |
| LI\_X3 | Used to pass xCC from CC-POIs to the MDF3. | ETSI TS 103 221-2 [8]. | See clause 5.3.3 |
| LI\_T2 | Used to pass triggering information from the IRI-TF to a Triggered IRI-POI. | ETSI TS 103 221-1 [7]. | See clause 5.2.4 |
| LI\_T3 | Used to pass triggering information from a CC-TF to a Triggered CC-POI. | ETSI TS 103 221-1 [7]. | See clause 5.2.4 |
| LI\_XQR | Used to pass queries from IQF to ICF and responses from ICF to IQF. | ETSI TS 103 221-1 [7]. | See clause 5.8 |
| LI\_XER | Used to pass identifier association event records from IEFs to ICF. | See Clause 5.9 | See clause 5.9 |
| LI\_XEM1 | Used by the LICF/LIPF to manage IEFs and ICF. | ETSI TS 103 221-1 [7]. | See clause 5.2.7 |
| LI\_ADMF | Used to pass intercept provisioning information form the LICF to the LIPF. | Out of scope of the present document. |  |
| LI\_MDF | Used by MDF2 and MDF3 in interactions necessary to correctly generate CC and IRI from xCC and xIRI. | Out of scope of the present document. |  |
| LI\_IQF | Used to pass information related to IEFs and ICFs to IQF. | Out of scope of the present document. |  |

## 4.3 Basic principles for external handover interfaces

This clause lists the external handover interfaces shown in clause 4.1, indicates the protocol used to realise each interface, and gives a reference to the relevant clauses of the present document that specify how the protocol is to be used for the given interface.

Table 4.3-1: External handover interfaces and related protocols

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Description | Protocol used to realise interface | Usage |
| LI\_HI1 | Used to send warrant and other interception request information from LEA to operator. | ETSI TS 103 120 [6] shall be supported  Other methods (e.g. manual exchange) may be used depending on national regulatory requirements. | See clause 5.4 |
| LI\_HI2 | Used to send IRI from the MDF2 to the LEMF | ETSI TS 102 232-1 [9] and ETSI TS 102 232-7 [10] shall be supported | See clause 5.5 |
| LI\_HI3 | Used to send CC from the MDF3 to the LEMF | ETSI TS 102 232-1 [9] and ETSI TS 102 232-7 [10] shall be supported | See clause 5.5 |
| LI\_HI4 | Used to send LI notification information from MDF2/3 to LEMF | ETSI TS 102 232-1 [9] and ETSI TS 102 232-7 [10] shall be supported | See clause 5.6 |
| LI\_HIQR | Used to send warrant and other identifier association query information from LEA to CSP and used by the CSP to send query responses to the LEA. | ETSI TS 103 120 [6] shall be supported | See clause 5.7 |

----------------------NEXT CHANGES---------------------

### 5.2.7 Usage for realising LI\_XEM1

For the purposes of realising LI\_XEM1 between the LIPF and an IEF, the LIPF plays the role of the “ADMF” as defined in ETSI TS 103 221-1 [7] reference model (clause 4.2), and the IEF plays the role of the “NE”.

The IEF shall be enabled by sending the following ActivateTask message from the LIPF.

Table 5.2.7-1: ActivateTask message for activating an IEF

|  |  |  |
| --- | --- | --- |
| ETSI TS 103 221-1 field name | Description | M/C/O |
| XID | Shall be set to a value assigned by the LIPF. | M |
| TargetIdentifiers | Shall contain a single Target Identifier of type "IdentityAssociation" (see table 5.2.7-2) | M |
| DeliveryType | Set to "X2Only". | M |
| ListOfDIDs | Shall give the DID of the delivery endpoint of the ICF(s) to which identity association events should be delivered. These delivery endpoints are configured using the CreateDestination message as described in ETSI TS 103 221-1 [7] clause 6.3.1 prior to the task activation. | M |

The following Target Identifier Type is defined for the use of LI\_XEM1. Unless otherwise specified, use of any other Target Identifier Type (including adding a target identifier more than once) shall result in the ActivateTask message being rejected with the appropriate error.

Table 5.2.7-2: Target Identifier Type for LI\_XEM1

|  |  |  |
| --- | --- | --- |
| Identifier type | ETSI TS 103 221-1 TargetIdentifier type | Definition |
| IdentityAssociationTargetIdentifier | TargetIdentifierExtension / IdentityAssociationTargetIdentifier | Empty tag (see XSD schema) |

The IEF may be reconfigured to send identity associations to a different ICF using a ModifyTask message to modify the delivery destinations.

The IEF shall be disabled by sending the following DeactivateTask message from the LIPF.

Table 5.2.7-3: DeactivateTask message for de-activating an IEF

|  |  |  |
| --- | --- | --- |
| ETSI TS 103 221-1 field name | Description | M/C/O |
| XID | Shall be set to the value assigned by the LIPF | M |

The LIPF should send one ActivateTask command to each IEF.

NOTE: The IEF may receive multiple ActivateTask messages conforming to Table 5.2.7-1, each of which can be independently deactivated. The IEF shall remain active as long as at least one valid Task remains active.

----------------------NEXT CHANGES---------------------

## 5.7 Protocols for LI\_HIQR

### 5.7.1 General

Functions having an LI\_HIQR interface shall support the use of ETSI TS 103 120 [6] to realise the interface.

In the event of a conflict between ETSI TS 103 120 [6] and the present document, the terms of the present document shall apply.

### 5.7.2 Usage for realising LI\_HIQR

#### 5.7.2.1 Request structure

LI\_HIQR requests are represented as an LDTaskObject (see ETSI TS 103 120 [6] clause 8.3), populated as follows:

Table 5.7.2-1: LDTaskObject representation of LI\_HIQR request

|  |  |  |
| --- | --- | --- |
| Field | Value | M/C/O |
| Reference | Reference to the authorization under which the request is made. The format of this field, and any procedures for allocating or validating it, are for national agreement. | M |
| DesiredStatus | Shall be set to "AwaitingDisclosure" | M |
| RequestDetails | Set according to table 5.7.2-2 below | M |

Table 5.7.2-2: RequestDetails structure

|  |  |  |
| --- | --- | --- |
| Field | Value | M/C/O |
| RequestType | Shall be set to one of the values as defined in Table 5.7.2-3 | M |
| ObservedTime | When the requestValues provides a temporary identity, this field shall be set to the observation time of that temporary identity.  When the requestValues provides a permanent identity, this is the time at which the LEA requires that the permanent to temporary association is applicable. | M |
| RequestValues | Set to the target identifier plus additional information required (see clause 5.7.2.2) | M |

NOTE: If the observed time is in the past, providing a successful query response is subject to associations still being available in the cache when the query is made to the ICF.

Table 5.7.2-3: RequestType Dictionary for LI\_HIQR

|  |  |
| --- | --- |
| Dictionary Owner | Dictionary Name |
| 3GPP | RequestType |
|  | |
| Defined DictionaryEntries | |
| Value | Meaning |
| IdentityAssociation | A request for a single IdentityAssociation response to the query provided |
| OngoingIdentityAssociation | A request for an ongoing series of IdentityAssociation responses matching the query provided. May only be used when the RequestValues contains a permanent identifier. The request shall be terminated by updating the LDTaskObject DesiredStatus to "Disclosed". |

Table 5.7.2-3 is formatted in accordance with ETSI TS 103 120 [6] Annex F.

#### 5.7.2.2 Request parameters

The RequestValues field shall contain one of the following:

- SUPI, given in either SUPIIMSI or SUPINAI formats as defined in ETSI TS 103 120 [6] clause C.2

- SUCI, given as defined in Table 5.7.2-4 below.

- 5G-S-TMSI, given as defined in Table 5.7.2-4 below

- 5G-GUTI, given as defined in Table 5.7.2-4 below

If a temporary identity is provided, the following shall also be present as RequestValues:

- CellIdentity, given as defined in Table 5.7.2-4 below.

- TrackingAreaIdentity, given as defined in Table 5.7.2-4 below.

The following RequestValue FormatTypes (see ETSI TS 103 120 [6] clause 8.3.5.4) are defined (which are not otherwise defined elsewhere).

Table 5.7.2-4: RequestValue FormatType extensions for LI\_HIQR Requests

| Format Owner | Format Name | Description | Format |
| --- | --- | --- | --- |
| 3GPP | SUCI | Subscription Concealed Identifier as per TS 23.003 [19] clause 2.2B | TS 29.509 [XA] clause 6.1.6.3.2 |
| 3GPP | 5GSTMSI | Shortened form of the 5G-GUTI. As defined in TS 23.003 [19] clause 2.11. Given as a hyphen-separated concatenation of:   * The string "5gstmsi" * The AMF Set ID given as three hexadecimal digits * The AMF Pointer given as two hexadecimal digits * The 5G-TMSI given as eight hexadecimal digits | Matches regular expression:  ^(5gstmsi-([0-3][0-9A-Fa-f]{2})-([0-3][0-9A-Fa-f])-([0-9A-Fa-f]{8}))$ |
| 3GPP | 5GGUTI | As defined in TS 23.003 [19] clause 2.10. Given as a hyphen separated concatenation of:   * The string "5gguti" * MCC given as a three decimal digits * MNC given as a two or three digit decimal digits * AMF region ID given as two hexadecimal digits * The AMF Set ID, AMF Pointer and 5G-TMSI as defined above | Matches regular expression:  ^(5gguti-([0-9]{3})-([0-9]{2,3})-([0-9A-Fa-f]{2})-([0-3][0-9A-Fa-f]{2})-([0-3][0-9A-Fa-f])-([0-9A-Fa-f]{8}))$ |
| 3GPP | NRCellIdentity | NR Cell ID (NCI), as defined in TS 23.003 [19] clause 19.6A | TS 29.571 [17] clause 5.4.2 |
| 3GPP | TrackingAreaCode | Tracking area code as defined in TS 23.003 [19] clause 19.4.2.3 | TS 29.571 [17] clause 5.4.2 |

#### 5.7.2.3 Response structure

The LI\_HIQR request is used to generate a request to the ICF over LI\_XQR (see clause 5.8). The response received over LI\_XQR is then transformed into an LI\_HIQR response.

LI\_HIQR responses are represented as XML following the IdentityAssociationResponse schema (see Annex E), delivered using a DELIVERY object (see ETSI TS 103 120 [6] clause 10).

The fields of each IdentityAssociationRecord shall be set as follows:

Table 5.7.2-5: IdentityAssociationRecord

|  |  |  |
| --- | --- | --- |
| Field | Value | M/C/O |
| SUPI | SUPI associated with the provided identity. | M |
| SUCI | SUCI associated with the provided identity, if available. | C |
| 5G-GUTI | 5G GUTI associated with the provided identity. | M |
| PEI | PEI associated with the provided identity during the association period, if known | C |
| AssociationStartTime | The time that the association between the SUPI and the temporary identity became valid. (See NOTE 1). | M |
| AssociationEndTime | The time that the association between the SUPI and the temporary identity ceased to be valid. Shall be omitted if the association is still valid. (See NOTE 1). | C |
| NOTE 1: If the association between the identifiers is only valid at a single point in time (i.e. SUCI), the AssociationStartTime and AssociationEndTime values shall both be set to that point in time. | | |

If no association is found which matches the criteria provided in the LI\_XQR request, then the LI\_XQR response contains zero records. Similarly, the LI\_HIQR response contains zero records.

The DeliveryObject Reference field (see ETSI TS 103 120 [6] clause 10.2.1) shall be set to the Reference of the LDTaskObject used in the request, to provide correlation between request and response.

The content manifest (see ETSI TS 103 120 [6] clause 10.2.2) shall be set to indicate the present document, using the following Specification Dictionary extension.

Table 5.7.2-6: Specification Dictionary

|  |  |
| --- | --- |
| Dictionary Owner | Dictionary Name |
| 3GPP | ManifestSpecification. |
|  | |
| Defined DictionaryEntries | |
| Value | Meaning |
| LIHIQRResponse | The delivery is according to the LIHIQRResponse schema (see Annex E) |

## 5.8 Protocols for LI\_XQR

LI\_XQR requests are realised using TS 103 221-1 [7] to transport the IdentityAssociationRequest and IdentityAssociationResponse messages (which are derived from the X1RequestMessage and X1ResponseMessage definitions in TS 103 221-1 [7]) as described in Annex E. The IdentityAssociationRequest message is populated as follows:

Table 5.8-1: IdentityAssociationRequest message for LI\_XQR

|  |  |  |
| --- | --- | --- |
| ETSI TS 103 221-1 field name | Description | M/C/O |
| ObservedTime | Observation time as provided over LI\_HIQR (see clause 5.7.2) | M |
| RequestValues | Set to the target identifier plus additional information specified in the LI\_HIQR request (see clause 5.7.2) | M |

Successful LI\_XQR responses are returned using the IdentityAssociationResponse message. Error conditions are reported using the normal error reporting mechanisms described in TS 103 221-1 [7].

LI\_XQR query responses are represented in XML following the IdentityAssociationResponse schema (see Annex E). The fields of the IdentityAssociationResponse record shall be populated as described in Table 5.7.2-5.

## 5.9 Protocols for LI\_XER

LI\_XER records are realised using a TLS connection as defined in clause 6.2.2A.2.3, with records BER-encoded as defined in Annex F.

----------------------NEXT CHANGES---------------------

#### 6.2.2.1 Provisioning over LI\_X1

The IRI-POI present in the AMF is provisioned over LI\_X1 by the LIPF using the X1 protocol as described in clause 5.2.2.

The POI in the AMF shall support the following target identifier formats in the ETSI TS 103 221-1 [7] messages (or equivalent if ETSI TS 103 221-1 [7] is not used):

- SUPIIMSI.

- SUPINAI.

- PEIIMEI.

- PEIIMEISV.

- GPSIMSISDN.

- GPSINAI.

Table 6.2.2.1-1 shows the minimum details of the LI\_X1 ActivateTask message used for provisioning the IRI-POI in the AMF.

Table 6.2.2.1-1: ActivateTask message for the IRI-POI in the AMF

|  |  |  |
| --- | --- | --- |
| ETSI TS 103 221-1 field name | Description | M/C/O |
| XID | XID assigned by LIPF. | M |
| TargetIdentifiers | One of the target identifiers listed in the paragraph above. | M |
| DeliveryType | Set to "X2Only". | M |
| ListOfDIDs | Delivery endpoints for LI\_X2 for the IRI-POI in the AMF. These delivery endpoints are configured using the CreateDestination message as described in ETSI TS 103 221-1 [7] clause 6.3.1 prior to the task activation. | M |
| TaskDetailsExtensions/  IdentifierAssociationExtensions | This field shall be included if the IRI POI is required to generate AMFIdentifierAssociation records (see clause 6.2.2.2.1). If the field is absent, AMFIdentifierAssociation records shall not be generated. | C |

Table 6.2.2.1-2: IdentifierAssociationExtensions Parameters

|  |  |  |
| --- | --- | --- |
| Field Name | Description | M/C/O |
| EventsGenerated | One of the following values:   * IdentifierAssocation * All   See clause 6.2.2.2.1 for the interpretation of this field. | M |

#### 6.2.2.2 Generation of xIRI over LI\_X2

##### 6.2.2.2.1 General

The IRI-POI present in the AMF shall send the xIRIs over LI\_X2 for each of the events listed in TS 33.127 [5] clause 6.2.2.4, the details of which are described in the following sub-clauses.

If the AMF receives one or more cell IDs in an N2 message (as specified in TS 38.413 [23]), the POI associated with the AMF shall report all of them.

The IRI-POI in the AMF shall only generate AMFIdentifierAssociation records when the IdentifierAssocationExtensions parameter has been received over LI\_X1 (see clause 6.2.2.1). If the IdentifierAssocationExtensions parameter is not received for a specific target the IRI-POI shall not generate AMFIdentifierAssociation records for that target. The AMF shall generate records according to the value of the EventsGenerated sub-parameter (see Table 6.2.2.1-2) as follows:

* IdentifierAssociation: AMFIdentifierAssociation and AMFLocationUpdate records shall be generated. No other record types shall be generated for that target.
* All: All AMF record types shall be generated.

##### 6.2.2.2.2 Registration

The IRI-POI in the AMF shall generate an xIRI containing an AMFRegistration record when the IRI-POI present in the AMF detects that a UE matching one of the target identifiers provided via LI\_X1 has successfully registered to the 5GS via 3GPP NG-RAN or non-3GPP access. Accordingly, the IRI-POI in the AMF generates the xIRI when the following event is detected:

- AMF sends a N1: REGISTRATION ACCEPT message to the target UE and the UE 5G Mobility Management (5GMM) state for the access type (3GPP NG-RAN or non-3GPP access) within the AMF is changed to 5GMM-REGISTERED.

Table 6.2.2-1: Payload for AMFRegistration record

|  |  |  |
| --- | --- | --- |
| Field name | Description | M/C/O |
| registrationType | Specifies the type of registration, see TS 24.501 [13] clause 9.11.3.7. This is derived from the information received from the UE in the REGISTRATION REQUEST message. | M |
| registrationResult | Specifies the result of registration, see TS 24.501 [13] clause 9.11.3.6. | M |
| slice | Provide, if available, one or more of the following:  - allowed NSSAI (see TS 24.501 [13] clause 9.11.3.37).  - configured NSSAI (see TS 24.501 [13] clause 9.11.3.37),  - rejected NSSAI (see TS 24.501 [13] clause 9.11.3.46).  This is derived from the information sent to the UE in the REGISTRATION ACCEPT message. | C |
| sUPI | SUPI associated with the registration (see clause 6.2.2.4). | M |
| sUCI | SUCI used in the registration, if available. | C |
| pEI | PEI provided by the UE during the registration, if available. | C |
| gPSI | GPSI obtained in the registration, if available as part of the subscription profile. | C |
| gUTI | 5G-GUTI provided as outcome of initial registration or used in other cases, see TS 24.501 [13] clause 5.5.1.2.2. | M |
| location | Location information determined by the network during the registration, if available.  Encoded as a *userLocation* parameter (*location>locationInfo>userLocation*) and, when Dual Connectivity is activated, as an *additionalCellIDs* parameter (*location>locationInfo>additionalCellIDs*), see Annex A. | C |
| non3GPPAccessEndpoint | UE's local IP address used to reach the N3IWF, if available. IP addresses are given as 4 octets (for IPv4) or 16 octets (for IPv6) with the most significant octet first (network byte order). | C |
| fiveGSTAIList | List of tracking areas associated with the registration area within which the UE is current registered, see TS 24.501 [13], clause 9.11.3.4. (See NOTE 1) | C |
| NOTE 1: List shall be included each time there is a change to the registration area. | | |

----------------------NEXT CHANGES---------------------

##### 6.2.2.2.5 Start of interception with registered UE

The IRI-POI in the AMF shall generate an xIRI containing an AMFStartOfInterceptionWithRegisteredUE record when the IRI-POI present in the AMF detects that interception is activated on a UE that has already been registered in the 5GS (see clause 6.2.2.4 on identity privacy). A UE is considered already registered to the 5GS when the 5GMM state for the access type (3GPP NG-RAN or non-3GPP access) for that UE is 5GMM-REGISTERED. Therefore, the IRI-POI present in the AMF shall generate the xIRI AMFStartOfInterceptionWithRegisteredUE record when it detects that a new interception for a UE is activated (i.e. provisioned by the LIPF) and the 5G mobility management state for the access type (3GPP NG-RAN or non-3GPP access) within the AMF for that UE is 5GMM-REGISTERED. If the UE is registered over both 3GPP NG-RAN and non-3GPP access, the IRI-POI present in the AMF shall generate an xIRI containing an AMFStartOfInterceptionWithRegisteredUE record for each access type.

Table 6.2.2-4: Payload for AMFStartOfInterceptionWithRegisteredUE record

|  |  |  |
| --- | --- | --- |
| Field name | Description | M/C/O |
| registrationResult | Specifies the result of registration, see TS 24.501 [13], clause 9.11.3.6. | M |
| registrationType | Specifies the type of registration, see TS 24.501 [13] clause 9.11.3.7, if available. | C |
| slice | Provide, if available, one or more of the following:  - allowed NSSAI (see TS 24.501 [13] clause 9.11.3.37).  - configured NSSAI (see TS 24.501 [13] clause 9.11.3.37).  - rejected NSSAI (see TS 24.501 [13] clause 9.11.3.46).  This is derived from the information that was sent to the UE in the REGISTRATION ACCEPT message. IRI-POI in AMF can include this information if and only if it retained the information that it had previously sent in the REGISTRATION ACCEPT message to the UE. | C |
| sUPI | SUPI associated with the registration (see clause 6.2.2.4). | M |
| sUCI | SUCI used in the registration, if available. | C |
| pEI | PEI provided by the UE during the registration, if available. | C |
| gPSI | GPSI obtained in the registration, if available as part of the subscription profile. | C |
| gUTI | 5G-GUTI provided as outcome of initial registration or used in other cases, see TS 24.501 [13], clause 5.5.1.2.2. | M |
| location | Location information, if available.  Encoded as a *userLocation* parameter (*location>locationInfo>userLocation*) and, when Dual Connectivity is activated, as an *additionalCellIDs* parameter (*location>locationInfo>additionalCellIDs*), see Annex A. | C |
| non3GPPAccessEndpoint | UE's local IP address used to reach the N3IWF, if available. IP addresses are given as 4 octets (for IPv4) or 16 octets (for IPv6) with the most significant octet first (network byte order). | C |
| timeOfRegistration | Time at which the last registration occurred, if available. This is the time stamp when the REGISTRATION ACCEPT message is sent to the UE or (when applicable) when the REGISTRATION COMPLETE is received from the UE.  Shall be given qualified with time zone information (i.e. as UTC or offset from UTC, not as local time). | C |
| fiveGSTAIList | List of tracking areas associated with the registration area within which the UE is current registered, see TS 24.501 [13], clause 9.11.3.4. (See NOTE 1) | C |
| NOTE 1: List shall be included each time there is a change to the registration area. | | |

----------------------NEXT CHANGES---------------------

##### 6.2.2.2.7 AMF identifier association

The IRI-POI present in the AMF shall generate an xIRI containing an AMFIdentifierAssociation record when the IRI-POI present in the AMF detects a new identifier association for a UE matching one of the target identifiers provided via LI\_X1. Generation of this record is subject to this record type being enabled for a specific target (see clause 6.2.2.2.1).

Table 6.2.2-6: Payload for AMFIdentifierAssociation record

|  |  |  |
| --- | --- | --- |
| Field name | Description | M/C/O |
| sUPI | SUPI associated with the procedure, if available (see NOTE 1). | M |
| sUCI | SUCI used in the procedure, if applicable and if available. | C |
| pEI | PEI used in the procedure, if available (see NOTE 1). | C |
| gPSI | GPSI used in the procedure, if available (see NOTE 1). | C |
| gUTI | 5G-GUTI used in the procedure, if available, see TS 24.501 [13], clause 9.11.3.4. | M | |
| location | Location information available when identifier association occurs.  Encoded as a *userLocation* parameter (*location>locationInfo>userLocation*) and, when Dual Connectivity is activated, as an *additionalCellIDs* parameter (*location>locationInfo>additionalCellIDs*), see Annex A. | M |
| fiveGSTAIList | List of tracking areas associated with the registration area within which the UE is current registered, see TS 24.501 [13], clause 9.11.3.4. (See NOTE 2) | C |
| NOTE 1: SUPI shall always be provided, in addition to the warrant target identifier if different to SUPI. Other identifiers shall be provided if available.  NOTE 2: List shall be included each time there is a change to the registration area. | | |

----------------------NEXT CHANGES---------------------

#### 6.2.2.3 Generation of IRI over LI\_HI2

When an xIRI is received over LI\_X2 from the IRI-POI in AMF, the MDF2 shall generate the corresponding IRI message and deliver over LI\_HI2 without undue delay. The IRI message shall contain a copy of the relevant record received in the xIRI over LI\_X2. This record may be enriched with any additional information available at the MDF (e.g. additional location information).

The timestamp field of the psHeader structure shall be set to the time at which the AMF event was observed (i.e. the timestamp field of the X2 PDU).

Table 6.2.2-6 shows the IRI type (see ETSI TS 102 232-1 [9] clause 5.2.10) to be used for each IRI message.

Table 6.2.2-6: IRI type for IRI messages

|  |  |
| --- | --- |
| IRI message | IRI type |
| AMFRegistration | REPORT |
| AMFDeregistration | REPORT |
| AMFLocationUpdate | REPORT |
| AMFStartOfInterceptionWithRegisteredUE | REPORT |
| AMFUnsuccessfulProcedure | REPORT |
| AMFIdentifierAssociation | REPORT |

These IRI messages shall omit the CIN (see ETSI TS 102 232-1 [9] clause 5.2.4).

The threeGPP33128DefinedIRI field in ETSI TS 102 232-7 [10] clause 15 shall be populated with the BER-encoded IRIPayload.

When an additional warrant is activated on a target UE and the LIPF uses the same XID for the additional warrant, the MDF2 shall be able to generate and deliver the IRI message containing the AMFStartOfInterceptionWithRegisteredUE record to the LEMF associated with the additional warrant without receiving a corresponding xIRI. The payload of the AMFStartOfInterceptionWithRegisteredUE record is specified in table 6.2.2-4.

----------------------NEXT CHANGES---------------------

### 6.2.2A Identifier Reporting for AMF

#### 6.2.2A.1 Activation of reporting over LI\_XEM1

The IEF in the AMF is activated and deactivated over LI\_XEM1 by the LIPF using the LI\_XEM1 protocol described in clause 5.2.7.

NOTE: Since the IEF reports association events for all UEs registered in the IEF’s parent AMF, unlike POIs there is no concept of provisioning an IEF with target identifiers.

Upon receiving a valid activate task message over LI\_XEM1, the IEF shall start generating records as defined in clause 6.2.2A.2.

Upon receiving a valid deactivate task message over LI\_XEM1, the IEF shall stop generating records as defined in clause 6.2.2A.2.

#### 6.2.2A.2 Generation of records over LI\_XER

6.2.2A.2.1 Events

The IEF in the AMF shall generate an IEFIdentifierAssociation record whenever the IEF present in the AMF detects a change in association between a SUPI and a 5G-GUTI for any UE registered with the AMF. The IEF shall send the IEFIdentifierAssociation records to the ICF over LI\_XER as defined in clause 5.9.

Accordingly, the IEF in the AMF generates IEFIdentifierAssociation records when any of the following events are detected;

* IEFAssociationRecord: Association of a 5G-GUTI to a SUPI, (this may also include SUCI to SUPI association).
* IEFDeassociationRecord: De-association of a 5G-GUTI from a SUPI.

NOTE1: The de-association of 5G-GUTI from a SUPI event record is only generated if a new 5G-GUTI is not allocated to a SUPI to update a previous association (e.g. at inter-AMF handover).

NOTE 2: As SUCIs are single use and only valid for a single authentication, they are only be valid at the single point in time when the association event is detected and reported to the ICF by the IEF.

In addition, when an IEF is activated as per clause 6.2.2A.1, the IEF shall generate associations event for all SUPIs which are registered in the AMF, where those identifier associations allocated prior to IEF activation remain current and are still available in the AMF (See NOTE 2).

NOTE 3: Only identifier associations which have been maintained by the AMF as part of normal network operations will be available.

6.2.2A.2.2 Association Events

For each association event, the IEF shall create an IEFAssociationRecord, as defined below.

Table 6.2.2A-1: Payload for IEFAssociationRecord

|  |  |  |
| --- | --- | --- |
| Field name | Description | M/C/O |
| sUPI | SUPI associated with detected association event. | M |
| fiveGGUTI | 5G-GUTI shall be provided. Encoded as per TS 24.501 [13] figure 9.11.3.4.1, omitting the first four octets. | M |
| timeStamp | Time at which the identifier association event occurred.  Shall be given qualified with time zone information (i.e. as UTC or offset from UTC, not as local time). | M |
| tAI | Last known TAI associated with the SUPI. Encoded as per TS 24.501 [13] clause 9.1.3.8, omitting the first octet. | M |
| nCGI | Last known nCGI(s) available when identifier association event detected. Given as a sequence of PLMNID (encoded as per TS 38.413 [23] clause 9.3.3.5) and NCI (encoded as per TS 38.413 [23] clause 9.3.1.7). | M |
| nCGITime | ueLocationTimestamp(s) of nCGIs if available in AMF as per TS 29 .571 [17] clause 5.4.4.9.  If ueLocationTimestamp(s) is not available, shall be populated with timeStamp(s) of when last known nCGI(s), were obtained and stored by the AMF. | M |
| sUCI | SUCI shall be provided when event is triggered by association of a SUCI to a SUPI. | C |
| pEI | PEI, (See NOTE 1). | C |
| fiveGSTAIList | List of tracking areas associated with the registration area within which the UE is current registered, see TS 24.501 [13], clause 9.11.3.4. (See NOTE 2) | C |
| NOTE 1: Shall be provided in first association record to ICF after PEI is available and following any change of PEI.  NOTE 2: As a minimum, list of tracking areas shall be included in the first association event for each SUPI registered (per UE session) with the AMF and additionally whenever the TAI list changes due to a change in registration area. | | |

For each de-association event, the IEF shall create an IEFDeassociationRecord, as defined below.

**Table 6.2.2A-2: Payload for IEFDeassociationRecord**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Description** | **M/C/O** |
| sUPI | SUPI associated with detected de-association event. | M |
| fiveGGUTI | 5G-GUTI shall be provided. Encoded as per TS 24.501 [13] figure 9.11.3.4.1, omitting the first four octets. | M |
| timeStamp | Time at which the identifier de-association event occurred.  Shall be given qualified with time zone information (i.e. as UTC or offset from UTC, not as local time). | M |
| nCGI | Last known nCGI(s) available when identifier de-association event detected. Given as a sequence of PLMNID (encoded as per TS 38.413 [23] clause 9.3.3.5) and NCI (encoded as per TS 38.413 [23] clause 9.3.1.7) | M |
| nCGITime | ueLocationTimestamp(s) of nCGIs if available in AMF as per TS 29 .571 [17] clause 5.4.4.9.  If ueLocationTimestamp(s) is not available, shall be populated with timeStamp(s) of when last known nCGI(s), were obtained and stored by the AMF. | M |

6.2.2A.2.3 Transmission to the ICF

When activated (see clause 5.2.7), the IEF shall establish a TLS connection to the ICF(s) as given over LI\_XEM1. If the IEF fails to establish a TLS connection, it shall report an error over LI\_XEM1 using the error reporting mechanisms described in TS 103 221-1 [7], and attempt to reconnect after a configurable period of time.

When a record has been generated as described in clause 6.2.2A.2.2, the IEF shall encode the IEFAssociationRecord or IEFDeassociationRecord as a BER-encoded IEFMessage structure, following the ASN.1 schema given in Annex F, and transmit it to the ICF over the established TLS connection.

The IEF may transmit a keepalive request using the keepalive record defined in Annex F. Upon receiving a keepalive request, the ICF shall respond with a keepaliveResponse record containing the same sequence number used in the request. The circumstances under which the IEF transmits keepalive requests is out of scope of the present document.

----------------------NEXT CHANGES---------------------

#### 6.3.2.2 Generation of xIRI over LI\_X2

##### 6.3.2.2.1 General

The IRI-POI present in the MME shall send the xIRIs over LI\_X2 for each of the events listed in TS 33.107 [36] clause 12.2.1.1, the details of which are specified in clause 12.2.3 of the same TS, and in case of SMS over NAS as specified in clause 18.2.4 of TS 33.107 [36].

For all records except MMEIdentifierAssociation (see clause 6.3.2.2.2), the IRI-POI present in the MME shall set the payload format to EpsHI2Operations.EpsIRIContent (value 14), see clause 5.3 and ETSI TS 103 221-2 [8] clause 5.4. The payload field shall contain an EpsHI2Operations.EpsIRIContent structure encoded according to TS 33.108 [12] clauses 10.5, 15.2 and B.9.

As the LIID may be not available at the MME but is mandatory in EpsHI2Operations.EpsIRIContent according to Annex B.9 of TS 33.108 [12], its value in the lawfulInterceptionIdentifier field of the encoded PDU shall be set to the fixed string "LIIDNotPresent".

In addition to the xIRIs events listed in TS 33.107 [36], the MME shall support the MMEIdentiferAssociation record in clause 6.3.2.2.2.

The IRI-POI in the MME shall only generate the MMEIdentifierAssociation record in the following scenarios;

* IdentifierAssociation: MMEIdentifierAssociation and Tracking Area/EPS Location Update (see TS 33.107 [36] clause 12.2.1.2) records shall be generated. No other record types shall be generated for that target.
* All: All AMF record types shall be generated.

##### 6.3.2.2.2 MME identifier association

The IRI-POI present in the MME shall generate an xIRI containing an MMEIdentifierAssociation record when the IRI-POI present in the MME detects a new identifier association for a UE matching one of the target identifiers provided via LI\_X1. Generation of this record is subject to this record type being enabled for a specific target (see clause 6.3.2.2.1).

Table 6.3.2-1: Payload for MMEIdentifierAssociation record

|  |  |  |
| --- | --- | --- |
| Field name | Description | M/C/O |
| iMSI | IMSI associated with the procedure. (see NOTE 1). | M |
| iMEI | IMEI used in the procedure, if available (see NOTE 1). | C |
| mSISDN | MSISDN used in the procedure, if available (see NOTE 1). | C |
| gUTI | LTE GUTI used in the procedure, if available. | M | |
| location | Location information available when identifier association occurs.  Encoded as a *userLocation* parameter (*location>locationInfo> userLocation*) and, when Dual Connectivity is activated, as an *additionalCellIDs* parameter (*location>locationInfo>additionalCellIDs*), see Annex A. | M |
| tAIList | List of tracking areas associated with the registration area within which the UE is current registered. (See NOTE 2). | C |
| NOTE 1: IMSI shall always be provided, in addition to the warrant target identifier if different to IMSI. Other identifiers shall be provided if available.  NOTE 2: List shall be included each time there is a change to the registration area. | | |

When transmitting the xIRI, the IRI-POI present in the MME shall set the payload format to 2, and provide the payload as a BER-encoded TS33128Payloads.XIRIPayloads structure.

#### 6.3.2.3 Generation of IRI over LI\_HI2

When an xIRI is received over LI\_X2 from the IRI-POI in the MME, the MDF2 shall generate the corresponding IRI message and deliver it over LI\_HI2 without undue delay. The IRI message shall contain a copy of the relevant record received in the xIRI over LI\_X2.

When option 2 specified in clause 6.3.1 is used, the MDF2 shall generate IRI messages based on the proprietary information received from the MME and provide it over LI\_HI2 without undue delay.

For all messages except MMEIdentifierAssociation, the IRI messages shall include an IRI payload encoded according to Annex B.9 of TS 33.108 [12]. The MDF2 shall encode the correct value of LIID in the IRI message, replacing the value "LIIDNotPresent" given in the xIRI (see clause 6.3.2.2).

For MMEIdentifierAssociation messages, the IRI message shall be encoded as an IRIEvent structure according to Annex B, and used to populate the threeGPP33128DefinedIRI field in ETSI TS 102 232-7 [10] clause 15.

The IRI messages shall be delivered over LI\_HI2 according to clause 10 of ETSI TS 102 232-7 [10].

----------------------NEXT CHANGES---------------------

## 7.6 Identifier Association Reporting

### 7.6.1 General

The IEF, ICF and IQF are responsible for detecting, storing and providing to the LEA permanent to temporary identifier associations, requested by the LEA in authorised requests. The IEF as defined in clause 6.2.2A is responsible for detecting and generating identifier associations records. The ICF is responsible for caching identifier associations for short duration and the IQF is responsible for handling requests from the LEA and providing those requests to the ICF in order to identify the matching identifier associations.

### 7.6.2 ICF

#### 7.6.2.1 General

The ICF is responsible for caching identifier associations provided in event records from the IEF over LI\_XER and handling queries and subsequent responses from the IQF for responses over LI\_XQR.

#### 7.6.2.2 ICF receipt of records over LI\_XER

When the ICF receives an identifier association event record over LI\_XER from an IEF (see clause 5.9), the ICF shall use the records to update the identifier associations cached by the ICF. The ICF shall handle the event records as described in clause 7.6.2.4.

#### 7.6.2.3 ICF Query and Response over LI\_XQR

When the ICF receives an identifier association query request from the IQF, the ICF shall search the cached identifier associations to establish a match, based on RequestValues received in the request (see clause 5.8), subject to clause 7.6.2.4.

Upon successful matching of 1 or more identifier associations, the ICF shall provide a response to the IQF using the IdentityAssociationResponse message as defined in clause 5.8. Where the ICF is not able to provide a single identifier association based on the RequestValues, the IQF is responsible for any subsequent handling of multiple identifier associations in terms of whether to provide all associations to the LEA over LI\_HIQR.

#### 7.6.2.4 ICF Identifier Association Event Handling

Upon receipt of an IEFIdentifierAssociation record as defined in clause 6.2.2A.2, the ICF shall cache the identifier association(s) contained within the record as followings;

* Where the IEFIdentifierAssociation record contains a new SUPI to SUCI association, the association shall be cached until the maximum cache duration is reached.
* Where no current SUPI to 5G-GUTI association is held in the ICF cache, the identifier association shall be held until a subsequent SUPI to 5G-GUTI association update is received or the CSP defined network wide re-authentication time value is exceeded (See NOTE 1).
* Where a previous SUPI to 5G-GUTI association is held in the cache, the previous association shall be subsequently cached until the maximum cache duration is reached and the new identifier association shall be held until a subsequent SUPI to 5G-GUTI association update is received or the CSP defined network wide re-authentication time value is exceeded (See NOTE 1).
* Where a SUPI to 5G\_GUTI association is indicated as de-associated by an IEF but no new 5G-GUTI has been allocated, the association shall be subsequently cached until the maximum cache duration is reached.

NOTE 1: The CSP defined network authentication time value is the time duration after which the CSP network will automatically reauthenticate UEs or perform a location update for an idle mode UE. This value is CSP defined, part of normal network operation not associated with LI functionality and outside the scope of the present document.

Where an IEFIdentifierAssociation record contains a PEI or a TAI list, the ICF shall store the received values and associate them both the current received SUPI to 5G-GUTI association (including SUPI to SUCI association where also present in the record) and any future association until;

* A subsequent IEFIdentifierAssociation record is receive which updates the PEI or TAI list values.
  + The old PEI / TAI list shall be retained in association with previous SUPI to 5G-GUTI or SUCI associations until those associations are deleted from cache.
  + New PEI / TAI list shall be used in association with both the association(s) with which it was received and any subsequent associations until another update is received.
* All SUPI associations for which the PEI / TAI list is valid are deleted from the cache.

When the ICF receives a query request from the IQF as defined in clause 7.6.2.3, the ICF shall search the cached identifier associations for a match. The ICF shall be able to use both time and TAI (as a single TAI and in relation to a TAI list) to identity the correct SUPI to 5G-GUTI or SUCI association(s).

NOTE 2: Use of nCGI to match associations based on physical location for SUCI / 5G-S-TIMSI to SUPI requests, is out of scope of the present document.

As the LEA and CSP are unlikely to have synchronised the time of identifier observation / association provided by the LEA in the query request, with NF time of the IEFs, the ICF shall search the cached identifier associations using a short window time duration both before and after (subject to maximum cache duration) the time provided by the LEA in the RequestValues over LI\_XQR.

NOTE 3: The search window duration before and after the LEA provided time value is outside the scope of the present document. The use of such a window is constrained by the availability of associations in the cache at any given point in time.

### 7.6.3 IQF

#### 7.6.2.1 General

The ICF is responsible for receiving and responding to LEA requests over LI\_HIQR. Following receipt of a request over LI\_HIQR, the IQF shall validate the request and ensure that the request is within the cache period of associations stored in the ICF. If the request if valid and within the ICF cache period, the IQF shall send an association search request to the ICF over LI\_XQR. If the request is not within the ICF cache period or overwise invalid, the IQF shall reject the request and respond to the LEA over LI\_HIQR.

Following receipt of an association search request response from the ICF over LI\_XQR, the IQF shall forward any matching identifier association(s) to the LEA over LI\_HIQR. If the ICF indicates zero matches were found based on the information provided in the initial request over LI\_HIQR, the IQF shall respond to the LEA over LI\_HIQR indicating that no identifier associations were found based on the request from the LEA.

If the ICF responds with multiple associations of 5G-GUTIs / SUCIs to a single SUPI, the IQF shall provide all matched associations to the LEA over LI\_HIQR. Handling in the case of multiple SUPIs to a single 5G-GUTI (where the initial request over LI\_HIQR is based on 5G-S-TIMSI or SUCI) is outside the scope of the present document.

#### 7.6.3.2 IQF Query and Response over LI\_HIQR

The IQF is responsible for receiving query requests from and providing query responses to the LEA over LI\_HIQR. Further details of LI\_HIQR messages are defined in clause 5.7.

#### 7.6.3.3 IQF Query and Response over LI\_XQR

The IQF is responsible for generating queries to and receiving query responses requests from the ICF over LI\_XQR, based on queries received from the LEA over LI\_HIQR. Further details of LI\_XQR messages are defined in clause 5.8.

----------------------NEXT CHANGES---------------------

Annex A (normative): Structure of both the Internal and External Interfaces

TS33128Payloads

{itu-t(0) identified-organization(4) etsi(0) securityDomain(2) lawfulIntercept(2) threeGPP(4) ts33128(19) r16(16) version4(4)}

DEFINITIONS IMPLICIT TAGS EXTENSIBILITY IMPLIED ::=

BEGIN

-- =============

-- Relative OIDs

-- =============

tS33128PayloadsOID RELATIVE-OID ::= {threeGPP(4) ts33128(19) r16(16) version4(4)}

xIRIPayloadOID RELATIVE-OID ::= {tS33128PayloadsOID xIRI(1)}

xCCPayloadOID RELATIVE-OID ::= {tS33128PayloadsOID xCC(2)}

iRIPayloadOID RELATIVE-OID ::= {tS33128PayloadsOID iRI(3)}

cCPayloadOID RELATIVE-OID ::= {tS33128PayloadsOID cC(4)}

lINotificationPayloadOID RELATIVE-OID ::= {tS33128PayloadsOID lINotification(5)}

-- ===============

-- X2 xIRI payload

-- ===============

XIRIPayload ::= SEQUENCE

{

xIRIPayloadOID [1] RELATIVE-OID,

event [2] XIRIEvent

}

XIRIEvent ::= CHOICE

{

-- Access and mobility related events, see clause 6.2.2

registration [1] AMFRegistration,

deregistration [2] AMFDeregistration,

locationUpdate [3] AMFLocationUpdate,

startOfInterceptionWithRegisteredUE [4] AMFStartOfInterceptionWithRegisteredUE,

unsuccessfulAMProcedure [5] AMFUnsuccessfulProcedure,

-- PDU session-related events, see clause 6.2.3

pDUSessionEstablishment [6] SMFPDUSessionEstablishment,

pDUSessionModification [7] SMFPDUSessionModification,

pDUSessionRelease [8] SMFPDUSessionRelease,

startOfInterceptionWithEstablishedPDUSession [9] SMFStartOfInterceptionWithEstablishedPDUSession,

unsuccessfulSMProcedure [10] SMFUnsuccessfulProcedure,

-- Subscriber-management related events, see clause 7.2.2

servingSystemMessage [11] UDMServingSystemMessage,

-- SMS-related events, see clause 6.2.5

sMSMessage [12] SMSMessage,

-- LALS-related events, see clause 7.3.3

lALSReport [13] LALSReport,

-- PDHR/PDSR-related events, see clause 6.2.3.4.1

pDHeaderReport [14] PDHeaderReport,

pDSummaryReport [15] PDSummaryReport,

-- tag 16 is reserved because there is no equivalent mDFCellSiteReport in XIRIEvent

-- MMS-related events, see clause 7.4.2

mMSSend [17] MMSSend,

mMSSendByNonLocalTarget [18] MMSSendByNonLocalTarget,

mMSNotification [19] MMSNotification,

mMSSendToNonLocalTarget [20] MMSSendToNonLocalTarget,

mMSNotificationResponse [21] MMSNotificationResponse,

mMSRetrieval [22] MMSRetrieval,

mMSDeliveryAck [23] MMSDeliveryAck,

mMSForward [24] MMSForward,

mMSDeleteFromRelay [25] MMSDeleteFromRelay,

mMSDeliveryReport [26] MMSDeliveryReport,

mMSDeliveryReportNonLocalTarget [27] MMSDeliveryReportNonLocalTarget,

mMSReadReport [28] MMSReadReport,

mMSReadReportNonLocalTarget [29] MMSReadReportNonLocalTarget,

mMSCancel [30] MMSCancel,

mMSMBoxStore [31] MMSMBoxStore,

mMSMBoxUpload [32] MMSMBoxUpload,

mMSMBoxDelete [33] MMSMBoxDelete,

mMSMBoxViewRequest [34] MMSMBoxViewRequest,

mMSMBoxViewResponse [35] MMSMBoxViewResponse,

-- PTC-related events, see clause 7.5.2

pTCRegistration [36] PTCRegistration,

pTCSessionInitiation [37] PTCSessionInitiation,

pTCSessionAbandon [38] PTCSessionAbandon,

pTCSessionStart [39] PTCSessionStart,

pTCSessionEnd [40] PTCSessionEnd,

pTCStartOfInterception [41] PTCStartOfInterception,

pTCPreEstablishedSession [42] PTCPreEstablishedSession,

pTCInstantPersonalAlert [43] PTCInstantPersonalAlert,

pTCPartyJoin [44] PTCPartyJoin,

pTCPartyDrop [45] PTCPartyDrop,

pTCPartyHold [46] PTCPartyHold,

pTCMediaModification [47] PTCMediaModification,

pTCGroupAdvertisement [48] PTCGroupAdvertisement,

pTCFloorControl [49] PTCFloorControl,

pTCTargetPresence [50] PTCTargetPresence,

pTCParticipantPresence [51] PTCParticipantPresence,

pTCListManagement [52] PTCListManagement,

pTCAccessPolicy [53] PTCAccessPolicy,

-- Identifier Association events, see clauses 6.2.2.2.7 and 6.3.2.2.2

aMFIdentifierAssocation [62] AMFIdentifierAssocation,

mMEIdentifierAssocation [63] MMEIdentifierAssocation

}

-- ==============

-- X3 xCC payload

-- ==============

-- No additional xCC payload definitions required in the present document.

-- ===============

-- HI2 IRI payload

-- ===============

IRIPayload ::= SEQUENCE

{

iRIPayloadOID [1] RELATIVE-OID,

event [2] IRIEvent,

targetIdentifiers [3] SEQUENCE OF IRITargetIdentifier OPTIONAL

}

IRIEvent ::= CHOICE

{

-- Registration-related events, see clause 6.2.2

registration [1] AMFRegistration,

deregistration [2] AMFDeregistration,

locationUpdate [3] AMFLocationUpdate,

startOfInterceptionWithRegisteredUE [4] AMFStartOfInterceptionWithRegisteredUE,

unsuccessfulRegistrationProcedure [5] AMFUnsuccessfulProcedure,

-- PDU session-related events, see clause 6.2.3

pDUSessionEstablishment [6] SMFPDUSessionEstablishment,

pDUSessionModification [7] SMFPDUSessionModification,

pDUSessionRelease [8] SMFPDUSessionRelease,

startOfInterceptionWithEstablishedPDUSession [9] SMFStartOfInterceptionWithEstablishedPDUSession,

unsuccessfulSessionProcedure [10] SMFUnsuccessfulProcedure,

-- Subscriber-management related events, see clause 7.2.2

servingSystemMessage [11] UDMServingSystemMessage,

-- SMS-related events, see clause 6.2.5

sMSMessage [12] SMSMessage,

-- LALS-related events, see clause 7.3.3

lALSReport [13] LALSReport,

-- PDHR/PDSR-related events, see clause 6.2.3.4.1

pDHeaderReport [14] PDHeaderReport,

pDSummaryReport [15] PDSummaryReport,

-- MDF-related events, see clause 7.3.4

mDFCellSiteReport [16] MDFCellSiteReport,

-- MMS-related events, see clause 7.4.2

mMSSend [17] MMSSend,

mMSSendByNonLocalTarget [18] MMSSendByNonLocalTarget,

mMSNotification [19] MMSNotification,

mMSSendToNonLocalTarget [20] MMSSendToNonLocalTarget,

mMSNotificationResponse [21] MMSNotificationResponse,

mMSRetrieval [22] MMSRetrieval,

mMSDeliveryAck [23] MMSDeliveryAck,

mMSForward [24] MMSForward,

mMSDeleteFromRelay [25] MMSDeleteFromRelay,

mMSDeliveryReport [26] MMSDeliveryReport,

mMSDeliveryReportNonLocalTarget [27] MMSDeliveryReportNonLocalTarget,

mMSReadReport [28] MMSReadReport,

mMSReadReportNonLocalTarget [29] MMSReadReportNonLocalTarget,

mMSCancel [30] MMSCancel,

mMSMBoxStore [31] MMSMBoxStore,

mMSMBoxUpload [32] MMSMBoxUpload,

mMSMBoxDelete [33] MMSMBoxDelete,

mMSMBoxViewRequest [34] MMSMBoxViewRequest,

mMSMBoxViewResponse [35] MMSMBoxViewResponse,

-- PTC-related events, see clause 7.5.2

pTCRegistration [36] PTCRegistration,

pTCSessionInitiation [37] PTCSessionInitiation,

pTCSessionAbandon [38] PTCSessionAbandon,

pTCSessionStart [39] PTCSessionStart,

pTCSessionEnd [40] PTCSessionEnd,

pTCStartOfInterception [41] PTCStartOfInterception,

pTCPreEstablishedSession [42] PTCPreEstablishedSession,

pTCInstantPersonalAlert [43] PTCInstantPersonalAlert,

pTCPartyJoin [44] PTCPartyJoin,

pTCPartyDrop [45] PTCPartyDrop,

pTCPartyHold [46] PTCPartyHold,

pTCMediaModification [47] PTCMediaModification,

pTCGroupAdvertisement [48] PTCGroupAdvertisement,

pTCFloorControl [49] PTCFloorControl,

pTCTargetPresence [50] PTCTargetPresence,

pTCParticipantPresence [51] PTCParticipantPresence,

pTCListManagement [52] PTCListManagement,

pTCAccessPolicy [53] PTCAccessPolicy,

-- Identifier Association events, see clauses 6.2.2.2.7 and 6.3.2.2.2

aMFIdentifierAssocation [62] AMFIdentifierAssocation,

mMEIdentifierAssocation [63] MMEIdentifierAssocation

}

IRITargetIdentifier ::= SEQUENCE

{

identifier [1] TargetIdentifier,

provenance [2] TargetIdentifierProvenance OPTIONAL

}

-- ==============

-- HI3 CC payload

-- ==============

CCPayload ::= SEQUENCE

{

cCPayloadOID [1] RELATIVE-OID,

pDU [2] CCPDU

}

CCPDU ::= CHOICE

{

uPFCCPDU [1] UPFCCPDU,

extendedUPFCCPDU [2] ExtendedUPFCCPDU,

mMSCCPDU [3] MMSCCPDU

}

-- ===========================

-- HI4 LI notification payload

-- ===========================

LINotificationPayload ::= SEQUENCE

{

lINotificationPayloadOID [1] RELATIVE-OID,

notification [2] LINotificationMessage

}

LINotificationMessage ::= CHOICE

{

lINotification [1] LINotification

}

-- ==================

-- 5G AMF definitions

-- ==================

-- See clause 6.2.2.2.2 for details of this structure

AMFRegistration ::= SEQUENCE

{

registrationType [1] AMFRegistrationType,

registrationResult [2] AMFRegistrationResult,

slice [3] Slice OPTIONAL,

sUPI [4] SUPI,

sUCI [5] SUCI OPTIONAL,

pEI [6] PEI OPTIONAL,

gPSI [7] GPSI OPTIONAL,

gUTI [8] FiveGGUTI,

location [9] Location OPTIONAL,

non3GPPAccessEndpoint [10] UEEndpointAddress OPTIONAL,

fiveGSTAIList [11] TAIList OPTIONAL

}

-- See clause 6.2.2.2.3 for details of this structure

AMFDeregistration ::= SEQUENCE

{

deregistrationDirection [1] AMFDirection,

accessType [2] AccessType,

sUPI [3] SUPI OPTIONAL,

sUCI [4] SUCI OPTIONAL,

pEI [5] PEI OPTIONAL,

gPSI [6] GPSI OPTIONAL,

gUTI [7] FiveGGUTI OPTIONAL,

cause [8] FiveGMMCause OPTIONAL,

location [9] Location OPTIONAL

}

-- See clause 6.2.2.2.4 for details of this structure

AMFLocationUpdate ::= SEQUENCE

{

sUPI [1] SUPI,

sUCI [2] SUCI OPTIONAL,

pEI [3] PEI OPTIONAL,

gPSI [4] GPSI OPTIONAL,

gUTI [5] FiveGGUTI OPTIONAL,

location [6] Location

}

-- See clause 6.2.2.2.5 for details of this structure

AMFStartOfInterceptionWithRegisteredUE ::= SEQUENCE

{

registrationResult [1] AMFRegistrationResult,

registrationType [2] AMFRegistrationType OPTIONAL,

slice [3] Slice OPTIONAL,

sUPI [4] SUPI,

sUCI [5] SUCI OPTIONAL,

pEI [6] PEI OPTIONAL,

gPSI [7] GPSI OPTIONAL,

gUTI [8] FiveGGUTI,

location [9] Location OPTIONAL,

non3GPPAccessEndpoint [10] UEEndpointAddress OPTIONAL,

timeOfRegistration [11] Timestamp OPTIONAL,

fiveGSTAIList [12] TAIList OPTIONAL

}

-- See clause 6.2.2.2.6 for details of this structure

AMFUnsuccessfulProcedure ::= SEQUENCE

{

failedProcedureType [1] AMFFailedProcedureType,

failureCause [2] AMFFailureCause,

requestedSlice [3] NSSAI OPTIONAL,

sUPI [4] SUPI OPTIONAL,

sUCI [5] SUCI OPTIONAL,

pEI [6] PEI OPTIONAL,

gPSI [7] GPSI OPTIONAL,

gUTI [8] FiveGGUTI OPTIONAL,

location [9] Location OPTIONAL

}

-- =================

-- 5G AMF parameters

-- =================

AMFID ::= SEQUENCE

{

aMFRegionID [1] AMFRegionID,

aMFSetID [2] AMFSetID,

aMFPointer [3] AMFPointer

}

AMFDirection ::= ENUMERATED

{

networkInitiated(1),

uEInitiated(2)

}

AMFFailedProcedureType ::= ENUMERATED

{

registration(1),

sMS(2),

pDUSessionEstablishment(3)

}

AMFFailureCause ::= CHOICE

{

fiveGMMCause [1] FiveGMMCause,

fiveGSMCause [2] FiveGSMCause

}

AMFPointer ::= INTEGER (0..63)

AMFRegistrationResult ::= ENUMERATED

{

threeGPPAccess(1),

nonThreeGPPAccess(2),

threeGPPAndNonThreeGPPAccess(3)

}

AMFRegionID ::= INTEGER (0..255)

AMFRegistrationType ::= ENUMERATED

{

initial(1),

mobility(2),

periodic(3),

emergency(4)

}

AMFSetID ::= INTEGER (0..1023)

-- ==================

-- 5G SMF definitions

-- ==================

-- See clause 6.2.3.2.2 for details of this structure

SMFPDUSessionEstablishment ::= SEQUENCE

{

sUPI [1] SUPI OPTIONAL,

sUPIUnauthenticated [2] SUPIUnauthenticatedIndication OPTIONAL,

pEI [3] PEI OPTIONAL,

gPSI [4] GPSI OPTIONAL,

pDUSessionID [5] PDUSessionID,

gTPTunnelID [6] FTEID,

pDUSessionType [7] PDUSessionType,

sNSSAI [8] SNSSAI OPTIONAL,

uEEndpoint [9] SEQUENCE OF UEEndpointAddress OPTIONAL,

non3GPPAccessEndpoint [10] UEEndpointAddress OPTIONAL,

location [11] Location OPTIONAL,

dNN [12] DNN,

aMFID [13] AMFID OPTIONAL,

hSMFURI [14] HSMFURI OPTIONAL,

requestType [15] FiveGSMRequestType,

accessType [16] AccessType OPTIONAL,

rATType [17] RATType OPTIONAL,

sMPDUDNRequest [18] SMPDUDNRequest OPTIONAL

}

-- See clause 6.2.3.2.3 for details of this structure

SMFPDUSessionModification ::= SEQUENCE

{

sUPI [1] SUPI OPTIONAL,

sUPIUnauthenticated [2] SUPIUnauthenticatedIndication OPTIONAL,

pEI [3] PEI OPTIONAL,

gPSI [4] GPSI OPTIONAL,

sNSSAI [5] SNSSAI OPTIONAL,

non3GPPAccessEndpoint [6] UEEndpointAddress OPTIONAL,

location [7] Location OPTIONAL,

requestType [8] FiveGSMRequestType,

accessType [9] AccessType OPTIONAL,

rATType [10] RATType OPTIONAL

}

-- See clause 6.2.3.2.4 for details of this structure

SMFPDUSessionRelease ::= SEQUENCE

{

sUPI [1] SUPI,

pEI [2] PEI OPTIONAL,

gPSI [3] GPSI OPTIONAL,

pDUSessionID [4] PDUSessionID,

timeOfFirstPacket [5] Timestamp OPTIONAL,

timeOfLastPacket [6] Timestamp OPTIONAL,

uplinkVolume [7] INTEGER OPTIONAL,

downlinkVolume [8] INTEGER OPTIONAL,

location [9] Location OPTIONAL

}

-- See clause 6.2.3.2.5 for details of this structure

SMFStartOfInterceptionWithEstablishedPDUSession ::= SEQUENCE

{

sUPI [1] SUPI OPTIONAL,

sUPIUnauthenticated [2] SUPIUnauthenticatedIndication OPTIONAL,

pEI [3] PEI OPTIONAL,

gPSI [4] GPSI OPTIONAL,

pDUSessionID [5] PDUSessionID,

gTPTunnelID [6] FTEID,

pDUSessionType [7] PDUSessionType,

sNSSAI [8] SNSSAI OPTIONAL,

uEEndpoint [9] SEQUENCE OF UEEndpointAddress,

non3GPPAccessEndpoint [10] UEEndpointAddress OPTIONAL,

location [11] Location OPTIONAL,

dNN [12] DNN,

aMFID [13] AMFID OPTIONAL,

hSMFURI [14] HSMFURI OPTIONAL,

requestType [15] FiveGSMRequestType,

accessType [16] AccessType OPTIONAL,

rATType [17] RATType OPTIONAL,

sMPDUDNRequest [18] SMPDUDNRequest OPTIONAL

}

-- See clause 6.2.3.2.6 for details of this structure

SMFUnsuccessfulProcedure ::= SEQUENCE

{

failedProcedureType [1] SMFFailedProcedureType,

failureCause [2] FiveGSMCause,

initiator [3] Initiator,

requestedSlice [4] NSSAI OPTIONAL,

sUPI [5] SUPI OPTIONAL,

sUPIUnauthenticated [6] SUPIUnauthenticatedIndication OPTIONAL,

pEI [7] PEI OPTIONAL,

gPSI [8] GPSI OPTIONAL,

pDUSessionID [9] PDUSessionID OPTIONAL,

uEEndpoint [10] SEQUENCE OF UEEndpointAddress OPTIONAL,

non3GPPAccessEndpoint [11] UEEndpointAddress OPTIONAL,

dNN [12] DNN OPTIONAL,

aMFID [13] AMFID OPTIONAL,

hSMFURI [14] HSMFURI OPTIONAL,

requestType [15] FiveGSMRequestType OPTIONAL,

accessType [16] AccessType OPTIONAL,

rATType [17] RATType OPTIONAL,

sMPDUDNRequest [18] SMPDUDNRequest OPTIONAL,

location [19] Location OPTIONAL

}

-- =================

-- 5G SMF parameters

-- =================

SMFFailedProcedureType ::= ENUMERATED

{

pDUSessionEstablishment(1),

pDUSessionModification(2),

pDUSessionRelease(3)

}

-- ==================

-- 5G UPF definitions

-- ==================

UPFCCPDU ::= OCTET STRING

-- See clause 6.2.3.8 for the details of this structure

ExtendedUPFCCPDU ::= SEQUENCE

{

payload [1] UPFCCPDUPayload,

qFI [2] QFI OPTIONAL

}

-- =================

-- 5G UPF parameters

-- =================

UPFCCPDUPayload ::= CHOICE

{

uPFIPCC [1] OCTET STRING,

uPFEthernetCC [2] OCTET STRING,

uPFUnstructuredCC [3] OCTET STRING

}

QFI ::= INTEGER (0..63)

-- ==================

-- 5G UDM definitions

-- ==================

UDMServingSystemMessage ::= SEQUENCE

{

sUPI [1] SUPI,

pEI [2] PEI OPTIONAL,

gPSI [3] GPSI OPTIONAL,

gUAMI [4] GUAMI OPTIONAL,

gUMMEI [5] GUMMEI OPTIONAL,

pLMNID [6] PLMNID OPTIONAL,

servingSystemMethod [7] UDMServingSystemMethod

}

-- =================

-- 5G UDM parameters

-- =================

UDMServingSystemMethod ::= ENUMERATED

{

amf3GPPAccessRegistration(0),

amfNon3GPPAccessRegistration(1),

unknown(2)

}

-- ===================

-- 5G SMSF definitions

-- ===================

-- See clause 6.2.5.3 for details of this structure

SMSMessage ::= SEQUENCE

{

originatingSMSParty [1] SMSParty,

terminatingSMSParty [2] SMSParty,

direction [3] Direction,

transferStatus [4] SMSTransferStatus,

otherMessage [5] SMSOtherMessageIndication OPTIONAL,

location [6] Location OPTIONAL,

peerNFAddress [7] SMSNFAddress OPTIONAL,

peerNFType [8] SMSNFType OPTIONAL,

sMSTPDUData [9] SMSTPDUData OPTIONAL

}

-- ==================

-- 5G SMSF parameters

-- ==================

SMSParty ::= SEQUENCE

{

sUPI [1] SUPI OPTIONAL,

pEI [2] PEI OPTIONAL,

gPSI [3] GPSI OPTIONAL

}

SMSTransferStatus ::= ENUMERATED

{

transferSucceeded(1),

transferFailed(2),

undefined(3)

}

SMSOtherMessageIndication ::= BOOLEAN

SMSNFAddress ::= CHOICE

{

iPAddress [1] IPAddress,

e164Number [2] E164Number

}

SMSNFType ::= ENUMERATED

{

sMSGMSC(1),

iWMSC(2),

sMSRouter(3)

}

SMSTPDUData ::= CHOICE

{

sMSTPDU [1] SMSTPDU

}

SMSTPDU ::= OCTET STRING (SIZE(1..270))

-- ===============

-- MMS definitions

-- ===============

MMSSend ::= SEQUENCE

{

transactionID [1] UTF8String,

version [2] MMSVersion,

dateTime [3] Timestamp,

originatingMMSParty [4] MMSParty,

terminatingMMSParty [5] SEQUENCE OF MMSParty OPTIONAL,

cCRecipients [6] SEQUENCE OF MMSParty OPTIONAL,

bCCRecipients [7] SEQUENCE OF MMSParty OPTIONAL,

direction [8] MMSDirection,

subject [9] MMSSubject OPTIONAL,

messageClass [10] MMSMessageClass OPTIONAL,

expiry [11] MMSExpiry,

desiredDeliveryTime [12] Timestamp OPTIONAL,

priority [13] MMSPriority OPTIONAL,

senderVisibility [14] BOOLEAN OPTIONAL,

deliveryReport [15] BOOLEAN OPTIONAL,

readReport [16] BOOLEAN OPTIONAL,

store [17] BOOLEAN OPTIONAL,

state [18] MMState OPTIONAL,

flags [19] MMFlags OPTIONAL,

replyCharging [20] MMSReplyCharging OPTIONAL,

applicID [21] UTF8String OPTIONAL,

replyApplicID [22] UTF8String OPTIONAL,

auxApplicInfo [23] UTF8String OPTIONAL,

contentClass [24] MMSContentClass OPTIONAL,

dRMContent [25] BOOLEAN OPTIONAL,

adaptationAllowed [26] MMSAdaptation OPTIONAL,

contentType [27] MMSContentType,

responseStatus [28] MMSResponseStatus,

responseStatusText [29] UTF8String OPTIONAL,

messageID [30] UTF8String

}

MMSSendByNonLocalTarget ::= SEQUENCE

{

version [1] MMSVersion,

transactionID [2] UTF8String,

messageID [3] UTF8String,

terminatingMMSParty [4] SEQUENCE OF MMSParty,

originatingMMSParty [5] MMSParty,

direction [6] MMSDirection,

contentType [7] MMSContentType,

messageClass [8] MMSMessageClass OPTIONAL,

dateTime [9] Timestamp,

expiry [10] MMSExpiry OPTIONAL,

deliveryReport [11] BOOLEAN OPTIONAL,

priority [12] MMSPriority OPTIONAL,

senderVisibility [13] BOOLEAN OPTIONAL,

readReport [14] BOOLEAN OPTIONAL,

subject [15] MMSSubject OPTIONAL,

forwardCount [16] INTEGER OPTIONAL,

previouslySentBy [17] MMSPreviouslySentBy OPTIONAL,

prevSentByDateTime [18] Timestamp OPTIONAL,

applicID [19] UTF8String OPTIONAL,

replyApplicID [20] UTF8String OPTIONAL,

auxApplicInfo [21] UTF8String OPTIONAL,

contentClass [22] MMSContentClass OPTIONAL,

dRMContent [23] BOOLEAN OPTIONAL,

adaptationAllowed [24] MMSAdaptation OPTIONAL

}

MMSNotification ::= SEQUENCE

{

transactionID [1] UTF8String,

version [2] MMSVersion,

originatingMMSParty [3] MMSParty OPTIONAL,

direction [4] MMSDirection,

subject [5] MMSSubject OPTIONAL,

deliveryReportRequested [6] BOOLEAN OPTIONAL,

stored [7] BOOLEAN OPTIONAL,

messageClass [8] MMSMessageClass,

priority [9] MMSPriority OPTIONAL,

messageSize [10] INTEGER,

expiry [11] MMSExpiry,

replyCharging [12] MMSReplyCharging OPTIONAL

}

MMSSendToNonLocalTarget ::= SEQUENCE

{

version [1] MMSVersion,

transactionID [2] UTF8String,

messageID [3] UTF8String,

terminatingMMSParty [4] SEQUENCE OF MMSParty,

originatingMMSParty [5] MMSParty,

direction [6] MMSDirection,

contentType [7] MMSContentType,

messageClass [8] MMSMessageClass OPTIONAL,

dateTime [9] Timestamp,

expiry [10] MMSExpiry OPTIONAL,

deliveryReport [11] BOOLEAN OPTIONAL,

priority [12] MMSPriority OPTIONAL,

senderVisibility [13] BOOLEAN OPTIONAL,

readReport [14] BOOLEAN OPTIONAL,

subject [15] MMSSubject OPTIONAL,

forwardCount [16] INTEGER OPTIONAL,

previouslySentBy [17] MMSPreviouslySentBy OPTIONAL,

prevSentByDateTime [18] Timestamp OPTIONAL,

applicID [19] UTF8String OPTIONAL,

replyApplicID [20] UTF8String OPTIONAL,

auxApplicInfo [21] UTF8String OPTIONAL,

contentClass [22] MMSContentClass OPTIONAL,

dRMContent [23] BOOLEAN OPTIONAL,

adaptationAllowed [24] MMSAdaptation OPTIONAL

}

MMSNotificationResponse ::= SEQUENCE

{

transactionID [1] UTF8String,

version [2] MMSVersion,

direction [3] MMSDirection,

status [4] MMStatus,

reportAllowed [5] BOOLEAN OPTIONAL

}

MMSRetrieval ::= SEQUENCE

{

transactionID [1] UTF8String,

version [2] MMSVersion,

messageID [3] UTF8String,

dateTime [4] Timestamp,

originatingMMSParty [5] MMSParty OPTIONAL,

previouslySentBy [6] MMSPreviouslySentBy OPTIONAL,

prevSentByDateTime [7] Timestamp OPTIONAL,

terminatingMMSParty [8] SEQUENCE OF MMSParty OPTIONAL,

cCRecipients [9] SEQUENCE OF MMSParty OPTIONAL,

direction [10] MMSDirection,

subject [11] MMSSubject OPTIONAL,

state [12] MMState OPTIONAL,

flags [13] MMFlags OPTIONAL,

messageClass [14] MMSMessageClass OPTIONAL,

priority [15] MMSPriority,

deliveryReport [16] BOOLEAN OPTIONAL,

readReport [17] BOOLEAN OPTIONAL,

replyCharging [18] MMSReplyCharging OPTIONAL,

retrieveStatus [19] MMSRetrieveStatus OPTIONAL,

retrieveStatusText [20] UTF8String OPTIONAL,

applicID [21] UTF8String OPTIONAL,

replyApplicID [22] UTF8String OPTIONAL,

auxApplicInfo [23] UTF8String OPTIONAL,

contentClass [24] MMSContentClass OPTIONAL,

dRMContent [25] BOOLEAN OPTIONAL,

replaceID [26] UTF8String OPTIONAL,

contentType [27] UTF8String OPTIONAL

}

MMSDeliveryAck ::= SEQUENCE

{

transactionID [1] UTF8String,

version [2] MMSVersion,

reportAllowed [3] BOOLEAN OPTIONAL,

status [4] MMStatus,

direction [5] MMSDirection

}

MMSForward ::= SEQUENCE

{

transactionID [1] UTF8String,

version [2] MMSVersion,

dateTime [3] Timestamp OPTIONAL,

originatingMMSParty [4] MMSParty,

terminatingMMSParty [5] SEQUENCE OF MMSParty OPTIONAL,

cCRecipients [6] SEQUENCE OF MMSParty OPTIONAL,

bCCRecipients [7] SEQUENCE OF MMSParty OPTIONAL,

direction [8] MMSDirection,

expiry [9] MMSExpiry OPTIONAL,

desiredDeliveryTime [10] Timestamp OPTIONAL,

deliveryReportAllowed [11] BOOLEAN OPTIONAL,

deliveryReport [12] BOOLEAN OPTIONAL,

store [13] BOOLEAN OPTIONAL,

state [14] MMState OPTIONAL,

flags [15] MMFlags OPTIONAL,

contentLocationReq [16] UTF8String,

replyCharging [17] MMSReplyCharging OPTIONAL,

responseStatus [18] MMSResponseStatus,

responseStatusText [19] UTF8String OPTIONAL,

messageID [20] UTF8String OPTIONAL,

contentLocationConf [21] UTF8String OPTIONAL,

storeStatus [22] MMSStoreStatus OPTIONAL,

storeStatusText [23] UTF8String OPTIONAL

}

MMSDeleteFromRelay ::= SEQUENCE

{

transactionID [1] UTF8String,

version [2] MMSVersion,

direction [3] MMSDirection,

contentLocationReq [4] SEQUENCE OF UTF8String,

contentLocationConf [5] SEQUENCE OF UTF8String,

deleteResponseStatus [6] MMSDeleteResponseStatus,

deleteResponseText [7] SEQUENCE OF UTF8String

}

MMSMBoxStore ::= SEQUENCE

{

transactionID [1] UTF8String,

version [2] MMSVersion,

direction [3] MMSDirection,

contentLocationReq [4] UTF8String,

state [5] MMState OPTIONAL,

flags [6] MMFlags OPTIONAL,

contentLocationConf [7] UTF8String OPTIONAL,

storeStatus [8] MMSStoreStatus,

storeStatusText [9] UTF8String OPTIONAL

}

MMSMBoxUpload ::= SEQUENCE

{

transactionID [1] UTF8String,

version [2] MMSVersion,

direction [3] MMSDirection,

state [4] MMState OPTIONAL,

flags [5] MMFlags OPTIONAL,

contentType [6] UTF8String,

contentLocation [7] UTF8String OPTIONAL,

storeStatus [8] MMSStoreStatus,

storeStatusText [9] UTF8String OPTIONAL,

mMessages [10] SEQUENCE OF MMBoxDescription

}

MMSMBoxDelete ::= SEQUENCE

{

transactionID [1] UTF8String,

version [2] MMSVersion,

direction [3] MMSDirection,

contentLocationReq [4] SEQUENCE OF UTF8String,

contentLocationConf [5] SEQUENCE OF UTF8String OPTIONAL,

responseStatus [6] MMSDeleteResponseStatus,

responseStatusText [7] UTF8String OPTIONAL

}

MMSDeliveryReport ::= SEQUENCE

{

version [1] MMSVersion,

messageID [2] UTF8String,

terminatingMMSParty [3] SEQUENCE OF MMSParty,

mMSDateTime [4] Timestamp,

responseStatus [5] MMSResponseStatus,

responseStatusText [6] UTF8String OPTIONAL,

applicID [7] UTF8String OPTIONAL,

replyApplicID [8] UTF8String OPTIONAL,

auxApplicInfo [9] UTF8String OPTIONAL

}

MMSDeliveryReportNonLocalTarget ::= SEQUENCE

{

version [1] MMSVersion,

transactionID [2] UTF8String,

messageID [3] UTF8String,

terminatingMMSParty [4] SEQUENCE OF MMSParty,

originatingMMSParty [5] MMSParty,

direction [6] MMSDirection,

mMSDateTime [7] Timestamp,

forwardToOriginator [8] BOOLEAN OPTIONAL,

status [9] MMStatus,

statusExtension [10] MMStatusExtension,

statusText [11] MMStatusText,

applicID [12] UTF8String OPTIONAL,

replyApplicID [13] UTF8String OPTIONAL,

auxApplicInfo [14] UTF8String OPTIONAL

}

MMSReadReport ::= SEQUENCE

{

version [1] MMSVersion,

messageID [2] UTF8String,

terminatingMMSParty [3] SEQUENCE OF MMSParty,

originatingMMSParty [4] SEQUENCE OF MMSParty,

direction [5] MMSDirection,

mMSDateTime [6] Timestamp,

readStatus [7] MMSReadStatus,

applicID [8] UTF8String OPTIONAL,

replyApplicID [9] UTF8String OPTIONAL,

auxApplicInfo [10] UTF8String OPTIONAL

}

MMSReadReportNonLocalTarget ::= SEQUENCE

{

version [1] MMSVersion,

transactionID [2] UTF8String,

terminatingMMSParty [3] SEQUENCE OF MMSParty,

originatingMMSParty [4] SEQUENCE OF MMSParty,

direction [5] MMSDirection,

messageID [6] UTF8String,

mMSDateTime [7] Timestamp,

readStatus [8] MMSReadStatus,

readStatusText [9] MMSReadStatusText OPTIONAL,

applicID [10] UTF8String OPTIONAL,

replyApplicID [11] UTF8String OPTIONAL,

auxApplicInfo [12] UTF8String OPTIONAL

}

MMSCancel ::= SEQUENCE

{

transactionID [1] UTF8String,

version [2] MMSVersion,

cancelID [3] UTF8String,

direction [4] MMSDirection

}

MMSMBoxViewRequest ::= SEQUENCE

{

transactionID [1] UTF8String,

version [2] MMSVersion,

contentLocation [3] UTF8String OPTIONAL,

state [4] SEQUENCE OF MMState OPTIONAL,

flags [5] SEQUENCE OF MMFlags OPTIONAL,

start [6] INTEGER OPTIONAL,

limit [7] INTEGER OPTIONAL,

attributes [8] SEQUENCE OF UTF8String OPTIONAL,

totals [9] INTEGER OPTIONAL,

quotas [10] MMSQuota OPTIONAL

}

MMSMBoxViewResponse ::= SEQUENCE

{

transactionID [1] UTF8String,

version [2] MMSVersion,

contentLocation [3] UTF8String OPTIONAL,

state [4] SEQUENCE OF MMState OPTIONAL,

flags [5] SEQUENCE OF MMFlags OPTIONAL,

start [6] INTEGER OPTIONAL,

limit [7] INTEGER OPTIONAL,

attributes [8] SEQUENCE OF UTF8String OPTIONAL,

mMSTotals [9] BOOLEAN OPTIONAL,

mMSQuotas [10] BOOLEAN OPTIONAL,

mMessages [11] SEQUENCE OF MMBoxDescription

}

MMBoxDescription ::= SEQUENCE

{

contentLocation [1] UTF8String OPTIONAL,

messageID [2] UTF8String OPTIONAL,

state [3] MMState OPTIONAL,

flags [4] SEQUENCE OF MMFlags OPTIONAL,

dateTime [5] Timestamp OPTIONAL,

originatingMMSParty [6] MMSParty OPTIONAL,

terminatingMMSParty [7] SEQUENCE OF MMSParty OPTIONAL,

cCRecipients [8] SEQUENCE OF MMSParty OPTIONAL,

bCCRecipients [9] SEQUENCE OF MMSParty OPTIONAL,

messageClass [10] MMSMessageClass OPTIONAL,

subject [11] MMSSubject OPTIONAL,

priority [12] MMSPriority OPTIONAL,

deliveryTime [13] Timestamp OPTIONAL,

readReport [14] BOOLEAN OPTIONAL,

messageSize [15] INTEGER OPTIONAL,

replyCharging [16] MMSReplyCharging OPTIONAL,

previouslySentBy [17] MMSPreviouslySentBy OPTIONAL,

previouslySentByDateTime [18] Timestamp OPTIONAL,

contentType [19] UTF8String OPTIONAL

}

-- =========

-- MMS CCPDU

-- =========

MMSCCPDU ::= SEQUENCE

{

version [1] MMSVersion,

transactionID [2] UTF8String,

mMSContent [3] OCTET STRING

}

-- ==============

-- MMS parameters

-- ==============

MMSAdaptation ::= SEQUENCE

{

allowed [1] BOOLEAN,

overriden [2] BOOLEAN

}

MMSCancelStatus ::= ENUMERATED

{

cancelRequestSuccessfullyReceived(1),

cancelRequestCorrupted(2)

}

MMSContentClass ::= ENUMERATED

{

text(1),

imageBasic(2),

imageRich(3),

videoBasic(4),

videoRich(5),

megaPixel(6),

contentBasic(7),

contentRich(8)

}

MMSContentType ::= UTF8String

MMSDeleteResponseStatus ::= ENUMERATED

{

ok(1),

errorUnspecified(2),

errorServiceDenied(3),

errorMessageFormatCorrupt(4),

errorSendingAddressUnresolved(5),

errorMessageNotFound(6),

errorNetworkProblem(7),

errorContentNotAccepted(8),

errorUnsupportedMessage(9),

errorTransientFailure(10),

errorTransientSendingAddressUnresolved(11),

errorTransientMessageNotFound(12),

errorTransientNetworkProblem(13),

errorTransientPartialSuccess(14),

errorPermanentFailure(15),

errorPermanentServiceDenied(16),

errorPermanentMessageFormatCorrupt(17),

errorPermanentSendingAddressUnresolved(18),

errorPermanentMessageNotFound(19),

errorPermanentContentNotAccepted(20),

errorPermanentReplyChargingLimitationsNotMet(21),

errorPermanentReplyChargingRequestNotAccepted(22),

errorPermanentReplyChargingForwardingDenied(23),

errorPermanentReplyChargingNotSupported(24),

errorPermanentAddressHidingNotSupported(25),

errorPermanentLackOfPrepaid(26)

}

MMSDirection ::= ENUMERATED

{

fromTarget(0),

toTarget(1)

}

MMSElementDescriptor ::= SEQUENCE

{

reference [1] UTF8String,

parameter [2] UTF8String OPTIONAL,

value [3] UTF8String OPTIONAL

}

MMSExpiry ::= SEQUENCE

{

expiryPeriod [1] INTEGER,

periodFormat [2] MMSPeriodFormat

}

MMFlags ::= SEQUENCE

{

length [1] INTEGER,

flag [2] MMStateFlag,

flagString [3] UTF8String

}

MMSMessageClass ::= ENUMERATED

{

personal(1),

advertisement(2),

informational(3),

auto(4)

}

MMSParty ::= SEQUENCE

{

mMSPartyIDs [1] SEQUENCE OF MMSPartyID,

nonLocalID [2] NonLocalID

}

MMSPartyID ::= CHOICE

{

e164Number [1] E164Number,

emailAddress [2] EmailAddress,

iMSI [3] IMSI,

iMPU [4] IMPU,

iMPI [5] IMPI,

sUPI [6] SUPI,

gPSI [7] GPSI

}

MMSPeriodFormat ::= ENUMERATED

{

absolute(1),

relative(2)

}

MMSPreviouslySent ::= SEQUENCE

{

previouslySentByParty [1] MMSParty,

sequenceNumber [2] INTEGER,

previousSendDateTime [3] Timestamp

}

MMSPreviouslySentBy ::= SEQUENCE OF MMSPreviouslySent

MMSPriority ::= ENUMERATED

{

low(1),

normal(2),

high(3)

}

MMSQuota ::= SEQUENCE

{

quota [1] INTEGER,

quotaUnit [2] MMSQuotaUnit

}

MMSQuotaUnit ::= ENUMERATED

{

numMessages(1),

bytes(2)

}

MMSReadStatus ::= ENUMERATED

{

read(1),

deletedWithoutBeingRead(2)

}

MMSReadStatusText ::= UTF8String

MMSReplyCharging ::= ENUMERATED

{

requested(0),

requestedTextOnly(1),

accepted(2),

acceptedTextOnly(3)

}

MMSResponseStatus ::= ENUMERATED

{

ok(1),

errorUnspecified(2),

errorServiceDenied(3),

errorMessageFormatCorrupt(4),

errorSendingAddressUnresolved(5),

errorMessageNotFound(6),

errorNetworkProblem(7),

errorContentNotAccepted(8),

errorUnsupportedMessage(9),

errorTransientFailure(10),

errorTransientSendingAddressUnresolved(11),

errorTransientMessageNotFound(12),

errorTransientNetworkProblem(13),

errorTransientPartialSuccess(14),

errorPermanentFailure(15),

errorPermanentServiceDenied(16),

errorPermanentMessageFormatCorrupt(17),

errorPermanentSendingAddressUnresolved(18),

errorPermanentMessageNotFound(19),

errorPermanentContentNotAccepted(20),

errorPermanentReplyChargingLimitationsNotMet(21),

errorPermanentReplyChargingRequestNotAccepted(22),

errorPermanentReplyChargingForwardingDenied(23),

errorPermanentReplyChargingNotSupported(24),

errorPermanentAddressHidingNotSupported(25),

errorPermanentLackOfPrepaid(26)

}

MMSRetrieveStatus ::= ENUMERATED

{

success(1),

errorTransientFailure(2),

errorTransientMessageNotFound(3),

errorTransientNetworkProblem(4),

errorPermanentFailure(5),

errorPermanentServiceDenied(6),

errorPermanentMessageNotFound(7),

errorPermanentContentUnsupported(8)

}

MMSStoreStatus ::= ENUMERATED

{

success(1),

errorTransientFailure(2),

errorTransientNetworkProblem(3),

errorPermanentFailure(4),

errorPermanentServiceDenied(5),

errorPermanentMessageFormatCorrupt(6),

errorPermanentMessageNotFound(7),

errorMMBoxFull(8)

}

MMState ::= ENUMERATED

{

draft(1),

sent(2),

new(3),

retrieved(4),

forwarded(5)

}

MMStateFlag ::= ENUMERATED

{

add(1),

remove(2),

filter(3)

}

MMStatus ::= ENUMERATED

{

expired(1),

retrieved(2),

rejected(3),

deferred(4),

unrecognized(5),

indeterminate(6),

forwarded(7),

unreachable(8)

}

MMStatusExtension ::= ENUMERATED

{

rejectionByMMSRecipient(0),

rejectionByOtherRS(1)

}

MMStatusText ::= UTF8String

MMSSubject ::= UTF8String

MMSVersion ::= SEQUENCE

{

majorVersion [1] INTEGER,

minorVersion [2] INTEGER

}

-- ==================

-- 5G PTC definitions

-- ==================

PTCRegistration ::= SEQUENCE

{

pTCTargetInformation [1] PTCTargetInformation,

pTCServerURI [2] UTF8String,

pTCRegistrationRequest [3] PTCRegistrationRequest,

pTCRegistrationOutcome [4] PTCRegistrationOutcome

}

PTCSessionInitiation ::= SEQUENCE

{

pTCTargetInformation [1] PTCTargetInformation,

pTCDirection [2] Direction,

pTCServerURI [3] UTF8String,

pTCSessionInfo [4] PTCSessionInfo,

pTCOriginatingID [5] PTCTargetInformation,

pTCParticipants [6] SEQUENCE OF PTCTargetInformation OPTIONAL,

pTCParticipantPresenceStatus [7] MultipleParticipantPresenceStatus OPTIONAL,

location [8] Location OPTIONAL,

pTCBearerCapability [9] UTF8String OPTIONAL,

pTCHost [10] PTCTargetInformation OPTIONAL

}

PTCSessionAbandon ::= SEQUENCE

{

pTCTargetInformation [1] PTCTargetInformation,

pTCDirection [2] Direction,

pTCSessionInfo [3] PTCSessionInfo,

location [4] Location OPTIONAL,

pTCAbandonCause [5] INTEGER

}

PTCSessionStart ::= SEQUENCE

{

pTCTargetInformation [1] PTCTargetInformation,

pTCDirection [2] Direction,

pTCServerURI [3] UTF8String,

pTCSessionInfo [4] PTCSessionInfo,

pTCOriginatingID [5] PTCTargetInformation,

pTCParticipants [6] SEQUENCE OF PTCTargetInformation OPTIONAL,

pTCParticipantPresenceStatus [7] MultipleParticipantPresenceStatus OPTIONAL,

location [8] Location OPTIONAL,

pTCHost [9] PTCTargetInformation OPTIONAL,

pTCBearerCapability [10] UTF8String OPTIONAL

}

PTCSessionEnd ::= SEQUENCE

{

pTCTargetInformation [1] PTCTargetInformation,

pTCDirection [2] Direction,

pTCServerURI [3] UTF8String,

pTCSessionInfo [4] PTCSessionInfo,

pTCParticipants [5] SEQUENCE OF PTCTargetInformation OPTIONAL,

location [6] Location OPTIONAL,

pTCSessionEndCause [7] PTCSessionEndCause

}

PTCStartOfInterception ::= SEQUENCE

{

pTCTargetInformation [1] PTCTargetInformation,

pTCDirection [2] Direction,

preEstSessionID [3] PTCSessionInfo OPTIONAL,

pTCOriginatingID [4] PTCTargetInformation,

pTCSessionInfo [5] PTCSessionInfo OPTIONAL,

pTCHost [6] PTCTargetInformation OPTIONAL,

pTCParticipants [7] SEQUENCE OF PTCTargetInformation OPTIONAL,

pTCMediaStreamAvail [8] BOOLEAN OPTIONAL,

pTCBearerCapability [9] UTF8String OPTIONAL

}

PTCPreEstablishedSession ::= SEQUENCE

{

pTCTargetInformation [1] PTCTargetInformation,

pTCServerURI [2] UTF8String,

rTPSetting [3] RTPSetting,

pTCMediaCapability [4] UTF8String,

pTCPreEstSessionID [5] PTCSessionInfo,

pTCPreEstStatus [6] PTCPreEstStatus,

pTCMediaStreamAvail [7] BOOLEAN OPTIONAL,

location [8] Location OPTIONAL,

pTCFailureCode [9] PTCFailureCode OPTIONAL

}

PTCInstantPersonalAlert ::= SEQUENCE

{

pTCTargetInformation [1] PTCTargetInformation,

pTCIPAPartyID [2] PTCTargetInformation,

pTCIPADirection [3] Direction

}

PTCPartyJoin ::= SEQUENCE

{

pTCTargetInformation [1] PTCTargetInformation,

pTCDirection [2] Direction,

pTCSessionInfo [3] PTCSessionInfo,

pTCParticipants [4] SEQUENCE OF PTCTargetInformation OPTIONAL,

pTCParticipantPresenceStatus [5] MultipleParticipantPresenceStatus OPTIONAL,

pTCMediaStreamAvail [6] BOOLEAN OPTIONAL,

pTCBearerCapability [7] UTF8String OPTIONAL

}

PTCPartyDrop ::= SEQUENCE

{

pTCTargetInformation [1] PTCTargetInformation,

pTCDirection [2] Direction,

pTCSessionInfo [3] PTCSessionInfo,

pTCPartyDrop [4] PTCTargetInformation,

pTCParticipantPresenceStatus [5] PTCParticipantPresenceStatus OPTIONAL

}

PTCPartyHold ::= SEQUENCE

{

pTCTargetInformation [1] PTCTargetInformation,

pTCDirection [2] Direction,

pTCSessionInfo [3] PTCSessionInfo,

pTCParticipants [4] SEQUENCE OF PTCTargetInformation OPTIONAL,

pTCHoldID [5] SEQUENCE OF PTCTargetInformation,

pTCHoldRetrieveInd [6] BOOLEAN

}

PTCMediaModification ::= SEQUENCE

{

pTCTargetInformation [1] PTCTargetInformation,

pTCDirection [2] Direction,

pTCSessionInfo [3] PTCSessionInfo,

pTCMediaStreamAvail [4] BOOLEAN OPTIONAL,

pTCBearerCapability [5] UTF8String

}

PTCGroupAdvertisement ::=SEQUENCE

{

pTCTargetInformation [1] PTCTargetInformation,

pTCDirection [2] Direction,

pTCIDList [3] SEQUENCE OF PTCTargetInformation OPTIONAL,

pTCGroupAuthRule [4] PTCGroupAuthRule OPTIONAL,

pTCGroupAdSender [5] PTCTargetInformation,

pTCGroupNickname [6] UTF8String OPTIONAL

}

PTCFloorControl ::= SEQUENCE

{

pTCTargetInformation [1] PTCTargetInformation,

pTCDirection [2] Direction,

pTCSessioninfo [3] PTCSessionInfo,

pTCFloorActivity [4] SEQUENCE OF PTCFloorActivity,

pTCFloorSpeakerID [5] PTCTargetInformation OPTIONAL,

pTCMaxTBTime [6] INTEGER OPTIONAL,

pTCQueuedFloorControl [7] BOOLEAN OPTIONAL,

pTCQueuedPosition [8] INTEGER OPTIONAL,

pTCTalkBurstPriority [9] PTCTBPriorityLevel OPTIONAL,

pTCTalkBurstReason [10] PTCTBReasonCode OPTIONAL

}

PTCTargetPresence ::= SEQUENCE

{

pTCTargetInformation [1] PTCTargetInformation,

pTCTargetPresenceStatus [2] PTCParticipantPresenceStatus

}

PTCParticipantPresence ::= SEQUENCE

{

pTCTargetInformation [1] PTCTargetInformation,

pTCParticipantPresenceStatus [2] PTCParticipantPresenceStatus

}

PTCListManagement ::= SEQUENCE

{

pTCTargetInformation [1] PTCTargetInformation,

pTCDirection [2] Direction,

pTCListManagementType [3] PTCListManagementType OPTIONAL,

pTCListManagementAction [4] PTCListManagementAction OPTIONAL,

pTCListManagementFailure [5] PTCListManagementFailure OPTIONAL,

pTCContactID [6] PTCTargetInformation OPTIONAL,

pTCIDList [7] SEQUENCE OF PTCIDList OPTIONAL,

pTCHost [8] PTCTargetInformation OPTIONAL

}

PTCAccessPolicy ::= SEQUENCE

{

pTCTargetInformation [1] PTCTargetInformation,

pTCDirection [2] Direction,

pTCAccessPolicyType [3] PTCAccessPolicyType OPTIONAL,

pTCUserAccessPolicy [4] PTCUserAccessPolicy OPTIONAL,

pTCGroupAuthRule [5] PTCGroupAuthRule OPTIONAL,

pTCContactID [6] PTCTargetInformation OPTIONAL,

pTCAccessPolicyFailure [7] PTCAccessPolicyFailure OPTIONAL

}

-- =================

-- 5G PTC parameters

-- =================

PTCRegistrationRequest ::= ENUMERATED

{

register(1),

reRegister(2),

deRegister(3)

}

PTCRegistrationOutcome ::= ENUMERATED

{

success(1),

failure(2)

}

PTCSessionEndCause ::= ENUMERATED

{

initiaterLeavesSession(1),

definedParticipantLeaves(2),

numberOfParticipants(3),

sessionTimerExpired(4),

pTCSpeechInactive(5),

allMediaTypesInactive(6)

}

PTCTargetInformation ::= SEQUENCE

{

identifiers [1] SEQUENCE SIZE(1..MAX) OF PTCIdentifiers

}

PTCIdentifiers ::= CHOICE

{

mCPTTID [1] UTF8String,

instanceIdentifierURN [2] UTF8String,

pTCChatGroupID [3] PTCChatGroupID,

iMPU [4] IMPU,

iMPI [5] IMPI

}

PTCSessionInfo ::= SEQUENCE

{

pTCSessionURI [1] UTF8String,

pTCSessionType [2] PTCSessionType

}

PTCSessionType ::= ENUMERATED

{

ondemand(1),

preEstablished(2),

adhoc(3),

prearranged(4),

groupSession(5)

}

MultipleParticipantPresenceStatus ::= SEQUENCE OF PTCParticipantPresenceStatus

PTCParticipantPresenceStatus ::= SEQUENCE

{

presenceID [1] PTCTargetInformation,

presenceType [2] PTCPresenceType,

presenceStatus [3] BOOLEAN

}

PTCPresenceType ::= ENUMERATED

{

pTCClient(1),

pTCGroup(2)

}

PTCPreEstStatus ::= ENUMERATED

{

established(1),

modified(2),

released(3)

}

RTPSetting ::= SEQUENCE

{

iPAddress [1] IPAddress,

portNumber [2] PortNumber

}

PTCIDList ::= SEQUENCE

{

pTCPartyID [1] PTCTargetInformation,

pTCChatGroupID [2] PTCChatGroupID

}

PTCChatGroupID ::= SEQUENCE

{

groupIdentity [1] UTF8String

}

PTCFloorActivity ::= ENUMERATED

{

tBCPRequest(1),

tBCPGranted(2),

tBCPDeny(3),

tBCPIdle(4),

tBCPTaken(5),

tBCPRevoke(6),

tBCPQueued(7),

tBCPRelease(8)

}

PTCTBPriorityLevel ::= ENUMERATED

{

preEmptive(1),

highPriority(2),

normalPriority(3),

listenOnly(4)

}

PTCTBReasonCode ::= ENUMERATED

{

noQueuingAllowed(1),

oneParticipantSession(2),

listenOnly(3),

exceededMaxDuration(4),

tBPrevented(5)

}

PTCListManagementType ::= ENUMERATED

{

contactListManagementAttempt(1),

groupListManagementAttempt(2),

contactListManagementResult(3),

groupListManagementResult(4),

requestUnsuccessful(5)

}

PTCListManagementAction ::= ENUMERATED

{

create(1),

modify(2),

retrieve(3),

delete(4),

notify(5)

}

PTCAccessPolicyType ::= ENUMERATED

{

pTCUserAccessPolicyAttempt(1),

groupAuthorizationRulesAttempt(2),

pTCUserAccessPolicyQuery(3),

groupAuthorizationRulesQuery(4),

pTCUserAccessPolicyResult(5),

groupAuthorizationRulesResult(6),

requestUnsuccessful(7)

}

PTCUserAccessPolicy ::= ENUMERATED

{

allowIncomingPTCSessionRequest(1),

blockIncomingPTCSessionRequest(2),

allowAutoAnswerMode(3),

allowOverrideManualAnswerMode(4)

}

PTCGroupAuthRule ::= ENUMERATED

{

allowInitiatingPTCSession(1),

blockInitiatingPTCSession(2),

allowJoiningPTCSession(3),

blockJoiningPTCSession(4),

allowAddParticipants(5),

blockAddParticipants(6),

allowSubscriptionPTCSessionState(7),

blockSubscriptionPTCSessionState(8),

allowAnonymity(9),

forbidAnonymity(10)

}

PTCFailureCode ::= ENUMERATED

{

sessionCannotBeEstablished(1),

sessionCannotBeModified(2)

}

PTCListManagementFailure ::= ENUMERATED

{

requestUnsuccessful(1),

requestUnknown(2)

}

PTCAccessPolicyFailure ::= ENUMERATED

{

requestUnsuccessful(1),

requestUnknown(2)

}

-- ===================

-- 5G LALS definitions

-- ===================

LALSReport ::= SEQUENCE

{

sUPI [1] SUPI OPTIONAL,

pEI [2] PEI OPTIONAL,

gPSI [3] GPSI OPTIONAL,

location [4] Location OPTIONAL

}

-- =====================

-- PDHR/PDSR definitions

-- =====================

PDHeaderReport ::= SEQUENCE

{

pDUSessionID [1] PDUSessionID,

sourceIPAddress [2] IPAddress,

sourcePort [3] PortNumber OPTIONAL,

destinationIPAddress [4] IPAddress,

destinationPort [5] PortNumber OPTIONAL,

nextLayerProtocol [6] NextLayerProtocol,

iPv6flowLabel [7] IPv6FlowLabel OPTIONAL,

direction [8] Direction,

packetSize [9] INTEGER

}

PDSummaryReport ::= SEQUENCE

{

pDUSessionID [1] PDUSessionID,

sourceIPAddress [2] IPAddress,

sourcePort [3] PortNumber OPTIONAL,

destinationIPAddress [4] IPAddress,

destinationPort [5] PortNumber OPTIONAL,

nextLayerProtocol [6] NextLayerProtocol,

iPv6flowLabel [7] IPv6FlowLabel OPTIONAL,

direction [8] Direction,

pDSRSummaryTrigger [9] PDSRSummaryTrigger,

firstPacketTimestamp [10] Timestamp,

lastPacketTimestamp [11] Timestamp,

packetCount [12] INTEGER,

byteCount [13] INTEGER

}

-- ====================

-- PDHR/PDSR parameters

-- ====================

PDSRSummaryTrigger ::= ENUMERATED

{

timerExpiry(1),

packetCount(2),

byteCount(3)

}

-- ==================================

-- Identifier Association definitions

-- ==================================

AMFIdentifierAssocation ::= SEQUENCE

{

sUPI [1] SUPI,

sUCI [2] SUCI OPTIONAL,

pEI [3] PEI OPTIONAL,

gPSI [4] GPSI OPTIONAL,

gUTI [5] FiveGGUTI,

location [6] Location,

fiveGSTAIList [7] TAIList OPTIONAL

}

MMEIdentifierAssocation ::= SEQUENCE

{

iMSI [1] IMSI,

iMEI [2] IMEI OPTIONAL,

mSISDN [3] MSISDN OPTIONAL,

gUTI [4] GUTI,

location [5] Location,

tAIList [6] TAIList OPTIONAL

}

-- =================================

-- Identifier Association parameters

-- =================================

GUTI ::= SEQUENCE

{

mCC [1] MCC,

mNC [2] MNC,

mMEGroupID [3] MMEGroupID,

mMECode [4] MMECode,

mTMSI [5] TMSI

}

MMEGroupID ::= OCTET STRING (SIZE(2))

MMECode ::= OCTET STRING (SIZE(1))

TMSI ::= OCTET STRING (SIZE(4))

-- ===========================

-- LI Notification definitions

-- ===========================

LINotification ::= SEQUENCE

{

notificationType [1] LINotificationType,

appliedTargetID [2] TargetIdentifier OPTIONAL,

appliedDeliveryInformation [3] SEQUENCE OF LIAppliedDeliveryInformation OPTIONAL,

appliedStartTime [4] Timestamp OPTIONAL,

appliedEndTime [5] Timestamp OPTIONAL

}

-- ==========================

-- LI Notification parameters

-- ==========================

LINotificationType ::= ENUMERATED

{

activation(1),

deactivation(2),

modification(3)

}

LIAppliedDeliveryInformation ::= SEQUENCE

{

hI2DeliveryIPAddress [1] IPAddress OPTIONAL,

hI2DeliveryPortNumber [2] PortNumber OPTIONAL,

hI3DeliveryIPAddress [3] IPAddress OPTIONAL,

hI3DeliveryPortNumber [4] PortNumber OPTIONAL

}

-- ===============

-- MDF definitions

-- ===============

MDFCellSiteReport ::= SEQUENCE OF CellInformation

-- =================

-- Common Parameters

-- =================

AccessType ::= ENUMERATED

{

threeGPPAccess(1),

nonThreeGPPAccess(2),

threeGPPandNonThreeGPPAccess(3)

}

Direction ::= ENUMERATED

{

fromTarget(1),

toTarget(2)

}

DNN ::= UTF8String

E164Number ::= NumericString (SIZE(1..15))

EmailAddress ::= UTF8String

FiveGGUTI ::= SEQUENCE

{

mCC [1] MCC,

mNC [2] MNC,

aMFRegionID [3] AMFRegionID,

aMFSetID [4] AMFSetID,

aMFPointer [5] AMFPointer,

fiveGTMSI [6] FiveGTMSI

}

FiveGMMCause ::= INTEGER (0..255)

FiveGSMRequestType ::= ENUMERATED

{

initialRequest(1),

existingPDUSession(2),

initialEmergencyRequest(3),

existingEmergencyPDUSession(4),

modificationRequest(5),

reserved(6),

mAPDURequest(7)

}

FiveGSMCause ::= INTEGER (0..255)

FiveGTMSI ::= INTEGER (0..4294967295)

FTEID ::= SEQUENCE

{

tEID [1] INTEGER (0.. 4294967295),

iPv4Address [2] IPv4Address OPTIONAL,

iPv6Address [3] IPv6Address OPTIONAL

}

GPSI ::= CHOICE

{

mSISDN [1] MSISDN,

nAI [2] NAI

}

GUAMI ::= SEQUENCE

{

aMFID [1] AMFID,

pLMNID [2] PLMNID

}

GUMMEI ::= SEQUENCE

{

mMEID [1] MMEID,

mCC [2] MCC,

mNC [3] MNC

}

HomeNetworkPublicKeyID ::= OCTET STRING

HSMFURI ::= UTF8String

IMEI ::= NumericString (SIZE(14))

IMEISV ::= NumericString (SIZE(16))

IMPI ::= NAI

IMPU ::= CHOICE

{

sIPURI [1] SIPURI,

tELURI [2] TELURI

}

IMSI ::= NumericString (SIZE(6..15))

Initiator ::= ENUMERATED

{

uE(1),

network(2),

unknown(3)

}

IPAddress ::= CHOICE

{

iPv4Address [1] IPv4Address,

iPv6Address [2] IPv6Address

}

IPv4Address ::= OCTET STRING (SIZE(4))

IPv6Address ::= OCTET STRING (SIZE(16))

IPv6FlowLabel ::= INTEGER(0..1048575)

MACAddress ::= OCTET STRING (SIZE(6))

MCC ::= NumericString (SIZE(3))

MNC ::= NumericString (SIZE(2..3))

MMEID ::= SEQUENCE

{

mMEGI [1] MMEGI,

mMEC [2] MMEC

}

MMEC ::= NumericString

MMEGI ::= NumericString

MSISDN ::= NumericString (SIZE(1..15))

NAI ::= UTF8String

NextLayerProtocol ::= INTEGER(0..255)

NonLocalID ::= ENUMERATED

{

local(1),

nonLocal(2)

}

NSSAI ::= SEQUENCE OF SNSSAI

PLMNID ::= SEQUENCE

{

mCC [1] MCC,

mNC [2] MNC

}

PDUSessionID ::= INTEGER (0..255)

PDUSessionType ::= ENUMERATED

{

iPv4(1),

iPv6(2),

iPv4v6(3),

unstructured(4),

ethernet(5)

}

PEI ::= CHOICE

{

iMEI [1] IMEI,

iMEISV [2] IMEISV

}

PortNumber ::= INTEGER(0..65535)

ProtectionSchemeID ::= INTEGER (0..15)

RATType ::= ENUMERATED

{

nR(1),

eUTRA(2),

wLAN(3),

virtual(4),

nBIOT(5),

wireline(6),

wirelineCable(7),

wirelineBBF(8),

lTEM(9),

nRU(10),

eUTRAU(11),

trustedN3GA(12),

trustedWLAN(13),

uTRA(14),

gERA(15)

}

RejectedNSSAI ::= SEQUENCE OF RejectedSNSSAI

RejectedSNSSAI ::= SEQUENCE

{

causeValue [1] RejectedSliceCauseValue,

sNSSAI [2] SNSSAI

}

RejectedSliceCauseValue ::= INTEGER (0..255)

RoutingIndicator ::= INTEGER (0..9999)

SchemeOutput ::= OCTET STRING

SIPURI ::= UTF8String

Slice ::= SEQUENCE

{

allowedNSSAI [1] NSSAI OPTIONAL,

configuredNSSAI [2] NSSAI OPTIONAL,

rejectedNSSAI [3] RejectedNSSAI OPTIONAL

}

SMPDUDNRequest ::= OCTET STRING

SNSSAI ::= SEQUENCE

{

sliceServiceType [1] INTEGER (0..255),

sliceDifferentiator [2] OCTET STRING (SIZE(3)) OPTIONAL

}

SUCI ::= SEQUENCE

{

mCC [1] MCC,

mNC [2] MNC,

routingIndicator [3] RoutingIndicator,

protectionSchemeID [4] ProtectionSchemeID,

homeNetworkPublicKeyID [5] HomeNetworkPublicKeyID,

schemeOutput [6] SchemeOutput

}

SUPI ::= CHOICE

{

iMSI [1] IMSI,

nAI [2] NAI

}

SUPIUnauthenticatedIndication ::= BOOLEAN

TargetIdentifier ::= CHOICE

{

sUPI [1] SUPI,

iMSI [2] IMSI,

pEI [3] PEI,

iMEI [4] IMEI,

gPSI [5] GPSI,

mSISDN [6] MSISDN,

nAI [7] NAI,

iPv4Address [8] IPv4Address,

iPv6Address [9] IPv6Address,

ethernetAddress [10] MACAddress

}

TargetIdentifierProvenance ::= ENUMERATED

{

lEAProvided(1),

observed(2),

matchedOn(3),

other(4)

}

TELURI ::= UTF8String

Timestamp ::= GeneralizedTime

UEEndpointAddress ::= CHOICE

{

iPv4Address [1] IPv4Address,

iPv6Address [2] IPv6Address,

ethernetAddress [3] MACAddress

}

-- ===================

-- Location parameters

-- ===================

Location ::= SEQUENCE

{

locationInfo [1] LocationInfo OPTIONAL,

positioningInfo [2] PositioningInfo OPTIONAL,

locationPresenceReport [3] LocationPresenceReport OPTIONAL

}

CellSiteInformation ::= SEQUENCE

{

geographicalCoordinates [1] GeographicalCoordinates,

azimuth [2] INTEGER (0..359) OPTIONAL,

operatorSpecificInformation [3] UTF8String OPTIONAL

}

-- TS 29.518 [22], clause 6.4.6.2.6

LocationInfo ::= SEQUENCE

{

userLocation [1] UserLocation OPTIONAL,

currentLoc [2] BOOLEAN OPTIONAL,

geoInfo [3] GeographicArea OPTIONAL,

rATType [4] RATType OPTIONAL,

timeZone [5] TimeZone OPTIONAL,

additionalCellIDs [6] SEQUENCE OF CellInformation OPTIONAL

}

-- TS 29.571 [17], clause 5.4.4.7

UserLocation ::= SEQUENCE

{

eUTRALocation [1] EUTRALocation OPTIONAL,

nRLocation [2] NRLocation OPTIONAL,

n3GALocation [3] N3GALocation OPTIONAL

}

-- TS 29.571 [17], clause 5.4.4.8

EUTRALocation ::= SEQUENCE

{

tAI [1] TAI,

eCGI [2] ECGI,

ageOfLocatonInfo [3] INTEGER OPTIONAL,

uELocationTimestamp [4] Timestamp OPTIONAL,

geographicalInformation [5] UTF8String OPTIONAL,

geodeticInformation [6] UTF8String OPTIONAL,

globalNGENbID [7] GlobalRANNodeID OPTIONAL,

cellSiteInformation [8] CellSiteInformation OPTIONAL,

globalENbID [9] GlobalRANNodeID OPTIONAL

}

-- TS 29.571 [17], clause 5.4.4.9

NRLocation ::= SEQUENCE

{

tAI [1] TAI,

nCGI [2] NCGI,

ageOfLocatonInfo [3] INTEGER OPTIONAL,

uELocationTimestamp [4] Timestamp OPTIONAL,

geographicalInformation [5] UTF8String OPTIONAL,

geodeticInformation [6] UTF8String OPTIONAL,

globalGNbID [7] GlobalRANNodeID OPTIONAL,

cellSiteInformation [8] CellSiteInformation OPTIONAL

}

-- TS 29.571 [17], clause 5.4.4.10

N3GALocation ::= SEQUENCE

{

tAI [1] TAI OPTIONAL,

n3IWFID [2] N3IWFIDNGAP OPTIONAL,

uEIPAddr [3] IPAddr OPTIONAL,

portNumber [4] INTEGER OPTIONAL

}

-- TS 38.413 [23], clause 9.3.2.4

IPAddr ::= SEQUENCE

{

iPv4Addr [1] IPv4Address OPTIONAL,

iPv6Addr [2] IPv6Address OPTIONAL

}

-- TS 29.571 [17], clause 5.4.4.28

GlobalRANNodeID ::= SEQUENCE

{

pLMNID [1] PLMNID,

aNNodeID [2] ANNodeID,

nID [3] NID OPTIONAL

}

ANNodeID ::= CHOICE

{

n3IWFID [1] N3IWFIDSBI,

gNbID [2] GNbID,

nGENbID [3] NGENbID,

eNbID [4] ENbID

}

-- TS 38.413 [23], clause 9.3.1.6

GNbID ::= BIT STRING(SIZE(22..32))

-- TS 29.571 [17], clause 5.4.4.4

TAI ::= SEQUENCE

{

pLMNID [1] PLMNID,

tAC [2] TAC,

nID [3] NID OPTIONAL

}

TAIList ::= SEQUENCE OF TAI

-- TS 29.571 [17], clause 5.4.4.5

ECGI ::= SEQUENCE

{

pLMNID [1] PLMNID,

eUTRACellID [2] EUTRACellID,

nID [3] NID OPTIONAL

}

-- TS 29.571 [17], clause 5.4.4.6

NCGI ::= SEQUENCE

{

pLMNID [1] PLMNID,

nRCellID [2] NRCellID,

nID [3] NID OPTIONAL

}

RANCGI ::= CHOICE

{

eCGI [1] ECGI,

nCGI [2] NCGI

}

CellInformation ::= SEQUENCE

{

rANCGI [1] RANCGI,

cellSiteinformation [2] CellSiteInformation OPTIONAL,

timeOfLocation [3] Timestamp OPTIONAL

}

-- TS 38.413 [23], clause 9.3.1.57

N3IWFIDNGAP ::= BIT STRING (SIZE(16))

-- TS 29.571 [17], clause 5.4.4.28

N3IWFIDSBI ::= UTF8String

-- TS 29.571 [17], table 5.4.2-1

TAC ::= OCTET STRING (SIZE(2..3))

-- TS 38.413 [23], clause 9.3.1.9

EUTRACellID ::= BIT STRING (SIZE(28))

-- TS 38.413 [23], clause 9.3.1.7

NRCellID ::= BIT STRING (SIZE(36))

-- TS 38.413 [23], clause 9.3.1.8

NGENbID ::= CHOICE

{

macroNGENbID [1] BIT STRING (SIZE(20)),

shortMacroNGENbID [2] BIT STRING (SIZE(18)),

longMacroNGENbID [3] BIT STRING (SIZE(21))

}

-- TS 23.003 [19], clause 12.7.1 encoded as per TS 29.571 [17], clause 5.4.2

NID ::= UTF8String (SIZE(11))

-- TS 36.413 [38], clause 9.2.1.37

ENbID ::= CHOICE

{

macroENbID [1] BIT STRING (SIZE(20)),

homeENbID [2] BIT STRING (SIZE(28)),

shortMacroENbID [3] BIT STRING (SIZE(18)),

longMacroENbID [4] BIT STRING (SIZE(21))

}

-- TS 29.518 [22], clause 6.4.6.2.3

PositioningInfo ::= SEQUENCE

{

positionInfo [1] LocationData OPTIONAL,

rawMLPResponse [2] RawMLPResponse OPTIONAL

}

RawMLPResponse ::= CHOICE

{

-- The following parameter contains a copy of unparsed XML code of the

-- MLP response message, i.e. the entire XML document containing

-- a <slia> (described in OMA-TS-MLP-V3\_5-20181211-C [20], clause 5.2.3.2.2) or

-- a <slirep> (described in OMA-TS-MLP-V3\_5-20181211-C [20], clause 5.2.3.2.3) MLP message.

mLPPositionData [1] UTF8String,

-- OMA MLP result id, defined in OMA-TS-MLP-V3\_5-20181211-C [20], Clause 5.4

mLPErrorCode [2] INTEGER (1..699)

}

-- TS 29.572 [24], clause 6.1.6.2.3

LocationData ::= SEQUENCE

{

locationEstimate [1] GeographicArea,

accuracyFulfilmentIndicator [2] AccuracyFulfilmentIndicator OPTIONAL,

ageOfLocationEstimate [3] AgeOfLocationEstimate OPTIONAL,

velocityEstimate [4] VelocityEstimate OPTIONAL,

civicAddress [5] CivicAddress OPTIONAL,

positioningDataList [6] SET OF PositioningMethodAndUsage OPTIONAL,

gNSSPositioningDataList [7] SET OF GNSSPositioningMethodAndUsage OPTIONAL,

eCGI [8] ECGI OPTIONAL,

nCGI [9] NCGI OPTIONAL,

altitude [10] Altitude OPTIONAL,

barometricPressure [11] BarometricPressure OPTIONAL

}

-- TS 29.518 [22], clause 6.2.6.2.5

LocationPresenceReport ::= SEQUENCE

{

type [1] AMFEventType,

timestamp [2] Timestamp,

areaList [3] SET OF AMFEventArea OPTIONAL,

timeZone [4] TimeZone OPTIONAL,

accessTypes [5] SET OF AccessType OPTIONAL,

rMInfoList [6] SET OF RMInfo OPTIONAL,

cMInfoList [7] SET OF CMInfo OPTIONAL,

reachability [8] UEReachability OPTIONAL,

location [9] UserLocation OPTIONAL,

additionalCellIDs [10] SEQUENCE OF CellInformation OPTIONAL

}

-- TS 29.518 [22], clause 6.2.6.3.3

AMFEventType ::= ENUMERATED

{

locationReport(1),

presenceInAOIReport(2)

}

-- TS 29.518 [22], clause 6.2.6.2.16

AMFEventArea ::= SEQUENCE

{

presenceInfo [1] PresenceInfo OPTIONAL,

lADNInfo [2] LADNInfo OPTIONAL

}

-- TS 29.571 [17], clause 5.4.4.27

PresenceInfo ::= SEQUENCE

{

presenceState [1] PresenceState OPTIONAL,

trackingAreaList [2] SET OF TAI OPTIONAL,

eCGIList [3] SET OF ECGI OPTIONAL,

nCGIList [4] SET OF NCGI OPTIONAL,

globalRANNodeIDList [5] SET OF GlobalRANNodeID OPTIONAL,

globalENbIDList [6] SET OF GlobalRANNodeID OPTIONAL

}

-- TS 29.518 [22], clause 6.2.6.2.17

LADNInfo ::= SEQUENCE

{

lADN [1] UTF8String,

presence [2] PresenceState OPTIONAL

}

-- TS 29.571 [17], clause 5.4.3.20

PresenceState ::= ENUMERATED

{

inArea(1),

outOfArea(2),

unknown(3),

inactive(4)

}

-- TS 29.518 [22], clause 6.2.6.2.8

RMInfo ::= SEQUENCE

{

rMState [1] RMState,

accessType [2] AccessType

}

-- TS 29.518 [22], clause 6.2.6.2.9

CMInfo ::= SEQUENCE

{

cMState [1] CMState,

accessType [2] AccessType

}

-- TS 29.518 [22], clause 6.2.6.3.7

UEReachability ::= ENUMERATED

{

unreachable(1),

reachable(2),

regulatoryOnly(3)

}

-- TS 29.518 [22], clause 6.2.6.3.9

RMState ::= ENUMERATED

{

registered(1),

deregistered(2)

}

-- TS 29.518 [22], clause 6.2.6.3.10

CMState ::= ENUMERATED

{

idle(1),

connected(2)

}

-- TS 29.572 [24], clause 6.1.6.2.5

GeographicArea ::= CHOICE

{

point [1] Point,

pointUncertaintyCircle [2] PointUncertaintyCircle,

pointUncertaintyEllipse [3] PointUncertaintyEllipse,

polygon [4] Polygon,

pointAltitude [5] PointAltitude,

pointAltitudeUncertainty [6] PointAltitudeUncertainty,

ellipsoidArc [7] EllipsoidArc

}

-- TS 29.572 [24], clause 6.1.6.3.12

AccuracyFulfilmentIndicator ::= ENUMERATED

{

requestedAccuracyFulfilled(1),

requestedAccuracyNotFulfilled(2)

}

-- TS 29.572 [24], clause 6.1.6.2.17

VelocityEstimate ::= CHOICE

{

horVelocity [1] HorizontalVelocity,

horWithVertVelocity [2] HorizontalWithVerticalVelocity,

horVelocityWithUncertainty [3] HorizontalVelocityWithUncertainty,

horWithVertVelocityAndUncertainty [4] HorizontalWithVerticalVelocityAndUncertainty

}

-- TS 29.572 [24], clause 6.1.6.2.14

CivicAddress ::= SEQUENCE

{

country [1] UTF8String,

a1 [2] UTF8String OPTIONAL,

a2 [3] UTF8String OPTIONAL,

a3 [4] UTF8String OPTIONAL,

a4 [5] UTF8String OPTIONAL,

a5 [6] UTF8String OPTIONAL,

a6 [7] UTF8String OPTIONAL,

prd [8] UTF8String OPTIONAL,

pod [9] UTF8String OPTIONAL,

sts [10] UTF8String OPTIONAL,

hno [11] UTF8String OPTIONAL,

hns [12] UTF8String OPTIONAL,

lmk [13] UTF8String OPTIONAL,

loc [14] UTF8String OPTIONAL,

nam [15] UTF8String OPTIONAL,

pc [16] UTF8String OPTIONAL,

bld [17] UTF8String OPTIONAL,

unit [18] UTF8String OPTIONAL,

flr [19] UTF8String OPTIONAL,

room [20] UTF8String OPTIONAL,

plc [21] UTF8String OPTIONAL,

pcn [22] UTF8String OPTIONAL,

pobox [23] UTF8String OPTIONAL,

addcode [24] UTF8String OPTIONAL,

seat [25] UTF8String OPTIONAL,

rd [26] UTF8String OPTIONAL,

rdsec [27] UTF8String OPTIONAL,

rdbr [28] UTF8String OPTIONAL,

rdsubbr [29] UTF8String OPTIONAL,

prm [30] UTF8String OPTIONAL,

pom [31] UTF8String OPTIONAL

}

-- TS 29.572 [24], clause 6.1.6.2.15

PositioningMethodAndUsage ::= SEQUENCE

{

method [1] PositioningMethod,

mode [2] PositioningMode,

usage [3] Usage

}

-- TS 29.572 [24], clause 6.1.6.2.16

GNSSPositioningMethodAndUsage ::= SEQUENCE

{

mode [1] PositioningMode,

gNSS [2] GNSSID,

usage [3] Usage

}

-- TS 29.572 [24], clause 6.1.6.2.6

Point ::= SEQUENCE

{

geographicalCoordinates [1] GeographicalCoordinates

}

-- TS 29.572 [24], clause 6.1.6.2.7

PointUncertaintyCircle ::= SEQUENCE

{

geographicalCoordinates [1] GeographicalCoordinates,

uncertainty [2] Uncertainty

}

-- TS 29.572 [24], clause 6.1.6.2.8

PointUncertaintyEllipse ::= SEQUENCE

{

geographicalCoordinates [1] GeographicalCoordinates,

uncertainty [2] UncertaintyEllipse,

confidence [3] Confidence

}

-- TS 29.572 [24], clause 6.1.6.2.9

Polygon ::= SEQUENCE

{

pointList [1] SET SIZE (3..15) OF GeographicalCoordinates

}

-- TS 29.572 [24], clause 6.1.6.2.10

PointAltitude ::= SEQUENCE

{

point [1] GeographicalCoordinates,

altitude [2] Altitude

}

-- TS 29.572 [24], clause 6.1.6.2.11

PointAltitudeUncertainty ::= SEQUENCE

{

point [1] GeographicalCoordinates,

altitude [2] Altitude,

uncertaintyEllipse [3] UncertaintyEllipse,

uncertaintyAltitude [4] Uncertainty,

confidence [5] Confidence

}

-- TS 29.572 [24], clause 6.1.6.2.12

EllipsoidArc ::= SEQUENCE

{

point [1] GeographicalCoordinates,

innerRadius [2] InnerRadius,

uncertaintyRadius [3] Uncertainty,

offsetAngle [4] Angle,

includedAngle [5] Angle,

confidence [6] Confidence

}

-- TS 29.572 [24], clause 6.1.6.2.4

GeographicalCoordinates ::= SEQUENCE

{

latitude [1] UTF8String,

longitude [2] UTF8String,

mapDatumInformation [3] OGCURN OPTIONAL

}

-- TS 29.572 [24], clause 6.1.6.2.22

UncertaintyEllipse ::= SEQUENCE

{

semiMajor [1] Uncertainty,

semiMinor [2] Uncertainty,

orientationMajor [3] Orientation

}

-- TS 29.572 [24], clause 6.1.6.2.18

HorizontalVelocity ::= SEQUENCE

{

hSpeed [1] HorizontalSpeed,

bearing [2] Angle

}

-- TS 29.572 [24], clause 6.1.6.2.19

HorizontalWithVerticalVelocity ::= SEQUENCE

{

hSpeed [1] HorizontalSpeed,

bearing [2] Angle,

vSpeed [3] VerticalSpeed,

vDirection [4] VerticalDirection

}

-- TS 29.572 [24], clause 6.1.6.2.20

HorizontalVelocityWithUncertainty ::= SEQUENCE

{

hSpeed [1] HorizontalSpeed,

bearing [2] Angle,

uncertainty [3] SpeedUncertainty

}

-- TS 29.572 [24], clause 6.1.6.2.21

HorizontalWithVerticalVelocityAndUncertainty ::= SEQUENCE

{

hspeed [1] HorizontalSpeed,

bearing [2] Angle,

vSpeed [3] VerticalSpeed,

vDirection [4] VerticalDirection,

hUncertainty [5] SpeedUncertainty,

vUncertainty [6] SpeedUncertainty

}

-- The following types are described in TS 29.572 [24], table 6.1.6.3.2-1

Altitude ::= UTF8String

Angle ::= INTEGER (0..360)

Uncertainty ::= INTEGER (0..127)

Orientation ::= INTEGER (0..180)

Confidence ::= INTEGER (0..100)

InnerRadius ::= INTEGER (0..65535)

AgeOfLocationEstimate ::= INTEGER (0..32767)

HorizontalSpeed ::= UTF8String

VerticalSpeed ::= UTF8String

SpeedUncertainty ::= UTF8String

BarometricPressure ::= INTEGER (30000..155000)

-- TS 29.572 [24], clause 6.1.6.3.13

VerticalDirection ::= ENUMERATED

{

upward(1),

downward(2)

}

-- TS 29.572 [24], clause 6.1.6.3.6

PositioningMethod ::= ENUMERATED

{

cellID(1),

eCID(2),

oTDOA(3),

barometricPresure(4),

wLAN(5),

bluetooth(6),

mBS(7),

motionSensor(8)

}

-- TS 29.572 [24], clause 6.1.6.3.7

PositioningMode ::= ENUMERATED

{

uEBased(1),

uEAssisted(2),

conventional(3)

}

-- TS 29.572 [24], clause 6.1.6.3.8

GNSSID ::= ENUMERATED

{

gPS(1),

galileo(2),

sBAS(3),

modernizedGPS(4),

qZSS(5),

gLONASS(6)

}

-- TS 29.572 [24], clause 6.1.6.3.9

Usage ::= ENUMERATED

{

unsuccess(1),

successResultsNotUsed(2),

successResultsUsedToVerifyLocation(3),

successResultsUsedToGenerateLocation(4),

successMethodNotDetermined(5)

}

-- TS 29.571 [17], table 5.2.2-1

TimeZone ::= UTF8String

-- Open Geospatial Consortium URN [35]

OGCURN ::= UTF8String

END

----------------------NEXT CHANGES---------------------

Annex C (normative): XSD Schema for LI\_X1 extensions

<?xml version="1.0" encoding="utf-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"

xmlns="urn:3GPP:ns:li:3GPPX1Extensions:r16:v3"

targetNamespace="urn:3GPP:ns:li:3GPPX1Extensions:r16:v3"

elementFormDefault="qualified">

<xs:complexType name="X1Extensions">

<xs:sequence>

<xs:element name="Extension" type="X1Extension" minOccurs="1" maxOccurs="unbounded"></xs:element>

</xs:sequence>

</xs:complexType>

<xs:complexType name="PTCLIX1TargetIdentifierExtensions">

<xs:sequence>

<xs:element name="PTCLIX1TargetIdentifier" type="PTCLIX1TargetIdentifier" minOccurs="1" maxOccurs="unbounded"></xs:element>

</xs:sequence>

</xs:complexType>

<xs:complexType name="PTCLIX1TargetIdentifier">

<xs:choice>

<xs:element name="MCPTTID" type="MCPTTID"></xs:element>

<xs:element name="InstanceIdentifierURN" type="InstanceIdentifierURN"></xs:element>

<xs:element name="PTCChatGroupID" type="PTCChatGroupID"></xs:element>

</xs:choice>

</xs:complexType>

<xs:simpleType name="MCPTTID">

<xs:restriction base="xs:anyURI"></xs:restriction>

</xs:simpleType>

<xs:simpleType name="InstanceIdentifierURN">

<xs:restriction base="xs:anyURI"></xs:restriction>

</xs:simpleType>

<xs:simpleType name="PTCChatGroupID">

<xs:restriction base="xs:anyURI"></xs:restriction>

</xs:simpleType>

<xs:complexType name="UPFLIT3TargetIdentifierExtensions">

<xs:sequence>

<xs:element name="UPFLIT3TargetIdentifier" type="UPFLIT3TargetIdentifier" minOccurs="1" maxOccurs="unbounded"></xs:element>

</xs:sequence>

</xs:complexType>

<xs:complexType name="UPFLIT3TargetIdentifier">

<xs:choice>

<xs:element name="FSEID" type="FSEID"></xs:element>

<xs:element name="PDRID" type="xs:unsignedInt"></xs:element>

<xs:element name="QERID" type="xs:unsignedInt"></xs:element>

<xs:element name="NetworkInstance" type="xs:hexBinary"></xs:element>

<xs:element name="GTPTunnelDirection" type="GTPTunnelDirection"></xs:element>

<xs:element name="FTEID" type="FTEID"></xs:element>

</xs:choice>

</xs:complexType>

<xs:complexType name="FSEID">

<xs:sequence>

<xs:element name="SEID" type="xs:unsignedLong"></xs:element>

<xs:element name="IPv4Address" type="IPv4Address" minOccurs="0"></xs:element>

<xs:element name="IPv6Address" type="IPv6Address" minOccurs="0"></xs:element>

</xs:sequence>

</xs:complexType>

<xs:complexType name="FTEID">

<xs:sequence>

<xs:element name="TEID" type="xs:unsignedInt"></xs:element>

<xs:element name="IPv4Address" type="IPv4Address" minOccurs="0"></xs:element>

<xs:element name="IPv6Address" type="IPv6Address" minOccurs="0"></xs:element>

</xs:sequence>

</xs:complexType>

<xs:simpleType name="GTPTunnelDirection">

<xs:restriction base="xs:string">

<xs:enumeration value="Outbound"></xs:enumeration>

<xs:enumeration value="Inbound"></xs:enumeration>

</xs:restriction>

</xs:simpleType>

<xs:element name="IdentifierAssociationExtensions" type="IdentifierAssociationExtensions" ></xs:element>

<xs:complexType name="X1Extension">

<xs:choice>

<xs:element name="LALSLILCSTargetProvisioning" type="LALSLILCSTargetProvisioningExtensions"></xs:element>

<xs:element name="LALSLTFProvisioning" type="LALSLTFProvisioningExtensions"></xs:element>

<xs:element name="HeaderReporting" type="PDHRReportingExtensions"></xs:element>

<xs:element name="IdentifierAssociation" type="IdentifierAssociationExtensions"></xs:element>

</xs:choice>

</xs:complexType>

<xs:complexType name="LALSLILCSTargetProvisioningExtensions">

<xs:sequence>

<xs:element name="PositioningServiceType" type="PositioningServiceType"></xs:element>

<xs:element name="PositioningPeriodicity" type="PositioningPeriodicity" minOccurs="0"></xs:element>

<xs:element name="PositioningParameters" type="PositioningParameters" minOccurs="0"></xs:element>

</xs:sequence>

</xs:complexType>

<xs:simpleType name="PositioningServiceType">

<xs:restriction base="xs:string">

<xs:enumeration value="Immediate"></xs:enumeration>

<xs:enumeration value="Periodic"></xs:enumeration>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="PositioningPeriodicity">

<xs:restriction base="xs:nonNegativeInteger">

</xs:restriction>

</xs:simpleType>

<xs:complexType name="PositioningParameters">

<xs:sequence>

<xs:element name="RequestedLocationType" type="RequestedLocationType" minOccurs="0"></xs:element>

<xs:element name="RequestedResponseType" type="RequestedResponseType" minOccurs="0"></xs:element>

<xs:element name="MaxLocationAge" type="xs:nonNegativeInteger" minOccurs="0"></xs:element>

<xs:element name="ResponseTimingRequired" type="ResponseTimingRequired" minOccurs="0"></xs:element>

<xs:element name="ResponseTimer" type="xs:nonNegativeInteger" minOccurs="0"></xs:element>

<xs:element name="HorizontalAccuracy" type="NumberWithQOSClass" minOccurs="0"></xs:element>

<xs:element name="AltitudeAccuracy" type="NumberWithQOSClass" minOccurs="0"></xs:element>

<xs:element name="MotionStateRequest" type="EmptyElement" minOccurs="0"></xs:element>

</xs:sequence>

</xs:complexType>

<xs:simpleType name="RequestedLocationType">

<xs:restriction base="xs:string">

<xs:enumeration value="CURRENT"></xs:enumeration>

<xs:enumeration value="CURRENT\_OR\_LAST"></xs:enumeration>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="RequestedResponseType">

<xs:restriction base="xs:string">

<xs:enumeration value="SYNC"></xs:enumeration>

<xs:enumeration value="ASYNC"></xs:enumeration>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="ResponseTimingRequired">

<xs:restriction base="xs:string">

<xs:enumeration value="NO\_DELAY"></xs:enumeration>

<xs:enumeration value="LOW\_DELAY"></xs:enumeration>

<xs:enumeration value="DELAY\_TOL"></xs:enumeration>

</xs:restriction>

</xs:simpleType>

<xs:complexType name="NumberWithQOSClass">

<xs:simpleContent>

<xs:extension base="xs:nonNegativeInteger">

<xs:attribute name="qos\_class" type="QOSClass"></xs:attribute>

</xs:extension>

</xs:simpleContent>

</xs:complexType>

<xs:simpleType name="QOSClass">

<xs:restriction base="xs:string">

<xs:enumeration value="ASSURED"></xs:enumeration>

<xs:enumeration value="BEST\_EFFORT"></xs:enumeration>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="EmptyElement">

<xs:restriction base="xs:string">

<xs:enumeration value=""></xs:enumeration>

</xs:restriction>

</xs:simpleType>

<xs:complexType name="LALSLTFProvisioningExtensions">

<xs:sequence>

<xs:element name="LILCSClientAddress" type="LILCSClientIPAddress"></xs:element>

<xs:element name="PositioningParameters" type="PositioningParameters" minOccurs="0"></xs:element>

</xs:sequence>

</xs:complexType>

<xs:complexType name="LILCSClientIPAddress">

<xs:sequence>

<xs:choice>

<xs:element name="IPv4Address" type="IPv4Address"/>

<xs:element name="IPv6Address" type="IPv6Address"/>

</xs:choice>

</xs:sequence>

</xs:complexType>

<xs:simpleType name="IPv4Address">

<xs:restriction base="xs:token">

<xs:pattern value="((25[0-5]|2[0-4][0-9]|[01]?[0-9]?[0-9])\.){3}(25[0-5]|2[0-4][0-9]|[01]?[0-9]?[0-9])"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="IPv6Address">

<xs:restriction base="xs:token">

<xs:pattern value="([0-9a-f]{4}:){7}([0-9a-f]{4})"/>

</xs:restriction>

</xs:simpleType>

<xs:complexType name="PDHRReportingExtensions">

<xs:sequence>

<xs:element name="PDHType" type="PDHType"></xs:element>

</xs:sequence>

</xs:complexType>

<xs:complexType name="PDHType">

<xs:choice>

<xs:element name="PDHR" type="EmptyElement"></xs:element>

<xs:element name="PDSR" type="PDSRParameters"></xs:element>

</xs:choice>

</xs:complexType>

<xs:complexType name="PDSRParameters">

<xs:sequence>

<xs:element name="PDSRTriggerType" type="PDSRTriggerType"></xs:element>

</xs:sequence>

</xs:complexType>

<xs:complexType name="PDSRTriggerType">

<xs:choice>

<xs:element name="TimerExpiry" type="TimerExpiryInSeconds"></xs:element>

<xs:element name="PacketCount" type="xs:nonNegativeInteger"></xs:element>

<xs:element name="ByteCount" type="xs:nonNegativeInteger"></xs:element>

</xs:choice>

</xs:complexType>

<xs:simpleType name="TimerExpiryInSeconds">

<xs:restriction base="xs:nonNegativeInteger">

</xs:restriction>

</xs:simpleType>

<xs:complexType name="IdentifierAssociationExtensions">

<xs:sequence>

<xs:element name="IdentifierAssociationEventsGenerated" type="IdentifierAssociationEventsGenerated"></xs:element>

</xs:sequence>

</xs:complexType>

<xs:simpleType name="IdentifierAssociationEventsGenerated">

<xs:restriction base="xs:string">

<xs:enumeration value="IdentifierAssociation"></xs:enumeration>

<xs:enumeration value="All"></xs:enumeration>

</xs:restriction>

</xs:simpleType>

<xs:element name="IdentityAssociationTargetIdentifier" type="EmptyElement"></xs:element>

</xs:schema>

----------------------NEXT CHANGES---------------------

Annex E (normative): XSD Schema for Identity Association

<?xml version="1.0" encoding="utf-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"

xmlns="urn:3GPP:ns:li:3GPPIdentityExtensions:r16:v1"

xmlns:x1="http://uri.etsi.org/03221/X1/2017/10"

xmlns:common="http://uri.etsi.org/03280/common/2017/07"

targetNamespace="urn:3GPP:ns:li:3GPPIdentityExtensions:r16:v1"

elementFormDefault="qualified">

<xs:import namespace="http://uri.etsi.org/03221/X1/2017/10"></xs:import>

<xs:import namespace="http://uri.etsi.org/03280/common/2017/07"></xs:import>

<xs:complexType name="IdentityAssociationRequest">

<xs:complexContent>

<xs:extension base="x1:X1RequestMessage">

<xs:sequence>

<xs:element name="RequestDetails" type="RequestDetails"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

<xs:complexType name="RequestDetails">

<xs:sequence>

<xs:element name="Type" type="DictionaryEntry" minOccurs="0"/>

<xs:element name="StartTime" type="common:QualifiedDateTime" minOccurs="0"/>

<xs:element name="EndTime" type="common:QualifiedDateTime" minOccurs="0"/>

<xs:element name="ObservedTime" type="common:QualifiedDateTime" minOccurs="0"/>

<xs:element name="RequestValues" type="RequestValues" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="RequestValues">

<xs:sequence>

<xs:element name="RequestValue" type="RequestValue" minOccurs="1" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="RequestValue">

<xs:sequence>

<xs:element name="FormatType" type="FormatType"/>

<xs:element name="Value" type="common:LongString"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="FormatType">

<xs:sequence>

<xs:element name="FormatOwner" type="common:ShortString"/>

<xs:element name="FormatName" type="common:ShortString"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="DictionaryEntry">

<xs:sequence>

<xs:element name="Owner" type="common:ShortString"/>

<xs:element name="Name" type="common:ShortString"/>

<xs:element name="Value" type="common:ShortString"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="IdentityAssociationResponse">

<xs:complexContent>

<xs:extension base="x1:X1ResponseMessage">

<xs:sequence>

<xs:element name="ResponseDetails" type="IdentityResponseDetails"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

<xs:complexType name="IdentityResponseDetails">

<xs:sequence>

<xs:element name="Associations" type="IdentityAssociationRecords" minOccurs="1"></xs:element>

</xs:sequence>

</xs:complexType>

<xs:complexType name="IdentityAssociationRecords">

<xs:sequence>

<xs:element name="IdentityAssociationRecord" type="IdentityAssociationRecord" minOccurs="0" maxOccurs="unbounded"></xs:element>

</xs:sequence>

</xs:complexType>

<xs:complexType name="IdentityAssociationRecord">

<xs:sequence>

<xs:element name="SUPI" type="SUPI" minOccurs="0"></xs:element>

<xs:element name="FiveGGUTI" type="common:NAI" minOccurs="0"></xs:element>

<xs:element name="PEI" type="PEI" minOccurs="0"></xs:element>

<xs:element name="AssociationStartTime" type="common:QualifiedMicrosecondDateTime"></xs:element>

<xs:element name="AssociationEndTime" type="common:QualifiedMicrosecondDateTime"></xs:element>

</xs:sequence>

</xs:complexType>

<xs:complexType name="SUPI">

<xs:choice>

<xs:element name="SUPIIMSI" type="common:SUPIIMSI"></xs:element>

<xs:element name="SUPINAI" type="common:SUPINAI"></xs:element>

</xs:choice>

</xs:complexType>

<xs:complexType name="PEI">

<xs:choice>

<xs:element name="PEIIMEI" type="common:PEIIMEI"></xs:element>

<xs:element name="PEIIMEISV" type="common:PEIIMEISV"></xs:element>

<xs:element name="PEIMAC" type="common:MACAddress"></xs:element>

</xs:choice>

</xs:complexType>

</xs:schema>

----------------------NEXT CHANGES---------------------

Annex F (normative): ASN.1 schema for LI\_XER messages

TS33128IdentityAssociation

{itu-t(0) identified-organization(4) etsi(0) securityDomain(2) lawfulIntercept(2) threeGPP(4) ts33128IdentityAssociation(20) r16(16) version1(1)}

DEFINITIONS IMPLICIT TAGS EXTENSIBILITY IMPLIED ::=

BEGIN

ts33128IdentityAssociationOID RELATIVE-OID ::= {threeGPP(4) ts33128IdentityAssociation(20) r16(16) version1(1)}

iEFRecordOID RELATIVE-OID ::= {tS33128IdentityAssociationOID iEF(1)}

IEFMessage ::= SEQUENCE

{

tS33128IdentityAssociationOID [1] RELATIVE-OID,

record [2] IEFRecord

}

IEFRecord ::= CHOICE

{

associationRecord [1] IEFAssociationRecord,

deassociationRecord [2] IEFDeassociationRecord,

keepalive [3] IEFKeepaliveMessage,

keepaliveResponse [4] IEFKeepaliveMessage

}

IEFAssociationRecord ::= SEQUENCE

{

sUPI [1] SUPI,

fiveGGUTI [2] FiveGGUTI,

timestamp [3] GeneralizedTime,

tAI [4] TAI,

nCGI [5] NCGI,

nCGITime [6] GeneralizedTime,

sUCI [7] SUCI OPTIONAL,

pEI [8] PEI OPTIONAL,

fiveGSTAIList [9] FiveGSTAIList OPTIONAL

}

IEFDeassociationRecord ::= SEQUENCE

{

sUPI [1] SUPI,

fiveGGUTI [2] FiveGGUTI,

timestamp [3] GeneralizedTime,

nCGI [4] NCGI,

nCGITime [5] GeneralizedTime

}

FiveGGUTI ::= OCTET STRING (SIZE(14))

NCGI ::= SEQUENCE

{

pLMNID [1] PLMNID,

nCI [2] NCI

}

PLMNID ::= OCTET STRING SIZE(3)

NCI ::= BIT STRING (SIZE(36))

TAI ::= OCTET STRING (SIZE(6))

SUPI ::= CHOICE

{

iMSI [1] IMSI,

nAI [2] NAI

}

IMSI ::= NumericString (SIZE(6..15))

NAI ::= UTF8String

FiveGSTAIList ::= SEQUENCE OF TAI

PEI ::= CHOICE

{

iMEI [1] IMEI,

iMEISV [2] IMEISV,

mACAddress [3] MACAddress,

eUI64 [4] EUI64  
}

IMEI ::= NumericString (SIZE(14))

IMEISV ::= NumericString (SIZE(16))

MACAddress ::= OCTET STRING (SIZE(6))

EUI64 ::= OCTET STRING (SIZE(8))

SUCI ::= OCTET STRING (SIZE(8..3008))

IEFKeepaliveMessage ::= SEQUENCE

{

sequenceNumber [1] INTEGER

}

END

----------THE END OF CHANGES, NO MORE, ONLY THE GREAT UNKNOWN BEYOND THIS POINT. THERE MIGHT BE DRAGONS. DON’T FORGET YOUR TOWEL----------