ETSI TS 103 479 V1.2.1 (2023-03)

TECHNICAL SPECIFICATION

Emergency Communications (EMTEL); Core elements for network independent access to emergency services

Reference

RTS/EMTEL-00059

Keywords

emergency services, location, multimedia

ETSI

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

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

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Important notice

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

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

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

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

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

Notice of disclaimer & limitation of liability

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

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

Copyright Notification

Contents

Intellectual Property Rights

Essential patents

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

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

Trademarks

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

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

Foreword

This Technical Specification (TS) has been produced by ETSI Special Committee Emergency Communications (EMTEL).

Modal verbs terminology

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

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

Executive summary

The core elements for network independent access to emergency services provide facilities that support centralized mapping and routing functions for current and future emergency communications and operational requirements. The baseline is a network with the functional elements that comprise security measures and the routing capabilities being necessary to forward a call received at any concentration point based on the caller's location to the responsible emergency call centre. In addition, other functional elements and necessary protocols and procedures enabling interoperable and secure implementations are specified to allow multimedia communications as they evolve.

Introduction

At present, an emergency services infrastructure is based on straightforward technical building blocks and a few legal/regulatory aspects. Technical elements, typically part of an incumbent telephone service provider, ensure that emergency calls are routed to the most appropriate PSAP. Such routing is based on static information at the local telephone exchange that provides a mapping between the location of a calling line and the PSAP, or for a mobile call, between the location of the mobile network cell coverage and the PSAP. The mapping information itself is most often managed by the national regulator, and typically, mapping information is represented by dialling code/area code/cell identifier and a table that maps those codes to PSAPs, which are identified by unlisted and often un-dialable numbers.

However, the existing, legacy emergency services infrastructure is not designed in a way that enables interaction with enhanced services, or that current and future communications and operational requirements will be met. Simply put, the emergency services infrastructure has not kept up with technology, thus, is not able to provide the level of service that citizens expect. Hence, new technologies with a new architecture are introduced as core elements for network independent access to emergency services. These elements enable citizens/individuals to contact emergency services in different ways, using the same types of technology as those they use to communicate every day. It also makes possible that PSAPs receive more and better information about emergencies of all magnitudes and improves interoperability between emergency services.

1 Scope

The purpose of the present document is to describe the architecture, the core elements and corresponding technical interfaces for network independent access to emergency services. Elements are: Border Control Function (BCF), Emergency Service Routing Proxy (ESRP), Emergency Call Routing Function (ECRF), Public Safety Answering Point (PSAP), the Location Information Server (LIS), and the Call Transfer Bridge (BRIDGE).

The described architecture is currently named Next Generation 112 architecture.

2 References

2.1 Normative references

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

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

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

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

[1] ETSI ES 203 178: "Functional architecture to support European requirements on emergency caller location determination and transport".

[2] ETSI ES 203 283: "Protocol specifications for Emergency Service Caller Location determination and transport".

[3] ETSI TS 103 625: "Emergency Communications (EMTEL); Transporting Handset Location to PSAPs for Emergency Calls - Advanced Mobile Location".

[4] IETF RFC 2046 (November 1996): "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", N. Freed and N. Borenstein.

[5] IETF RFC 3261 (June 2002): "SIP: Session Initiation Protocol", J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler.

[6] IETF RFC 3262 (June 2002): "Reliability of Provisional Responses in Session Initiation Protocol (SIP)", J. Rosenberg and H. Schulzrinne.

[7] IETF RFC 3264 (June 2002): "An Offer/Answer Model with Session Description Protocol (SDP)", J. Rosenberg and H. Schulzrinne.

[8] IETF RFC 3311 (October 2002): "The Session Initiation Protocol (SIP) UPDATE Method", J. Rosenberg.

[9] IETF RFC 3325 (November 2002): "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity Within Trusted Networks", C. Jennings, J. Peterson and M. Watson.

[10] IETF RFC 3326 (December 2002): "The Reason Header Field for the Session Initiation Protocol (SIP)", D. Oran and G. Camarillo.

[11] IETF RFC 3428 (December 2002): "Session Initiation Protocol (SIP) Extension for Instant Messaging", B. Campbell, J. Rosenberg, H. Schulzrinne, C. Huitema and D. Gurle.

[12] IETF RFC 3515 (July 2003): "The Session Initiation Protocol (SIP) Refer Method", R. Sparks.

[13] IETF RFC 3550 (July 2003): "RTP: A Transport Protocol for Real-Time Applications", H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson.

[14] IETF RFC 3558 (July 2003): "RTP Payload Format for Enhanced Variable Rate Codecs (EVRC) and Selectable Mode Vocoders (SMV)", A. Li.

[15] IETF RFC 3711 (March 2004): "The Secure Real-time Transport Protocol (SRTP)", M. Baugher, D. McGrew, M. Naslund, E. Carrara and K. Norrman.

[16] IETF RFC 3841 (August 2004): "Caller Preferences for the Session Initiation Protocol (SIP)", J. Rosenberg, H. Schulzrinne and P. Kyzivat.

[17] IETF RFC 3856 (August 2004): "A Presence Event Package for the Session Initiation Protocol (SIP)", J. Rosenberg.

[18] IETF RFC 3891 (September 2004): "The Session Initiation Protocol (SIP) "Replaces" Header", R. Mahy, B. Biggs, and R. Dean.

[19] IETF RFC 3911 (October 2004): "The Session Initiation Protocol (SIP) "Join" Header", R. Mahy and D. Petrie.

[20] IETF RFC 3994 (January 2005): "Indication of Message Composition for Instant Messaging", H. Schulzrinne.

[21] IETF RFC 4103 (June 2005): "RTP Payload for Text Conversation", G. Hellstrom and P. Jones.

[22] IETF RFC 4119 (December 2005): "A Presence-Based GEOPRIV Location Object Format", J. Peterson.

[23] IETF RFC 7044 (February 2014): "An Extension to the Session Initiation Protocol (SIP) for Request History Information", M. Barnes, F. Audet, S. Schubert, J. van Elburg, C. Holmberg.

[24] IETF RFC 4412 (February 2006): "Communications Resource Priority for the Session Initiation Protocol (SIP)", H. Schulzrinne and J. Polk.

[25] IETF RFC 4566 (July 2006): "SDP: Session Description Protocol", M. Handley, V. Jacobson and C. Perkins.

[26] IETF RFC 4568 (July 2006): "Session Description Protocol (SDP) Security Descriptions for Media Streams", F. Andreasen, M. Baugher and D. Wing.

[27] IETF RFC 4579 (August 2006): "Session Initiation Protocol (SIP) Call Control - Conferencing for User Agents", A. Johnston and O. Levin.

[28] IETF RFC 4585 (July 2006): "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", J. Ott, S. Wenger, N. Sato, C. Burmeister and J. Rey.

[29] IETF RFC 4660 (September 2006): "Functional Description of Event Notification Filtering", H. Khartabil, E. Leppanen, M. Lonnfors and J. Costa-Requena.

[30] IETF RFC 4661 (September 2006): "An Extensible Markup Language (XML)-Based Format for Event Notification Filtering", H. Khartabil, E. Leppanen, M. Lonnfors and J. Costa-Requena.

[31] IETF RFC 4788 (January 2007): "Enhancements to RTP Payload Formats for EVRC Family Codecs", Q. Xie, and R. Kapoor.

[32] IETF RFC 4975 (September 2007): "The Message Session Relay Protocol (MSRP)", B. Campbell, R. Mahy and C. Jennings.

[33] IETF RFC 4976 (September 2007): "Relay Extensions for the Message Session Relay Protocol (MSRP)", C. Jennings, R. Mahy and A. B. Roach.

[34] IETF RFC 5031 (January 2008): "A Uniform Resource Name (URN) for Emergency and Other Well-Known Services", H. Schulzrinne.

[35] IETF RFC 5104 (February 2008): "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", S. Wenger, U. Chandra, M. Westerlund and B. Burman.

[36] IETF RFC 5168 (March 2008): "XML Schema for Media Control", O. Levin, R. Even and P. Hagendorf.

[37] IETF RFC 5188 (February 2008): "RTP Payload Format for the Enhanced Variable Rate Wideband Codec (EVRC-WB) and the Media Subtype Updates for EVRC-B Codec", H Desineni and Q. Xie.

[38] IETF RFC 5194 (June 2008): "Framework for Real-Time Text Over IP Using the Session Initiation Protocol (SIP)", A. vanWijk and G. Gybels.

[39] IETF RFC 5222 (August 2008): "LoST: A Location-to-Service Translation Protocol", T. Hardie, A. Newton and H. Schulzrinne and H. Tschofenig.

[40] IETF RFC 5223 (August 2008): "Discovering Location-to-Service Translation (LoST) Servers Using the Dynamic Host Configuration Protocol (DHCP)", H. Schulzrinne, J. Polk and H. Tschofenig.

[41] IETF RFC 5411 (February 2009): "A Hitchhiker's Guide to the Session Initiation Protocol (SIP)", J. Rosenberg.

[42] IETF RFC 5621 (September 2009): "Message Body Handling in the Session Initiation Protocol (SIP)", G. Camarillo.

[43] IETF RFC 5627 (October 2009): "Obtaining and Using Globally Routable User Agent URIs (GRUUs) in the Session Initiation Protocol (SIP)", J. Rosenberg.

[44] IETF RFC 5808 (May 2010): "Requirements for a Location-by-Reference Mechanism", R. Marshall.

[45] IETF RFC 5985 (September 2010): "HTTP-Enabled Location Delivery (HELD)", M. Barnes.

[46] IETF RFC 6086 (January 2011): "Session Initiation Protocol (SIP) INFO Method and Package Framework", C. Holmberg, E. Burger and H. Kaplan.

[47] IETF RFC 6155 (March 2011): "Use of Device Identity in HTTP-Enabled Location Delivery (HELD)", J. Winterbottom, H. Tschofenig and R. Barnes.

[48] IETF RFC 6442 (December 2011): "Location Conveyance for the Session Initiation Protocol", J. Polk, B. Rosen and J. Peterson.

[49] IETF RFC 6446 (January 2012): "Session Initiation Protocol (SIP) Event Notification Extension for Notification Rate Control", A. Niemi, K. Kiss and S. Loreto.

[50] IETF RFC 6447 (January 2012): "Filtering Location Notifications in the Session Initiation Protocol (SIP)", R. Mahy, B. Rosen and H. Tschofenig.

[51] IETF RFC 6665 (July 2012): "SIP-Specific Event Notification", A. B. Roach.

[52] IETF RFC 6753 (October 2012): "A Location Dereference Protocol Using HTTP-Enabled Location Delivery (HELD)", J. Winterbottom, H. Tschofenig, H. Schulzrinne and M. Thomson.

[53] IETF RFC 6849 (February 2013): "An Extension to the Session Description Protocol (SDP) and Real-time Transport Protocol (RTP) for Media Loopback", H. Kaplan, K. Hedayat, N. Venna, P. Jones and N. Stratton.

[54] IETF RFC 6881 (March 2013): "Best Current Practice for Communications Services in Support of Emergency Calling", B. Rosen and J. Polk.

[55] IETF RFC 6884 (March 2013): "RTP Payload Format for the Enhanced Variable Rate Narrowband-Wideband Codec (EVRC-NW)", Z. Fang.

[56] IETF RFC 7135 (May 2014): "Registering a SIP Resource Priority Header Field Namespace for Local Emergency Communications", J. Polk.

[57] IETF RFC 8446 (August 2018): "The Transport Layer Security (TLS) Protocol Version 1.3", E. Rescorla.

[58] OASIS (July 2010): "Common Alerting Protocol Version 1.2", Jacob Westfall.

[59] IETF RFC 5139 (February 2008): "Revised Civic Location Format for Presence Information Data Format Location Object (PIDF-LO)", J. Winterbottom and M. Thomson.

[60] IETF RFC 5491 (March 2009): "GEOPRIV Presence Information Data Format Location Object (PIDF-LO) Usage Clarification, Considerations, and Recommendations", J. Winterbottom, M. Thomson and H. Tschofenig.

[61] Recommendation ITU-T G.711 (11/1988): "Pulse code modulation (PCM) of voice frequencies".

[62] IETF RFC 9071 (July 2021): "RTP-Mixer Formatting of Multiparty Real-Time Text", G. Hellström.

[63] IETF RFC 6141 (March 2011): "Re-INVITE and Target-Refresh Request Handling in the Session Initiation Protocol (SIP)", G. Camarillo, C. Holmberg and Y. Ga.

2.2 Informative references

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

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

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

[i.1] EENA: "Next Generation 112 Long Term Definition", Version 1.1, March 2013.

[i.2] ETSI TS 101 470 (V1.1.1): "Emergency Communications (EMTEL); Total Conversation Access to Emergency Services".

[i.3] ETSI TR 103 201 (V1.1.1): "Emergency Communications (EMTEL); Total Conversation for emergency communications; implementation guidelines".

[i.4] ETSI TS 126 114 (V16.6.1): "Universal Mobile Telecommunications System (UMTS); LTE; 5G; IP Multimedia Subsystem (IMS); Multimedia telephony; Media handling and interaction (3GPP TS 26.114 version 16.6.1 Release 16)".

[i.5] ETSI TS 124 229 (V16.10.0): "Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; 5G; IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3 (3GPP TS 24.229 version 16.10.0 Release 16)".

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

emergency call: any type of emergency communications and associated media initiated by an individual and received by a Public Safety Answering Point (PSAP)

emergency service: service which provides urgent assistance in situations where there is a direct risk to life, general public safety, public/private property or the environment (e.g. police, fire, ambulance, coastguard)

NOTE: A PSAP may be an independent organisation or an integrated part of the emergency services.

Public Safety Answering Point (PSAP): physical location where an emergency call from an individual is first answered and from where a request for assistance may be made to the emergency services

3.2 Symbols

Void.

3.3 Abbreviations

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

AML	Advanced Mobile Location
AMR	Adaptive Multi-Rate
ANP	Access Network Provider
ASP	Application Service Provider
BCF	Border Control Function
CA	Certification Authority
CAP	Common Alerting Protocol
CERT	Computer Emergency Response Team
CPE	Call Processing Equipment
CR	Carriage Return
CTI	(ETSI) Center for Testing and Interoperability
DHE	Ephemeral Diffie-Hellman key exchange
ECRF	Emergency Call Routing Function
ECRIT	Emergency Context Resolution with Internet Technologies (IETF WG)
ECSP	Emergency Call Service Provider
EPSG	European Petroleum Survey Group
ES	ETSI Standard
ESInet	Emergency Services IP network
ESRF	Emergency Service Routing Function
ESRP	Emergency Service Routing Proxy
ETSI	European Telecommunications Standards Institute
EVRC	Enhanced Variable Rate Wideband Codec
EVRC-B	Enhanced Variable Rate Wideband Codec -B
GCM	Galois/Counter Mode
GIS	Geographic Information System
HELD	HTTP Enabled Location Delivery
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IANA	Internet Assigned Numbers Authority
IETF	Internet Engineering Task Force
IF	InterFace
IM	Instant Messaging
IMS	IP Multimedia Core Network Subsystem
IP	Internet Protocol
IT	Information Technology
ITU-T	International Telecommunications Union - Telecommunications
JSON	JavaScript Object Notation
LF	Line Feed
LIS	Location Information Server
LO	Location Object
LOST	LOcation to Service Translation
LS	Location Server
MPEG	Moving Picture Experts Group
MSD	Minimum Set of Data

MSRP	Message Session Relay Protocol
NE	Neighbouring Entity
NW	Narrowband-Wideband
PIDF	Presence Information Data Format
PIDF-LO	Presence Information Data Format - Location Object
PNNS	Protocol Naming and Numbering Service
PRF	Policy Routing Function
PSAP	Public Safety Answering Point
PSP	PSAP Service Provider
PSTN	Public Switched Telephone Network
RFC	Request For Comment
RSA	Rivest–Shamir–Adleman
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
RTSP	Real-time Streaming Protocol
SBC	Session Border Controller
SDES	SDP Security Descriptions
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SIPS	Session Initiation Protocol Secure
SMS	Short Message Service
SMSC	Short Message Service Center
SRS	Spatial Reference System
SRTCP	Secure Real-time Transport Control Protocol
SRTP	Secure Real-time Transport Protocol
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TR	(ETSI) Technical Report
TS	(ETSI) Technical Specification
UA	User Agent
UAC	User Agent Client
UAS	User Agent Server
UDP	User Datagram Protocol
UE	User Equipment
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
UTF	Unicode Transformation Format
VSP	Voice Service Provider
WB	Wideband
WGS	World Geodetic System
XML	eXtensible Markup Language

4 General

4.1 Overview

Per ETSI ES 203 178 [1], emergency calls originating in an ANP infrastructure are forwarded via a VSP to an ECSP where the appropriate PSAP or, in general terms, the Point-of-Interconnect to a PSP infrastructure is determined. In general, emergency calls are routed by the ESRF to the ESRP via a BCF utilizing interface ***ih***. Depending on national PSAP models the ESRP may then forward directly to the appropriate PSAP utilizing interface ***ij*** or use PSP internal facilities to determine the correct PSAP (e.g. in the case of nationally interconnected PSAPs).

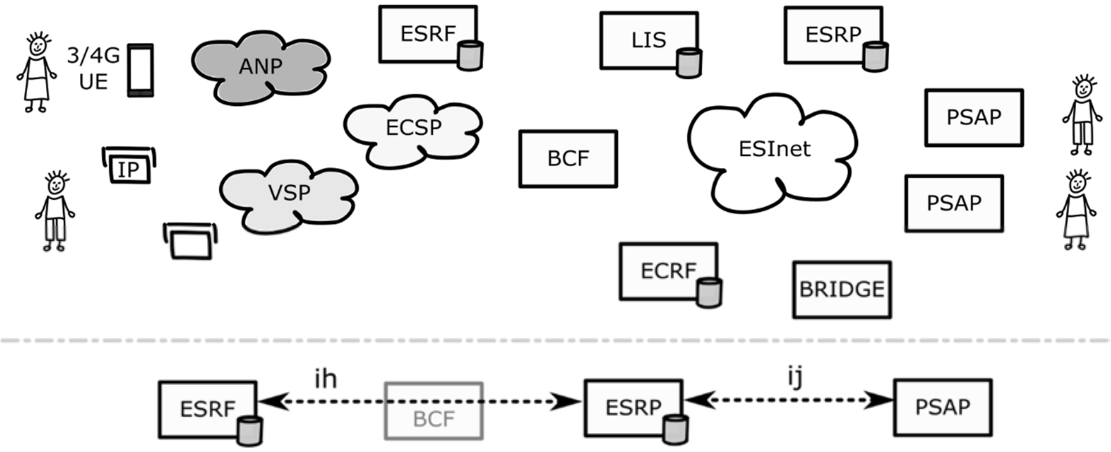

Figure 1: High level functional architecture

ETSI ES 203 283 [2] specifies interfaces ***ih*** and ***ij*** for basic emergency call routing services. The present document aims to extend these interfaces and to specify additional ones to cover PSP specific facilities considering security, location and policy-based routing. Standardization of ANP, VSP or ECSP specific entities are not covered in the present document.

The following architecture introduces functional elements that comprise an IP only PSP environment. Such elements provide security measures (BCF), location information (LIS), emergency call routing (ESRP), mapping PSAP boundaries to SIP URIs (ECRF), bridging (BRIDGE) and call processing equipment (PSAP).

4.2 Architecture

The definition of core elements for network independent access to emergency services is based on the core concept of the NG112 architecture as introduced in [i.1], the Emergency Services IP Network (ESInet). The ESInet is an emergency services network of networks that utilizes IP technology. ESInets are private, managed, and routed IP networks. An ESInet can serve a set of PSAPs, a region, a state, or a set of states. ESInets may be interconnected and shall be built upon common functions and interfaces making ESInets interoperable. The present document defines such functional elements with their external interfaces.

The NG112 architecture fits with the overall concept of different service provider roles as defined in ETSI ES 203 178 [1]. The present document addresses the specific needs of the PSAP Service Provider (PSP) domain and the inter-operator interfaces to other domains. These specific functions of the PSP domain extend the existing definition of the PSP domain (ETSI ES 203 178 [1]), e.g. for the deployment of more complex policies for destination selection.

The functional architecture introduced in ETSI ES 203 178 [1] identifies four service provider roles and a routing function as represented in Figure 2 to the left:

- Access Network Provider (ANP).
- Voice Service Provider (VSP).
- Emergency Call Service Provider (ECSP).
- PSAP Service Provider (PSP).
- Emergency Service Routing Function (ESRF).

The ANP, ECSP and PSP are in the same regulatory domain. The VSP can be inside or outside this domain. The present document extends this architecture with elements located within or accessed from a PSP infrastructure as shown in Figure 2:

- Border Control Function (BCF);
- Emergency Call Routing Function (ECRF);
- Call Bridging function (BRIDGE);

- Public Safety Answering Point (PSAP);
- Emergency Services Routing Proxy (ESRP); and
- Location Information Service (LIS).

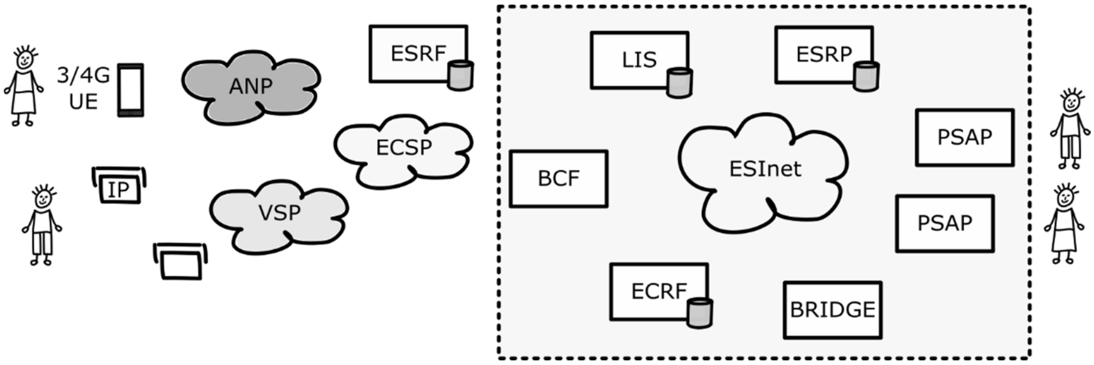

Figure 2: Core elements

4.3 Mandatory Interfaces

Mandatory interfaces are introduced by the present document to define media and signalling capabilities of an ESInet in addition to ETSI ES 203 283 [2]. Figure 3 shows interfaces as listed in the following:

SIP-1, SIP-2:

Interface between BCF, ESRP and PSAP elements that defines SIP transport and signalling capabilities. Note that interfaces respect *ih* and *ij* (ETSI ES 203 283 [2]) capabilities and introduce additional domain specific features.

SIP-E1, SIP-E2:

Interface between ESRP and PSAP elements that defines domain specific SIP event notification capabilities.

HTTP-1:

Interface between ESRP and PSAP elements that defines domain specific web service capabilities.

HTTP-2:

Interface between BCF and PSAP elements that defines domain specific web service capabilities.

LOST-1, LOST-2:

Interface between ESRP, PSAP and ECRF elements that defines LoST signalling capabilities.

LOST-3, LOST-4:

Interface between PSAP and ECRF elements that defines LoST signalling capabilities.

HELD-1, HELD-2:

Interface between ESRP or PSAP and LIS elements that defines location dereference and HELD signalling capabilities.

RTP-1, RTP-2:

Interface between BCF and PSAP elements that defines media transport capabilities and media types for audio, video and real-time text.

IM-1:

PSAP call handling capabilities to support instant messaging.

CAP-1:

PSAP call handling capabilities to support the common alerting protocol.

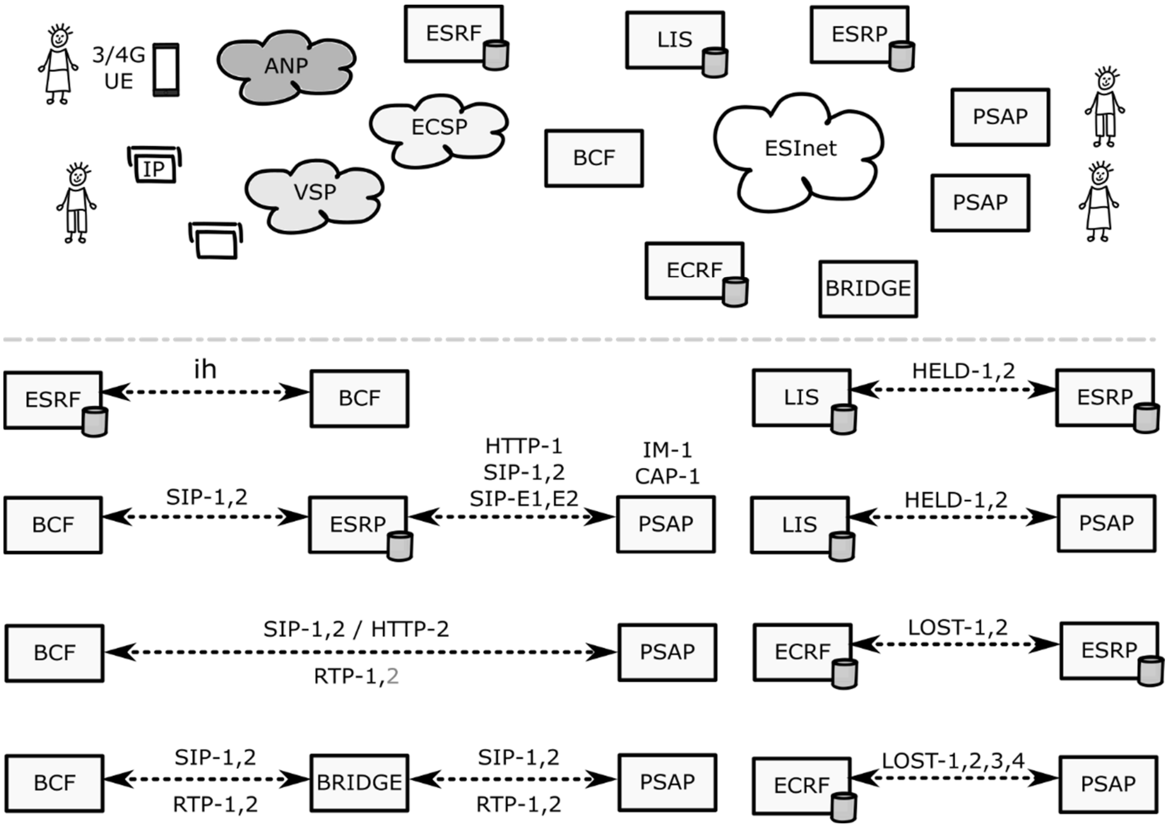

Figure 3: Considered mandatory interfaces

4.4 Optional Interfaces

Optional interfaces are introduced by the present document to define location conveyance via AML (ETSI TS 103 625 [3]) in addition to ETSI ES 203 283 [2]. Figure 4 shows interfaces as listed in the following:

AML-1:

Interface between SMSC and LIS that defines AML via SMS, as in ETSI TS 103 625 [3].

AML-2:

Interface between UE and LIS that defines AML via HTTPS push, as in ETSI TS 103 625 [3].

SIP-4:

Interface between 3GPP VoLTE UE and PSAP that defines in-dialog location refresh via SIP.

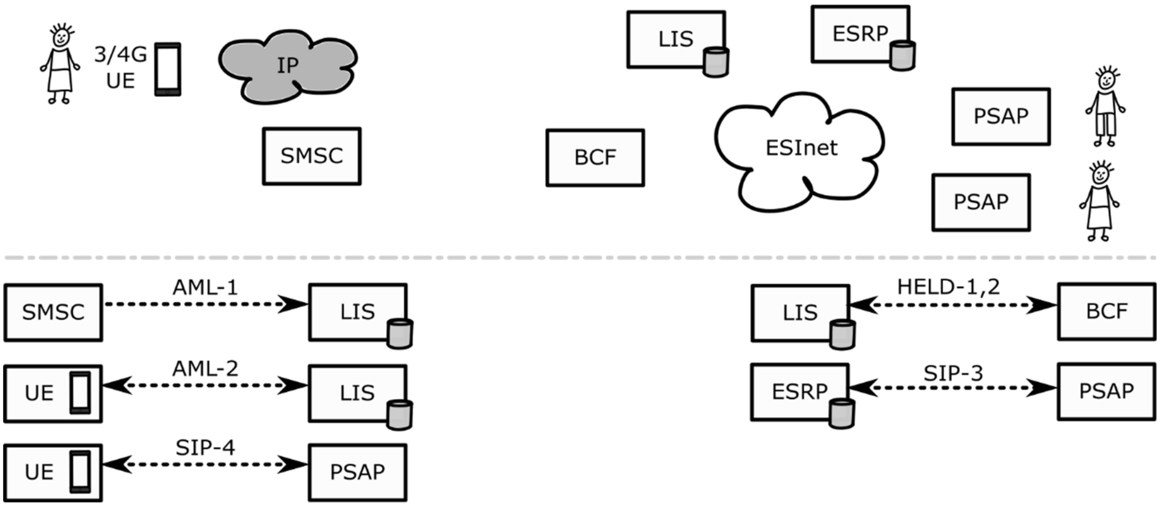

Figure 4: Considered optional interfaces

5 Entities

5.1 Border Control Function (BCF)

5.1.1 Overview

A BCF provides application specific functions at the SIP/SDP [25] protocol layer to perform interconnection between two operator domains at the entrance of the ESInet where all traffic from external networks transits a BCF.

A BCF is the entry point (point-of-interconnect) to the ESInet infrastructure where all traffic from external networks transits. The BCF comprises several distinct elements pertaining to network edge control, SIP message handling (SBC) and media forwarding (RTP [13] relay).

The BCF supports SIP interfaces upstream and downstream. The BCF, when it is the first active SIP element in the path of an emergency call, adds the Call Identifier and Incident Tracking Identifier to the call.

5.1.2 Mandatory Interfaces

To be compliant with the procedures in the present document, a BCF shall support:

1) the *ih* interface as specified in ETSI ES 203 283 [2];

2) the SIP-1 interface as specified in clause 6.1.1;

3) the SIP-2 interface as specified in clause 6.1.2;

4) the HTTP-2 interface as specified in clause 6.2.2;

5) the RTP-1 interface as specified in clause 6.6.1.

Figure 5 shows mandatory interfaces and Neighbouring entities.

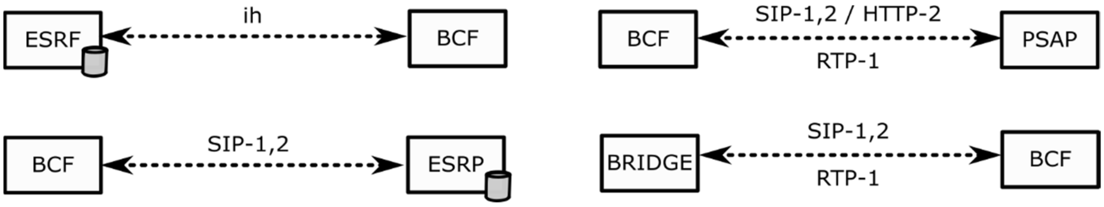

Figure 5: BCF mandatory interfaces

5.1.3 Optional Interfaces

In addition to all mandatory interfaces, a BCF may support:

1) the HELD-1 interface as specified in clause 6.5.1;

2) the HELD-2 interface as specified in clause 6.5.2;

3) the RTP-2 interface as specified in clause 6.6.2.

Figure 6 shows the optional interface and Neighbouring Entity (NE).

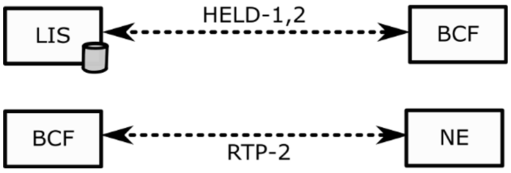

Figure 6: BCF optional interfaces

5.2 Emergency Service Routing Proxy (ESRP)

5.2.1 Overview

The Emergency Service Routing Proxy (ESRP) is the base routing function for emergency calls. ESRPs may operate in a chain (originating, intermediate, or terminating) within the ESInet with the basic function to route a call to the next hop until it reaches the appropriate PSAP.

ESRPs typically receive calls from upstream routing proxies. For the originating ESRP, this is typically a BCF and for an intermediate or terminating ESRP, this is the upstream ESRP. The destination of the call on the output of the ESRP is conceptually a queue, represented by a URI. In most cases, the queue is maintained on a downstream ESRP, and it is possible for more than one downstream element to *pull* calls from the queue. The queue is most often First-In First-Out, but in some cases, there can be out-of-order selections from the queue.

The primary input to an ESRP is a SIP message. The output is a SIP message with a Route header (possibly) rewritten, a Via header added, and in some cases, additional manipulation of the SIP message. To do its job, the ESRP has interfaces to the ECRF for location-based routing information, as well as various event notification sources to gather state, which is used by its Policy Routing Function (PRF).

For a received emergency call, it:

1) evaluates a policy *rule set* for the queue the call arrives on;

2) queries the Emergency Call Routing Function (ECRF) with the location included with the call to determine the *normal* next hop (smaller political or network subdivision, PSAP or call taker group) URI;

3) evaluates a policy rule set for that URI using other inputs available to it such as headers in the SIP message, time of day, PSAP state, etc.

The result of the policy rule evaluation is a URI. The ESRP forwards the call to the URI (which is a queue as above).

The function of the ESRP is to route a call to the next hop. In principle, ESRPs are used in several positions within an ESInet. An originating ESRP routes to the appropriate intermediate ESRP (if one exists), intermediate ESRPs route to the next level intermediate ESRP or to a terminating ESRP. A terminating ESRP routes to a PSAP's call handling system that has registered as dequeuing entity. In the case queue states are used in a PRF, the ESRP shall implement the dequeue registration interface as specified in clause 6.2.1, otherwise an ESRP may support plain SIP registration mechanisms as specified in clause 6.1.3. The selection of the proper mechanism is subject to national regulation.

The ESRP may also handle calls to what used to be called *administrative lines*, meaning calls directed to an E.164 number listed for a PSAP. It is suggested that such calls route through the BCF to an ESRP and be subject to the same security and policy routing as regular emergency calls. Such calls would not have a Geolocation header and the ESRP would not query an ECRF but would use the E.164 number to map to a PSAP URI (the same URI which the ECRF would yield) and use that URI as the *normal* next hop used to select the policy rule set to evaluate.

An ESRP is usually the outbound proxy for calls originating from the PSAP. The ESRP routes calls within the ESInet, and routes calls to destinations outside the ESInet through an appropriate gateway or SIP trunk to a PSTN or another carrier connection. Call-backs to the original caller are an example of such outgoing calls to external destinations. No policy rule set evaluation is used for outgoing calls. While an ESRP could be an incoming proxy for non-emergency calls, such use is beyond the scope of the present document.

5.2.2 Mandatory Interfaces

To be compliant with the procedures in the present document, an ESRP shall support:

1) the SIP-1 interface as specified in clause 6.1.1;

2) the SIP-2 interface as specified in clause 6.1.2;

3) the HTTP-1 interface as specified in clause 6.2.1;

4) the SIP-E1 interface as specified in clause 6.3.1;

5) the SIP-E2 interface as specified in clause 6.3.2;

6) the LOST-1 interface as specified in clause 6.4.1;

7) the LOST-2 interface as specified in clause 6.4.2;

8) the HELD-1 interface as specified in clause 6.5.1;

9) the HELD-2 interface as specified in clause 6.5.2.

Figure 7 shows mandatory interfaces and Neighbouring Entities (NE).

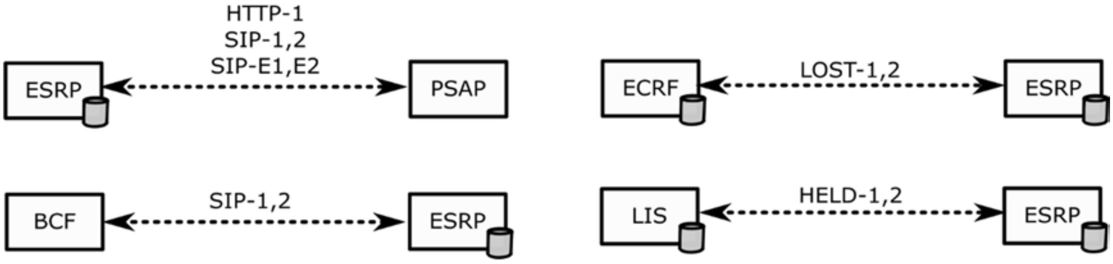

Figure 7: ESRP mandatory interfaces

5.2.3 Optional Interfaces

In addition to all mandatory interfaces, an ESRP may support:

1) the *ih*, *ij* interface as specified in ETSI ES 203 283 [2];

2) the SIP-3 interface as specified in clause 6.1.3;

3) the SIP-E3 interface as specified in clause 6.3.3;

4) the SIP-E4 interface as specified in clause 6.3.4;

5) the SIP-E5 interface as specified in clause 6.3.5.

Figure 8 shows optional interfaces and Neighbouring Entities (NE).

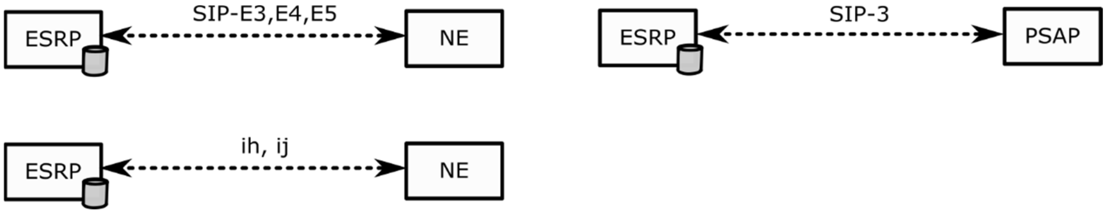

Figure 8: ESRP optional interfaces

5.2.4 Call Queueing

The destination of every routing decision is conceptually a queue of calls. The queue can be large or small, it can have one or many sources entering calls on the queue, and it can have one or many sources taking calls off the queue. All queues defined in the present document are normally First-In First-Out. A unique SIP URI identifies a queue. A queue is managed by an ESRP. A call sent to the queue URI shall route to the ESRP that manages it. Calls are enqueued by forwarding them to the URI (usually obtained by policy rule evaluation of an upstream ESRP). Calls are dequeued by the ESRP sending the call to a downstream entity.

ESRPs may manage multiple queues. For example, an ESRP may manage a queue that is used for normal emergency calls routed to the local ESInet, and one or more queues for calls that are diverted to it by overloaded ESRPs from other areas. Each queue shall have a unique URI that routes to the ESRP.

In practice, some ESRPs are simple IETF RFC 3261 [5] compliant SIP proxy servers making simple routing decisions per IETF RFC 3264 [7]. In such cases, the queue is considered to have a length of one (1) and its existence can be ignored.

The ESRP managing a queue may have a policy that controls which entities may enqueue and dequeue calls to the queue. The dequeuing entity registers (*DequeueRegistration*) to receive calls from the queue. The ESRP returns a call from an entity not in its policy with a `404` response.

The ESRP may maintain a *QueueState* notifier and track the number of calls in queue for the queues that it manages.

5.2.5 Policy Routing

Policy routing refers to the determination of the next hop a call or event is forwarded to by an ESRP. An ESRP shall support basic routing capabilities for calls directed to a specific queue URI. These are:

1) query the ECRF with the location included in the request to determine the *normal* next hop URI;

2) evaluate a policy rule set for that URI using other inputs available to it such as headers in the SIP message, time of day, PSAP state, etc.

The Policy Routing Function (PRF), as part of the ESRP, evaluates two or more policy rulesets: typically, one set determined by the queue the call arrives on (inbound ruleset), the other determined by the result of an ECRF query with the location of the caller (outbound ruleset). Basically, a policy ruleset contains rules, where each rule includes conditions and actions.

The PRF in an ESRP accepts calls directed to a specific queue URI. From that URI, it extracts its own *OriginationPolicy* from its policy store for that URI and executes the ruleset. The rules normally include at least one action `LoSTServiceURN(<urn>)` where urn is a service URN. Upon encountering the `LoSTServiceURN` action, the PRF queries its (configured) ECRF with the location received in the call using the urn parameter in the action. The resulting URI is a variable called `NormalNextHop`. The PRF extracts a *TerminationPolicy* from its policy store associated with the domain of `NormalNextHop` and executes the ruleset associated with that policy. The rules normally include the action `Route` to forward the call.

If the policy-store the ESRP uses does not contain a *TerminationPolicy* rule set for the `NormalNextHop` URI, the ESRP will route the call directly to that URI.

The destination of a `Route` action is usually the URI of a queue, but a simple proxy server can be the next hop. If the PRF has access to queue state of downstream entities it can use that state in evaluating rules. Rules normally have a `Route` action that sends the call to a queue that is available and not full.

The syntax is `Route(<recipient>, <cause>)`, where recipient is a URI which will become the Request URI for the outgoing SIP message, and the `<cause>` is an optional value used with the Reason header associated with a History-Info header in the outgoing SIP message.

Other actions that may occur in a *TerminationPolicy* include `Busy` and `Notify`. By using these mechanisms, the full range of call treatments can be applied to any class of call for any circumstance based on the PRF rule set.

Rules have a priority and if more than one rule evaluates to true, the rule with the highest priority prevails.

Usually, there is a *default* rule for use when everything is in normal status. Most calls will route via this rule, for example: `IF True THEN Route (NormalNextHop) {10}`. Other rules may exist for unusual circumstances.

For typical temporary overload, a specific PSAP would be delegated to take diverted calls (via a rule other than the default rule). A call is said to be diverted when it is sent to a PSAP other than the one serving the location of the caller, usually due to some failure or overload condition. A queue is established for that route, with one dequeuing PSAP. Such a diversion PSAP would be accepting calls on its normal queue as well as the diversion queue. Its rules can differentiate such calls from the queue they arrive on.

For more extensive overload, a group of PSAPs would subscribe to take calls from a designated queue. For example, all PSAPs in neighbouring counties might subscribe to a low priority rule for overload for a county PSAP. Similarly, all appropriate PSAPs in a state or region might dequeue for a `DenialOfServiceAttack` queue.

ESRPs managing a queue may receive calls from one or more upstream entities. Origination rules at the ESRP can govern how such calls are handled, as the URI used to get the call to the ESRP (which could be the name of a queue maintained at the ESRP) is an input to the PRF. When handling diverted calls, no ECRF dip may be needed (and thus no *TerminationPolicy* rule set is used). In such a case, the *OriginationPolicy* rule set would determine the next hop.

Rules can determine the priority of multiple queues feeding calls to the ESRP. PSAP ESRPs may dequeue for multiple call queues managed by it or other entities, placing them on internal queues for call takers.

5.3 Emergency Call Routing Function (ECRF)

5.3.1 Overview

Emergency calls will be routed to the appropriate PSAP by ESRPs based on the location of the caller. In addition, PSAPs may utilize the same routing functionality to determine how to route emergency calls to the correct responder. The functional element responsible for providing routing information to the various querying entities is the Emergency Call Routing Function (ECRF).

An ECRF provided by an emergency service authority and accessible from outside the ESInet shall permit querying by an IP client/endpoint, an IP routing proxy belonging to a VSP, an Emergency Services Routing Proxy (ESRP) in an ESInet, or by some combination of these.

An ECRF accessible inside an ESInet shall permit querying from any entity inside the ESInet. ECRFs provided by other entities may have their own policies on who may query them. An origination network may use an ECRF, or a similar function within its own network, to determine an appropriate route, equivalent to what would be determined by the authoritative ECRF, to the correct ESInet for the emergency call, subject to national regulation or deployment guidelines.

The ECRF shall be used within the ESInet to route calls to the correct PSAP and may be used by the PSAP to route calls to the correct responders.

The ECRF shall support a mechanism by which location information (either civic address or geo-coordinates) and a Service URN serve as input to a mapping function that returns a URI or dial string used to route an emergency call toward the appropriate PSAP for the caller's location, subject to national regulation or deployment guidelines.

In an ECRF, depending on the identity and credentials of the entity requesting the routing information, the response may identify the PSAP or an Emergency Service Routing Proxy (ESRP) that acts on behalf of the PSAP to provide final routing towards the PSAP.

ECRFs may be arranged in trees. The Forest Guide (a specific ECRF) contains entries for (nominally) state level ECRFs. A state ECRF may be authoritative for the entire state, or it may refer or recurse to regional or local ECRFs.

Entities may perform LoST server discovery (as defined in IETF RFC 5223 [40]) to find their local ECRF or may be provisioned with a LoST server address. An ECRF can either answer the query or will refer or recourse in the tree to an ECRF that will eventually lead to the correct response.

5.3.2 Mandatory Interfaces

To be compliant with the procedures in the present document, an ECRF shall support:

1) the LOST-1 interface as specified in clause 6.4.1;

2) the LOST-2 interface as specified in clause 6.4.2;

3) the LOST-3 interface as specified in clause 6.4.3;

4) the LOST-4 interface as specified in clause 6.4.4.

Figure 9 shows mandatory interfaces and Neighbouring entities.

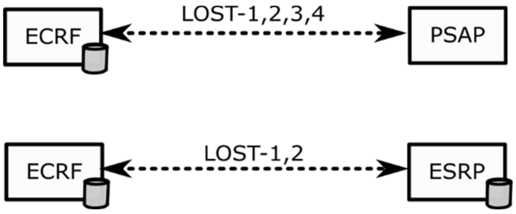

Figure 9: ECRF mandatory interfaces

5.3.3 Optional Interfaces

In addition to all mandatory interfaces, an ECRF may support:

1) the SIP-E3 interface as specified in clause 6.3.3;

2) the SIP-E4 interface as specified in clause 6.3.4;

3) the SIP-E5 interface as specified in clause 6.3.5.

Figure 10 shows the optional interface and Neighbouring Entity (NE).

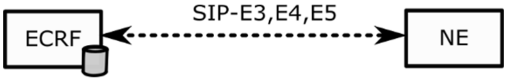

Figure 10: ECRF optional interfaces

5.3.4 Routing Query

When an ECRF receives a LoST query (as defined in IETF RFC 5222 [39]), it determines whether the query was received from an authenticated entity (e.g. an ESRP) and the type of service requested (i.e. emergency services). Authentication shall apply to all entities that initiate queries to the ECRF within the ESInet. TLS is used by all ECRFs within the ESInet, and credentials issued to the querying entity that are traceable to a Certificate Authority (CA) shall be accepted.

Devices and carriers outside the ESInet may not have credentials, therefore the ECRF should assume a common public identity for such queries. Accepting public queries is subject to regulation. Based on the service requested, the ECRF determines which URI is returned in the LoST response. This URI may be a SIP Point-of-Interconnect to the ESInet, a URI of a PSAP, or a downstream ESRP.

The ECRF is provisioned with a service boundary layer containing one or more service boundary polygons. Each of the polygons contains attributes that specify the service URN that the polygon applies to and the mapping the ECRF should return if the provided location is within the polygon. The ECRF returns the URI attribute of the service boundary matching the URN that contains the location.

If the proffered location is not specified as a point (that is the location in the query is a shape) and the shape intersects more than one service boundary with a given service URN, the ECRF response should be the URI of the service boundary with the greatest area of overlap (with a tie breaking policy for the case of equal area of overlap).

If more than one service boundary for the same service URN at a given location (point or civic address) exists in the ECRF, multiple `<mapping>` elements will be returned. The querier shall have local policy to determine how to handle the situation. In some cases, the ECRF may use the identity of the querier, or a distinguished service URN to return the URI of the correct agency. This condition only occurs for queries to an ECRF from within an ESInet. External queries shall only return one URI.

5.3.5 Service Boundary

Location represented by geodetic coordinates provides data that corresponds to a specific geographic location shape. A service boundary is represented by a polygon set. More than one polygon may occur in the set, for example, when the service area has holes or non-contiguous regions.

For each service URN supported by an ECRF, one or more layers will provide polygon sets associated with URIs. Two types of attribute are associated with these polygons:

- URN: the service URN this boundary is associated with.
- URI: a URI returned if the location is within the boundary.

The ECRF computes a response to a LoST query by finding the polygon with the service URN attribute matching that the one provided in the LoST query containing the location and returning the URI attribute of that polygon set. If the proffered location is a shape, that shape may overlap more than one service boundary. The response in that case may be determined by an algorithm in the ECRF and should be the greatest area of overlap but is not otherwise specified in the present document.

The ECRF plays a critical role in the location-based routing of emergency calls. Therefore, it is crucial that the data in the ECRF be accurate and authorized. It is expected that emergency service authorities will be responsible for maintaining the authoritative data for their jurisdiction in the ECRF. The data may be aggregated at a regional or state level, and the ECRF system provided at that level may be the responsibility of the associated state or regional emergency communications agency.

In addition, access or originating network operators may maintain replicas of the ECRF. Thus, the operation and maintenance of individual ECRFs may be the responsibility of the provider of the network in which they physically reside, but it is the emergency service authority that is responsible for maintaining the integrity of the source data housed within those systems. The authority may also provide input to the definition of the policy which dictates the granularity of the routing data returned by the ECRF (i.e. ESRP URIs vs. PSAP URIs), based on the identity of the query originator and/or service URN.

5.4 Public Safety Answering Point (PSAP)

5.4.1 Overview

A PSAP is a service, typically composed of more than one functional element. The functional elements that make up a PSAP are out of scope of the present document. The PSAP deploys the SIP call interface including the multimedia capability, and the non-human-initiated call (emergency event) capability. SIP transactions may contain a `Call-Info` header field with a URI referencing one or more Additional Data blocks. A PSAP should support to dereference the Additional Data URI and should have means to render such information to the user.

PSAPs recognize calls to their administrative numbers received from the ESInet (and distinguishable from normal emergency calls by the presence of the number in a `sip` or `tel` URI as `To` header value and the absence of the SOS Service URN in a Request). The SIP call interface may also be used to place non-emergency calls (including voice-only call backs) from the PSAP using normal SIP trunking mechanisms.

Outgoing calls may be placed via the ESInet using an ESRP as an outgoing proxy server. In most circumstances the ESRP will forward calls through a (configured) BCF to a public network.

5.4.2 Mandatory Interfaces

To be compliant with the procedures in the present document, a PSAP shall support:

1) the SIP-1 interface as specified in clause 6.1.1;

2) the SIP-2 interface as specified in clause 6.1.2;

3) the HTTP-1 interface as specified in clause 6.2.1;

4) the HTTP-2 interface as specified in clause 6.2.2;

5) the SIP-E1 interface as specified in clause 6.3.1;

6) the SIP-E2 interface as specified in clause 6.3.2;

7) the LOST-1 interface as specified in clause 6.4.1;

8) the LOST-2 interface as specified in clause 6.4.2;

9) the LOST-3 interface as specified in clause 6.4.3;

10) the LOST-4 interface as specified in clause 6.4.4:

11) the HELD-1 interface as specified in clause 6.5.1;

12) the HELD-2 interface as specified in clause 6.5.2;

13) the RTP-1 interface as specified in clause 6.6.1;

14) the RTP-2 interface as specified in clause 6.6.2;

15) the IM-1 interface as specified in clause 6.7;

16) the CAP-1 interface as specified in clause 6.8.

Figure 11 shows mandatory interfaces and Neighbouring Entities (NE).

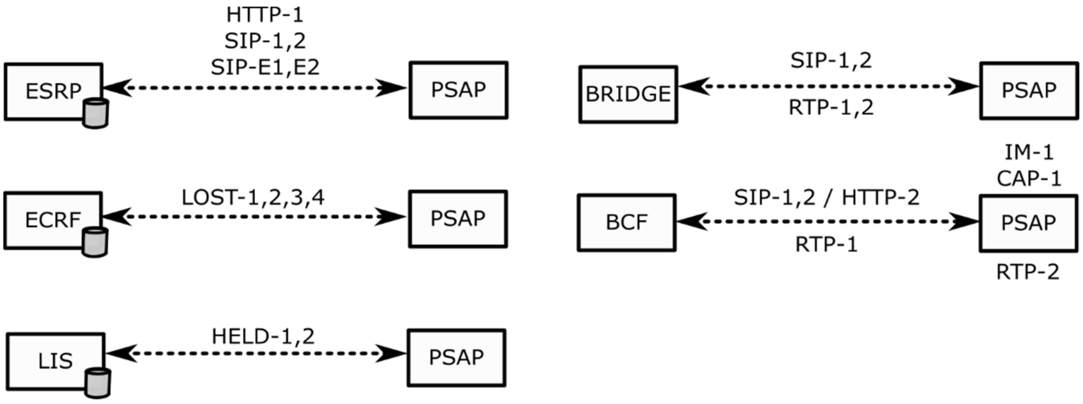

Figure 11: PSAP mandatory interfaces

5.4.3 Optional Interfaces

In addition to all mandatory interfaces, a PSAP may support:

1) the SIP-3 interface as specified in clause 6.1.3;

2) the SIP-4 interface as specified in clause 6.1.4;

3) the SIP-E3 interface as specified in clause 6.3.3;

4) the SIP-E4 interface as specified in clause 6.3.4;

5) the SIP-E5 interface as specified in clause 6.3.5;

6) the HELD-3 interface as specified in clause 6.5.3.

Figure 12 shows the optional interface and Neighbouring Entity (NE).

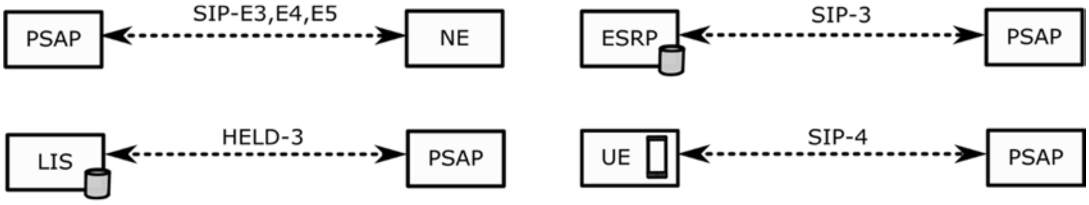

Figure 12: PSAP optional interfaces

5.5 Location Information Server (LIS)

5.5.1 Overview

Location is fundamental to the operation of the emergency services, and the generic functional entity that provides location is a Location Information Server (LIS). For the purposes of the present document, the only capabilities a LIS provides that are relevant are:

a) a dereference function defined for location by reference;

b) a function defined for location requests including different identities;

c) a function to permit requests from third parties;

d) an interface function with AML.

A Location Information Server supplies location in the form of a PIDF-LO (location by value) or a location URI (location by reference). The LIS also provides a "dereference" service for a location URI it supplies: given the URI, the LIS provides the location value as a PIDF-LO. A LIS may be a database, or a protocol interworking function to an access network specific protocol, or both. As a Location Server (LS), the LIS shall explicitly authorize requests from third parties (refer to IETF RFC 6155 [47], section 4.2) according to the policies that are provided by national regulation.

The LIS supplies location (by value or reference) to the endpoint, or proxy operating on behalf of the endpoint. The ESInet is not directly involved in that transaction: the resulting PIDF-LO or location URI appears in the initial SIP message in a Geolocation header.

If the LIS supplies location by reference, it also provides a dereferencing service for that location URI. Elements in the ESInet, including the ESRP and PSAP may dereference a location URI as part of processing a call.

NOTE: In the IETF, location information is a subset of Presence information. The present document uses only the PIDF and according to mechanisms that are described in the Presence service. Other parts of Presence information are not used in emergency calls.

The LIS may support SIP Presence to provide location-by-reference as defined by IETF RFC 5808 [44]. Using SIP Presence, the entity desiring location subscribes to the SIP Presence Event Package (IETF RFC 3856 [17]) at the location URI provided. The LIS sends NOTIFY transactions (IETF RFC 6665 [51]) containing a PIDF document that will include the location in the Location Object (LO) part, forming the PIDF-LO.

An immediate NOTIFY will be generated by the LIS upon acceptance of a subscription request. This would represent the current location of the target. The SUBSCRIBE includes an Expires header (IETF RFC 3261 [5]) which represents the subscribers requested expiration, and the 2XX response contains one that represents the server's actual expiration (which may be shorter, but not longer, than the subscriber's requested time).

An Expires header value of zero indicates a request for exactly one NOTIFY (that is the current location) with no further updates. Subscriptions expire when the call terminates if the LIS is call-aware.

The querier can limit how often further NOTIFYs are sent (before expiration of the subscription) using a filter (IETF RFC 4661 [30]). Rate limits (IETF RFC 6446 [49]) and Location filters (IETF RFC 6447 [50]) are useful for this application and shall be supported by the LIS if it supplies a SIP location URI.

If AML is enabled in the access network, a LIS may implement the capability to act as AML location hub or endpoint and implement SMS, data SMS and HTTP push mechanisms as defined in ETSI TS 103 625 [3]. Further the LIS shall support a conversion of AML format to PIDF-LO.

When location is provided by reference there is a need for the reference to be valid at least for the length of the call. Whether the reference should remain valid for some time beyond the duration of the call is a topic for future study as are the privacy considerations of such access.

5.5.2 Mandatory Interfaces

To be compliant with the procedures in the present document, a LIS shall support:

1) the HELD-1 interface as specified in clause 6.5.1;

2) the HELD-2 interface as specified in clause 6.5.2.

Figure 13 shows mandatory interfaces and Neighbouring entities.

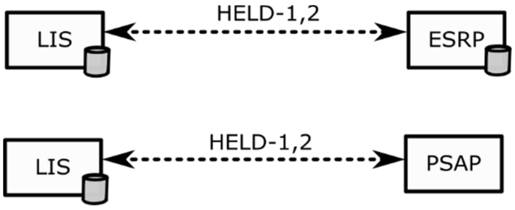

Figure 13: LIS mandatory interfaces

5.5.3 Optional Interfaces

In addition to all mandatory interfaces, a LIS may support:

1) the AML-1 interface as specified in ETSI TS 103 625 [3];

2) the AML-2 interface as specified in ETSI TS 103 625 [3];

3) the HELD-3 interface as specified in clause 6.5.3.

Figure 14 shows optional interfaces and Neighbouring entities.

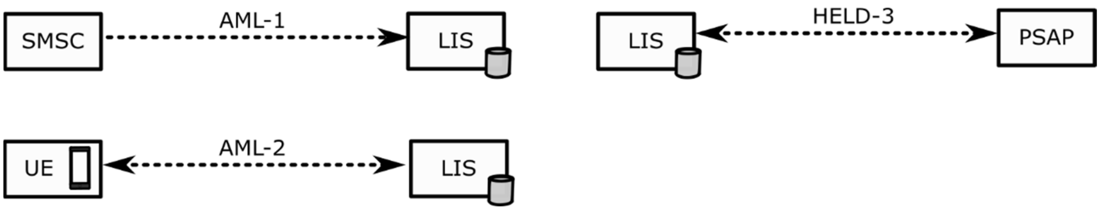

Figure 14: LIS optional interfaces

5.5.4 Location Representation

Location is represented by content in a PIDF-LO document as described in IETF RFC 4119 [22], and updated by IETF RFC 5139 [59] and IETF RFC 5491 [60]. All geodetic data shall WGS84 as the datum. The representation of the location object within the PIDF document shall utilize the `<tuple>` element as defined in IETF RFC 4119 [22].

A `<geopriv>` element shall describe a discrete location. Where a discrete location can be uniquely described in more than one way, each location description should reside in a separate `<tuple>` element with only one `<geopriv>` element per `<tuple>`.

Providing more than one location element in a single `<location-info>` element should only be used for representing compound location referring to the same place. For example, a geodetic location describing a point, and a civic location indicating the floor in a building.

Elements evaluating a PIDF-LO shall respect the order of `<geopriv>` elements in the presence document received, taking into account the guidance in IETF RFC 5491 [60].

NOTE: How to handle a list of elements is out of scope of the present document and may be subject to regulation.

5.6 Call Transfer Bridge (BRIDGE)

5.6.1 Overview

Bridging is used to transfer calls and conduct conferences. Bridges have a SIP signalling interface to create and maintain conferences and media mixing capability for voice, video, and text. A bridge is necessary to transfer a call because IP-based devices normally cannot mix media and transferring always adds the new party (for example, a call taker at a transfer-to PSAP) to the call before the transferor (for example, the original call taker at the PSAP which initially answered the call) drops off the call.

How bridging is employed is characterized by using a bridge only when it is needed during transferring or conferencing more than two parties. The rough transfer sequence for ad hoc, based on the procedures defined in IETF RFC 4579 [27], is:

1) PSAP creates a conference on the bridge.
2) PSAP REFERs the BCF to the bridge.
3) PSAP tears down the original PSAP-Caller leg.
4) PSAP REFERs transfer target (transfer-to PSAP for example) to the conference.
5) PSAP tears down its leg to the conference; the transfer-to PSAP and the caller remain.
6) Transfer-to PSAP REFERs the caller to itself.
7) Transfer-to PSAP terminates the conference.

5.6.2 Mandatory Interfaces

To be compliant with the procedures in the present document, a BRIDGE shall support:

1) the SIP-1 interface as specified in clause 6.1.1;

2) the SIP-2 interface as specified in clause 6.1.2;

3) the RTP-1 interface as specified in clause 6.6.1;

4) the RTP-2 interface as specified in clause 6.6.2.

Figure 15 shows mandatory interfaces and Neighbouring entities.

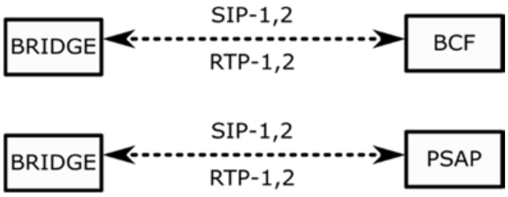

Figure 15: BRIDGE mandatory interfaces

5.6.3 Optional Interfaces

In addition to all mandatory interfaces, a BRIDGE may support:

1) the SIP-E3 interface as specified in clause 6.3.3;

2) the SIP-E4 interface as specified in clause 6.3.4;

3) the SIP-E5 interface as specified in clause 6.3.5.

Figure 16 shows the optional interface and Neighbouring Entity (NE).

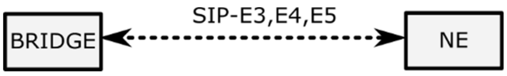

Figure 16: BRIDGE optional interfaces

6 Interfaces

6.1 Signalling

6.1.1 SIP Transport (SIP-1)

SIP signalling within the ESInet shall be TCP with TLS as defined in IETF RFC 8446 [57]. TLS version shall be 1.3 or higher and based on the cipher suites specified in clause C.5. If TLS 1.3 is not supported, fallback to TLS 1.2 is allowed. TLS implementations shall support mutual authentication, which implies both ends have an X.509 certificate available to the other party. How a certificate is created and issued by a Certificate Authority (CA) is out of scope of the present document.

Media streams for voice, video and text shall be carried on RTP over UDP or may be carried on RTSP over UDP.

6.1.2 SIP Session (SIP-2)

6.1.2.1 Overview

The call interface is SIP (as defined in IETF RFC 3261 [5]). All calls presented to the ESInet shall be SIP signalled. Calls are potentially multimedia, and can include one or more forms of media (audio, video and/or text). SIP is also the protocol used to call an emergency caller back, and for calls between agents within the ESInet.

NOTE: All ESInet elements support all forms of media described in the present document. Any given origination network or device may not support all media types, and support of specific media types by origination networks and devices may be subject to regulation.

Elements which process calls shall implement all the standards listed in section 3 (Core Standards) of IETF RFC 5411 [41]. Implementations are cautioned to be "*strict in what you send, and liberal in what you accept*" with respect to such standards. It is generally unacceptable to drop an emergency call because it does not meet standard details if it is reasonably possible to process the call.

The present document does not describe a change to any normative text in any IETF standards-track document. If there is any conflict between the present document and the IETF document concerning how the SIP protocol works, the IETF document is authoritative. Many elements of SIP have options, and the present document may restrict an implementation's use of such options within an ESInet.

There are three primary entities in a SIP protocol exchange:

1) The User Agent Client (UAC), which is the initiator of a "transaction" within SIP. In case of an emergency call, the calling party's end device is the UAC.

2) The User Agent Server (UAS), which is the target of a transaction within SIP. In case of an emergency call, the call taker's end device is the UAS.

3) A Proxy Server, which is an intermediary that assists in the routing of a call. Proxy servers are in the signalling path of a call, but not in the media path. A call may traverse several proxies. In a typical emergency call, the calling party's carrier may have two or more proxies. The ESInet has at least one proxy (an Emergency Services Routing Proxy) and typically has more than one.

6.1.2.2 SIP Methods

INVITE:

The INVITE method is used to initiate a call. The standard INVITE/OK/ACK sequence is to be followed, with allowance for intermediate (`1XX`) responses. It is generally unacceptable to refuse an INVITE request unless a SIP entity is under active attack and cannot respond.

Location is either included by value in the body of a SIP message, with a pointer to it (i.e. a cid URL) in the Geolocation header (IETF RFC 6442 [48]) of the SIP message, or as location by reference, where a location URI is populated in the Geolocation header.

When location is passed by value, processing elements along the path shall not change the location record. If location information changes, a new PIDF-LO with a different `<Provided-by>` element shall be created and passed in addition to the original location.

An emergency call has a Route header obtained from the ECRF based on the location of the call, and a Request URI containing a Service URN. Nominally, the SOS Service URN is `urn:service:sos` but may be `urn:service:sos.police`, `urn:service:sos.fire`, `urn:service:sos.ambulance`, or other sub-services as specified in IETF RFC 5031 [34] according to national regulation. As emergency calls maintain a service urn in the Request-URI, changes in the destination are accomplished via the Route header.

A PSAP should return a `180 Ringing` provisional response when an emergency call is queued for answer. `183 Session Progress` may be used in some specific circumstances. Other `1XX` response codes different to `100 Trying` should not be used by the PSAP due to uneven implementations of these responses.

The `180 Ringing` response should be repeated at approximately 3 second intervals if the call is not answered. When placing a call back, elements shall accept any `1XX` intermediate response and provide an appropriate indication to the caller. UACs within the ESInet shall generate an appropriate audible and in most cases a visual ring indication.

The PSAP should only return a `183 Session Progress` intermediate response when an emergency call is queued for answer. 183 Session Progress should be repeated at approximately 3 seconds interval if the call is not answered. When placing a callback, elements shall accept any `1XX` intermediate response and provide an appropriate indication to the caller. UACs within the ESInet shall generate an appropriate audible and in most cases a visual ring indication.

The normal response to an answered call is `200 OK`.

Emergency calls are usually not redirected, and thus 3XX responses are normally not used; however, 3XX may be used for calls within the ESInet, therefore elements that initiate calls within the network should appropriately respond as defined in IETF RFC 3261 [5].

NOTE: 112 is synonym for any dial string used to contact emergency services. Specifications in the present document neither require a single number nor any specific number to be used to contact a specific service.

Errors typically encountered in a SIP call should be handled as follows.

Table 1

Response Codes to SIP INVITE	Description
180 (Ringing)	An emergency call is queued for answer. It is recommended that no other 1XX response be used due to uneven implementations of these responses. 180 Ringing should be repeated at approximately 3 second intervals if the call is not answered.
200 (OK)	Normal response to an answered call.
3XX	Emergency calls are usually not redirected, and thus 3XX responses are normally not used. 3XX may be used for calls within the ESInet. SIP elements that initiate calls within the ESInet should appropriately respond as defined in IETF RFC 3261 [5].
400 (Bad Request)	A 112 call is so malformed that the BCF cannot parse the message.
401	Should not occur for a 112 call, but proxy authorization is required for all calls originated by entities within an ESInet.
402	Should not occur for a 112 call or an internal call.
403 (Forbidden)	Normally, 403 (Forbidden) should not occur, but if the BCF passes a malformed INVITE which downstream devices cannot handle, they may have no choice but to return 403.
404 (Not Found)	404 (Not Found) would normally not occur for a 112 call but may be used within the ESInet.
406 (Not Acceptable)	The 406 (Not Acceptable) should not occur for a 112 call because the INVITE should not have an Accept header that is unacceptable to the PSAP. If it does, 406 is the correct response.
408 (Request Timeout)	May be issued in an unplanned circumstance. Normally, this should not happen to a 112 call.
413 (Request Entity Too Large)	The BCF should accept any Request URI, but downstream elements may return 413 (Request Entity Too Large).
414 (Request-URI Too Long)	The BCF should accept any Request URI, but downstream elements may return 414 (Request-URI Too Long).
416 (Unsupported URI Scheme)	The BCF should accept any Request URI, but downstream elements may return 416 (Unsupported URI Scheme).
486 (Busy Here)	PSAPs may limit the number of test calls, and if that limit is exceeded, the response shall be 486 Busy Here.
600 (Busy Everywhere)	If the BCF detects an active attack, it should respond with 600 (Busy Everywhere), rather than another 4XX response.

Once a call is established, it may be necessary to modify some of the parameters of the call. For example, it may be necessary to change the media session parameters. In this case, an INVITE transaction on an existing session is used. This is termed a "re-INVITE" in SIP.

Re-INVITEs may be used on any call within the ESInet and may be initiated from either end of the call. Note that when the reINVITE is initiated by the called party, it becomes the UAC and the calling party becomes the UAS.

REFER:

The REFER method is used either:

- to transfer a call;
- to conference additional parties to a call.

REFER is defined in IETF RFC 3515 [12]. The REFER method indicates that the recipient (identified by the Request-URI) should contact a third party using the contact information provided in the `Refer-To` header of the request. The recipient of the REFER request sends an INVITE to the URI in the `Refer-To` header.

REFER creates an implicit subscription to a REFER event package. As with all SIP subscriptions the recipient of the REFER sends an immediate notify confirming instantiation of the subscription. When the INVITE is answered or fails, another NOTIFY is sent with success or failure of the REFER operation.

REFER is sometimes used with the Replaces header, which is dubbed "REFER/Replaces". This is used to replace a call leg with another call leg, an example being replacing a two-way call between the caller and call taker with a leg between the caller and the bridge, with another transaction used to create the leg between the call taker and the bridge.

BYE:

The BYE method is used to terminate a call. BYE may be initiated from either end. PSAPs shall accept a BYE request and honour it.

NOTE: There is a requirement to allow PSAPs to optionally control disconnect. There are no standards that describe how this is accomplished in SIP signalling, but discussion on the subject is ongoing in the IETF ECRIT work group and appropriate work in other SDOs will be required. A future edition of the present document is expected to describe how PSAP control of disconnect is implemented.

CANCEL:

An attempt to create a call with INVITE may be cancelled before it is completed by using the CANCEL method. CANCEL is used before the session is created (call establishment), BYE is used after the session was created. Of course, race conditions exist between the signalling of the session and the attempt to cancel it. These conditions are listed in IETF RFC 3261 [5]. CANCEL is the signalling used to abandon a call, and ESInet elements shall treat a cancelled call as such.

UPDATE:

UPDATE is defined in IETF RFC 3311 [8] and allows a client to update parameters of a session but has no impact on the state of a dialog. In general it is similar to a reINVITE, but unlike reINVITE, it can be sent before the initial INVITE transaction has been completed. Within an ESInet (including emergency calls) UPDATE may be used to change parameters of the conversation before the initial INVITE has been completed, or to refresh location information after the completion of the initial INVITE transaction.

OPTIONS:

OPTIONS may be used by an external caller, or inside the ESInet to determine the capabilities of the destination UA. All endpoints within the ESInet shall respond to an OPTIONS request, as defined in IETF RFC 3261 [5]. It would be unusual, but not improper, for an external caller to query the PSAP with OPTIONS before placing an emergency call.

An OPTIONS transaction is the preferred mechanism for maintaining a *keep-alive* between two SIP elements. Periodic OPTIONS transactions shall be used between ESRPs which normally pass calls between themselves, between the ESRP and the PSAPs, and between the PSAP and the bridge it normally uses. The period between OPTIONs used for keep-alive should be provisioned, and default to 1-minute (to be less than the TLS timeout period) intervals during periods of inactivity. Since OPTIONs requires an exchange of messages, only one member of a pair of "adjacent" SIP elements need initiate OPTIONS towards the other.

ACK:

The ACK request is used to acknowledge completion of a request. Strictly speaking, there are two cases of ACK, one used for a `2XX` series response (which is part of a three-way handshake, typically INVITE/200 (OK)/ACK) and a non-`2XX` response, which is a separate transaction.

MESSAGE:

The MESSAGE method, an extension to SIP, allows the transfer of Instant Messages and is also used to carry a Common Alerting Protocol (CAP) message. Since the MESSAGE request is an extension to SIP, it inherits all the request routing and security features of that protocol. MESSAGE requests carry the content in the form of MIME body parts. MESSAGE requests do not themselves initiate a SIP dialog or session.

MESSAGE requests may be sent in the context of a dialog or session initiated by some other SIP request (such as INVITE), for example in a multi-media call or text messaging session. For more information on MESSAGE refer to IETF RFC 3428 [11]. Non-human-associated calls are sent using MESSAGE requests outside of a session. Text messages or instant messages may be sent using MESSAGE within a session (in which case an interactive associated stream of such messages is established) or outside a session (in which case a set of disconnected stand-alone messages are sent). MESSAGE is part of the SIP/SIMPLE presence and messaging system.

INFO:

The INFO method as defined in IETF RFC 6086 [46], is used for communicating mid-session signalling information along the signalling path for a call. Video communication implementations are depending on use of INFO for requesting a full video frame when packets have been lost as specified in IETF RFC 5168 [36], therefore such use of INFO shall be supported. Orderly transition to the use of RTCP for media control can be achieved if the procedures of IETF RFC 5104 [35] are supported.

SUBSCRIBE/NOTIFY:

Subscribe/Notify, as defined in IETF RFC 6665 [51] is a mechanism to implement asynchronous events notification between two elements, for example, to request current state and updates to state from a remote element. SUBSCRIBE requests should contain an `Expires` header.

This `Expires` value indicates the duration of the subscription. To keep subscriptions effective beyond the duration communicated in the `Expires` header, subscribers need to refresh subscriptions on a periodic basis using a new SUBSCRIBE message on the same dialog.

NOTIFY messages are sent to inform subscribers of changes in state to which the subscriber has a subscription. Subscriptions are typically put in place using the SUBSCRIBE method. A NOTIFY message does not terminate its corresponding subscription. A single SUBSCRIBE request may trigger several NOTIFY requests.

PUBLISH:

PUBLISH is a SIP method for publishing event state. The PUBLISH method allows the user to create, modify and remove state in another entity which manages this state on behalf of the user. The request URI of a PUBLISH request is populated with the address of the resource for which the user wishes to publish event state. The body of a PUBLISH request carries the PUBLISH event state. For more information refer to IETF RFC 3911 [19].

6.1.2.3 Required SIP Headers

Table 2 shows the SIP header fields required in the INVITE and MESSAGE methods, recalling that the Request-URI will contain `urn:service:sos` or a sub-service of it as defined in IETF RFC 6881 [54], section 5.

Table 2

Header Field/Request	Defined In	See section (or IETF RFC 6881 [54])	Notes
Request-URI	IETF RFC 3261 [5] section 8.1.1.1	ED62 1.	"urn:service:sos" or a subservice of it
To	IETF RFC 3261 [5] sections 8.1.1.2 & 20.39	ED62 2.	Usually sip:112 or "urn:service:sos"
From	IETF RFC 3261 [5] sections 8.1.1.3 & 20.20	ED62 3.	Content cannot be trusted unless protected by an Identity header
Via	IETF RFC 3261 [5] sections 8.1.1.7 & 20.42		Occurs multiple times, once for each SIP element in the path
CSeq	IETF RFC 3261 [5] sections 8.1.1.5 & 20.16		Defines the order of transactions in a session
Call-ID	IETF RFC 3261 [5] sections 8.1.1.4 & 20.8		This is the SIP call id
Call-Info	IETF RFC 3261 [5] sections 8.1.1.10 & 20.9		May contain Additional Data, Call and Incident Tracking IDs
Contact	IETF RFC 3261 [5] sections 8.1.1.8 & 20.10	ED62 5.	Usually a "globally routable user agent URI" (gruu) as defined in IETF RFC 5627 [43]
Content-Length	IETF RFC 3261 [5] section 20.14		
Content-Type	IETF RFC 3261 [5] sections 8.2.3 & 20.15		Used, for example, in IETF RFC 4119 [22] and IETF RFC 4566 [25]
Geolocation	IETF RFC 6442 [48]	ED62 8.	
Geolocation-Routing	IETF RFC 6442 [48]	ED62 8.	Specifies if the Geolocation header field can be used for routing
History-Info	IETF RFC 7044 [23]		Indicates the call has been retargeted
P-Access-Network-Info	IETF RFC 3325 [9]		May contain cell site info in carrier specific formats
P-Asserted-Identity	IETF RFC 3325 [9]		Carries the identity of a user verified by authentication
Route	IETF RFC 3261 [5] section 20.34	ED62 4.	Usually the ESRP/PSAP URI on an incoming emergency call

6.1.2.4 Accepted SIP Headers

Table 3 shows the SIP header fields accepted in SIP methods.

Table 3

Header Field	Defined In	Notes
Max-Forwards	IETF RFC 3261 [5], section 20.22	Specifies the maximum number of SIP elements that may be traversed before assuming a routing loop has occurred
Accept-Contact	IETF RFC 3841 [16]	
Accept	IETF RFC 3261 [5], section 20.1	
Content-Encoding	IETF RFC 3261 [5], section 20.12	
Accept-Encoding	IETF RFC 3261 [5], section 20.2	
Content-Language	IETF RFC 3261 [5], section 20.13	
Accept-Language	IETF RFC 3261 [5], section 20.3	
Content-Disposition	IETF RFC 3261 [5], section 20.11	
Record-Route	IETF RFC 3261 [5], section 20.30	
Allow	IETF RFC 3261 [5], section 20.5	
Unsupported	IETF RFC 3261 [5], section 20.40	
Require	IETF RFC 3261 [5], section 20.32	
Proxy-Require	IETF RFC 3261 [5], section 20.29	
Expires	IETF RFC 3261 [5], section 20.19	
Min-Expires	IETF RFC 3261 [5], section 20.23	
Subject	IETF RFC 3261 [5], section 20.36	
Priority	IETF RFC 3261 [5], section 20.26	
Date	IETF RFC 3261 [5], section 20.17	
Timestamp	IETF RFC 3261 [5], section 20.38	
Organization	IETF RFC 3261 [5], section 20.25	

Header Field	Defined In	Notes
User-Agent	IETF RFC 3261 [5], section 20.41	
Server	IETF RFC 3261 [5], section 20.35	
Authorization	IETF RFC 3261 [5], section 20.7	
Authentication-Info	IETF RFC 3261 [5], section 20.6	
Proxy-Authenticate	IETF RFC 3261 [5], section 20.27	
Proxy-Authorization	IETF RFC 3261 [5], section 20.28	
WWW-Authenticate	IETF RFC 3261 [5], section 20.44	
Warning	IETF RFC 3261 [5], section 20.43	
Error-Info	IETF RFC 3261 [5], section 20.18	
Alert-Info	IETF RFC 3261 [5], section 20.4	
In-Reply-To	IETF RFC 3261 [5], section 20.21	
MIME-Version	IETF RFC 3261 [5], section 20.24	
Reply-To	IETF RFC 3261 [5], section 20.31	
Retry-After	IETF RFC 3261 [5], section 20.33	
RAck	IETF RFC 3262 [6], section 7.2	
RSeq	IETF RFC 3262 [6], section 7.1	
Event	IETF RFC 6665 [51], section 7.2.1	
Allow Events	IETF RFC 6665 [51], section 7.2.2	
Subscription-State	IETF RFC 6665 [51], section 7.2.3	
Replaces	IETF RFC 3891 [18]	
Resource-Priority	IETF RFC 4412 [24], section 3.1	

6.1.2.5 Resource Priority

The `Resource-Priority` header (as defined in IETF RFC 4412 [24]) is used on SIP calls to indicate priority that proxy servers give to specific calls. All SIP user agents that place calls within the ESInet shall be able to set `Resource-Priority`. All SIP proxy servers in the ESInet shall implement `Resource-Priority` and process calls in priority order when a queue of calls is waiting for service at the proxy server and, where needed, pre-empt lower priority calls.

BCFs shall police `Resource-Priority` for incoming SIP calls. Calls that appear to be emergency calls shall be marked with a provisioned `Resource-Priority`, which defaults to `esnet.1`. PSAP callbacks during handling of an incident use `esnet.0`. Callbacks outside of an incident are not marked. ESInets normally use the esnet namespace (as defined in IETF RFC 7135 [56]). The use of the namespace in an ESInet is defined as:

esnet.0	calls which relate to an incident in progress, but whose purpose is not critical
esnet.1	emergency calls traversing the ESInet
esnet.2	calls related to an incident in progress which are deemed critical
esnet.3- esnet.7	not defined

6.1.2.6 History-Info and Reason

When a call is retargeted by any routing element, the receiving entity shall have the ability to know why it got the call. For this reason, SIP elements in the ESInet shall support the `History-Info` header (as defined in IETF RFC 7044 [23]) and the associated Reason header (IETF RFC 3326 [10]). Elements which retarget a call, shall add a `History-Info` header indicating the original intended recipient, and the reason why the call was retargeted. ESInet elements shall be prepared to handle a `History-Info` (and its associated Reason header) added by any SIP element.

6.1.2.7 Call-Info

SIP INVITE or MESSAGE transactions may contain a `Call-Info` header field with a URI referencing one or more Additional Data blocks. The transaction to dereference the Additional Data shall be protected with TLS. The dereferencing entity, which may be a PSAP, uses its credentials to dereference the Additional Data URI and should have means to render information to the user.

Specification of `Call-Info` header values other than those listed in the present document are out of scope of the present document. The following example illustrates the use of `Call-Info` to provide a reference to an eCall MSD as part of the SIP message body.

```
Call-Info: <cid:1234567890@atlanta.example.com>;purpose=EmergencyCallData.eCall.MSD
```

The following example illustrates the use of Call-Info to provide a reference to an eCall MSD URL:

```
Call-Info: <https://ls.swisscom.com/abc357yc5ax3o>;purpose=EmergencyCallData.eCall.MSD
```

The first element in an ESInet that handles a call shall assign the Call Identifier. The form of a Call Identifier is a URN (see IETF RFC 5031 [34]) formed by the prefix `urn:emergency:uid:callid:`, a unique string containing alpha and/or numeric characters, the "`:`" character, and the Element Identifier of the element that first handled the call. The unique string portion of the Call Identifier shall be unique for each call the element handles over time. The length of the unique string portion of the Call Identifier shall be between 10 and 30 characters. The Call Identifier is added to a SIP message using the `Call-Info` header field with a purpose of `emergency-CallId`. The following example illustrates the use of `Call-Info` to provide a Call Identifier:

```
Call-Info: <urn:emergency:uid:callid:a56e556d871:bcf.at>;purpose=emergency-CallId
```

The first element in an ESInet that handles a call shall assign the Incident Identifier. The form of an Incident Identifier is a is a URN [34] formed by the prefix `urn:emergency:uid:incidentid:`, a unique string containing alpha and/or numeric characters, the "`:`" character, and the element identifier of the entity that first declared the incident. The unique string shall be unique for each Incident the element handles over time. The length of the unique string portion of the Incident Identifier shall be between 10 and 30 characters. Incident Tracking Identifiers are globally unique and there is an Incident associated with every emergency call. The Incident Tracking Identifier is locally generated and assigned by the first element in the ESInet that handles an emergency call or declares an incident. The Incident Identifier is added to a SIP message using a `Call-Info` header field with a purpose of `emergency-IncidentId`. The following example illustrates the use of `Call-Info` to provide an Incident Identifier:

```
Call-Info: <urn:emergency:uid:incidentid:56..3f:bcf.at>;purpose=emergency-IncidentId
```

The first BCF in an ESInet that handles a call shall assign the Source Identifier to allow a downstream element to mark a particular source of a call as a "bad actor" (usually due to receipt of a call that appears to be part of a deliberate attack on the system) and send a message to the BCF notifying it of this marking, as explained in clause 6.2.2. The form of an Source Identifier is a URN [34] formed by the prefix `urn:emergency:uid:sourceid:`, a unique string containing alpha and/or numeric characters, the "`:`" character, and the Element Identifier of the BCF. The unique string portion of the Source Identifier shall be unique for each source, the BCF handles over time. The length of the unique string portion of the Source Identifier shall be between 10 and 30 characters. The Source Identifier is added to a SIP message using the `Call-Info` header field with a purpose of `emergency-SourceId`. The following example illustrates the use of `Call-Info` to provide a Source Identifier:

```
Call-Info: <urn:emergency:uid:sourceid:a7231gc42:bcf.com>;purpose=emergency-SourceId
```

These identifiers shall be added to the initial message of a dialog forming transaction (INVITE) or the MESSAGE method. The identifiers should be added to all other SIP messages processed by the BCF.

6.1.2.8 SIP Message Bodies

All SIP elements in an ESInet shall support multipart MIME types as defined in IETF RFC 2046 [4] and shall support multipart message handling as specified in IETF RFC 5621 [42]. For example, location and session description may be present in a message body. All SIP elements shall allow additional body content (for example, images, jcards, vcards, eCall MSD, etc.) to pass to the PSAP.

6.1.2.9 SIP Element Overload

Any SIP element may encounter a condition in which it is asked to process more calls than it can handle. Elements shall not return `503 Busy Here` unless it is certain, by design and configuration that the upstream element can reliably cope with the error.

The present document specifies specific methods to avoid overload of calls to specific agencies using the routing rule and queue mechanisms, but a given SIP element may still encounter overload. To cope with such overload, SIP elements may implement the mechanisms described in clause 6.3.4.

6.1.2.10 Test Call

Elements in the SIP signalling path shall implement the test function described in IETF RFC 6881 [54]. As the function is designed to test if an emergency call was placed from the test-initiating device, the test mechanism should mimic the entire actual call path as closely as practical. Further the test mechanism shall be automatic, with no manual intervention required.

An INVITE message with the Service URN of `urn:service:sos.test` shall be interpreted as a request to initiate a test call. The PSAP should return a `200 OK` response in normal conditions, indicating that it will complete the test function. The PSAP may limit the number of test calls. If that limit is exceeded, the response shall be `486 Busy Here`. PSAPs should accept requests for secondary services such as `urn:service:sos.fire.test` and complete a test call. PSAP management may disable the test function (according to the PSAP policy).

If the PSAP accepts the test, it should return a body with MIME type `text/plain` consisting of the following contents:

- The name of the PSAP, terminated by a CR and LF.
- The service URN received, terminated by a CR and LF.
- The location reported with the call (in the geolocation header).

If the location was provided by value, the response would be a natural text version of the received location. If the location was provided by reference, the PSAP should dereference the location, using credentials acceptable to the LIS issued specifically for test purposes. The location returned may not be the same as the LIS would issue for an actual emergency call.

A PSAP accepting a test call should accept a media loopback test as in IETF RFC 6849 [53] and should support the `rtp-pkt-loopback` and `rtp-start-loopback` options. The PSAP CPE should specify a loopback attribute of `loopback-source`, indicating the PSAP being the mirror. The PSAP should loop back no more than 3 packets of each media type accepted (voice, video, text), after which the PSAP should send BYE.

PSAP CPE should refuse repeated requests for test from the same device (same Contact URI or source IP address/port) in a short period of time (e.g. within 2 minutes). Any refusal is signalled with a `486 Busy Here`.

6.1.3 SIP Registration (SIP-3)

6.1.3.1 Overview

The SIP Registration interface provides means to register a PSAP call handling equipment with a terminating ESRP. This interface may be used in ESInet deployments that do not require extended policy routing capabilities as specified in clause 5.2.5.

6.1.3.2 SIP Methods

REGISTER

REGISTER is a SIP method for adding, removing, or querying bindings. A REGISTER request can add a new binding between an address-of-record and one or more contact addresses. When a client sends a REGISTER request, it may suggest an expiration interval that indicates how long the client would like the registration to be valid.

An ESRP receiving a REGISTER request may subscribe for QueueState event notifications as defined in clause 6.3.1.

6.1.3.3 Required SIP Headers

Table 4 shows the SIP header fields required in the REGISTER method.

Table 4

Header Field/Request	Defined In	Notes
To	IETF RFC 3261 [5], sections 8.1.1.2 & 20.39	The To header field contains the address of record whose registration is to be created. Usually the name of the call queue configured for the PSAP
From	IETF RFC 3261 [5], sections 8.1.1.3 & 20.20	The From header field contains the address-of-record of the entity responsible for the registration, usually the PSAP
Contact	IETF RFC 3261 [5], sections 8.1.1.8 & 20.10	The Contact header field contains address bindings, usually a "globally routable user agent URI" (gruu) as defined in IETF RFC 5627 [43]

6.1.4 SIP Location Refresh (SIP-4)

6.1.4.1 Overview

Originating networks or handsets typically provide an early available location estimate in a SIP INVITE that is used to route emergency calls to the appropriate PSAP serving the caller. Since location information may change after the completion of the initial INVITE transaction, or the accuracy of the location may improve in the meantime, a location refresh is a way to send new location information to the connected PSAP. In ETSI TS 124 229 [i.5], SIP INVITE and SIP UPDATE are already defined as possible methods for location conveyance in different use cases, with the mobile device (UE) as the origin. The present document defines an unsolicited location refresh (providing new or more accurate location estimates) after the completion of the initial INVITE transaction.

6.1.4.2 SIP Method

UPDATE

The UPDATE request is constructed as would any other request within an existing dialog, as described in IETF RFC 3311 [8].

INVITE

The INVITE request is constructed as would any other request within an existing dialog, known as a re-INVITE, and as described in IETF RFC 6141 [63].

6.1.4.3 Required SIP Headers

Table 5 shows the SIP header fields required in the UPDATE or INVITE method of a location refresh.

Table 5

Header Field/Request	Defined In	See section (or IETF RFC 6881 [54])	Notes
To	IETF RFC 3261 [5] sections 8.1.1.2 & 20.39	ED62 2.	Usually sip:112 or "urn:service:sos"
From	IETF RFC 3261 [5] sections 8.1.1.3 & 20.20	ED62 3.	Content cannot be trusted unless protected by an Identity header
Via	IETF RFC 3261 [5] sections 8.1.1.7 & 20.42		Occurs multiple times, once for each SIP element in the path
CSeq	IETF RFC 3261 [5] sections 8.1.1.5 & 20.16		Defines the order of transactions in a session
Call-ID	IETF RFC 3261 [5] sections 8.1.1.4 & 20.8		This is the SIP call id
Call-Info	IETF RFC 3261 [5] sections 8.1.1.10 & 20.9		May contain Additional Data, Call and Incident Tracking IDs
Contact	IETF RFC 3261 [5] sections 8.1.1.8 & 20.10	ED62 5.	Usually a "globally routable user agent URI" (gruu) as defined in IETF RFC 5627 [43]
Content-Length	IETF RFC 3261 [5] section 20.14		
Content-Type	IETF RFC 3261 [5] sections 8.2.3 & 20.15		Used, for example, in IETF RFC 4119 [22] and IETF RFC 4566 [25]
Geolocation	IETF RFC 6442 [48]	ED62 8.	
Geolocation-Routing	IETF RFC 6442 [48]	ED62 8.	Since this header content is not needed for a refresh, it may simply be a copy of the initial request to meet requirements of IETF RFC 6442 [48]

6.1.4.4 Location Refresh

Messages that contain new location estimates shall be sent in an existing dialog with the appropriate PSAP (i.e. after the completion of the initial INVITE transaction) and shall pass location information via SIP UPDATE or re-INVITE either by value (PIDF-LO) or by reference (Location URI) as described in clause 6.1.2.2.

Location in an UPDATE or re-INVITE request passed by value in a PIDF-LO document shall be represented as defined in IETF RFC 4119 [22]. All geodetic data shall use WGS84 as the datum. The representation of the location object within the PIDF document shall utilize the 'tuple' element as defined in IETF RFC 4119 [22].

6.2 Web Services

6.2.1 Dequeue Registration (HTTP-1)

6.2.1.1 Overview

Dequeue Registration is a web service whereby the registering entity becomes one of the dequeuing entities, and the ESRP managing the queue will begin to send calls to it. Often, an ESRP will manage a queue where it is the only dequeuer, and this web service will not be needed. When there is more than one dequeuer, they register with this service.

If the ESRP that manages the queue is also a dequeuer, it need not register (to itself). The registration includes a value for `<DequeuePreference>` that is an integer from 1 - 5 (1 indicating lowest, 5 indicating highest preference). When dequeuing calls, the ESRP shall send calls to the entity with the highest preference available to take the call when it reaches the head of the queue. If more than one entity has the same `<DequeuePreference>`, the ESRP should fairly distribute calls to the set of entities with the same `<DequeuePreference>` measured over tens of minutes.

6.2.1.2 Parameter

Table 6: DequeueRegistrationRequest

Parameter	Condition	Description
queueUri	MANDATORY	SIP URI of queue to register on
dequeuerUri	MANDATORY	SIP URI of dequeuer (where to send calls)
expirationTime	MANDATORY	Requested time in seconds this registration will expire
dequeuePreference	OPTIONAL	Integer from 1 - 5 indicating queuing preference. 5 indicating highest preference. Default: 1

Table 7: DequeueRegistrationResponse

Parameter	Condition	Description
expirationTime	MANDATORY	Time in seconds this registration will expire

- Error Codes:
  - 200 OK.
  - 400 Bad Request.
  - 454 Unspecified Error.
  - 456 Bad Queue.
  - 457 Bad dequeuePreference.
  - 458 Policy Violation.

The `<expirationTime>` in the response is the actual expiration, which may be equal to or greater than that in the request depending on the local policy of the ESRP. A request `<expirationTime>` of zero is a request to deregister. The entity managing the queue has a policy of identifying which elements are permitted to register to be a dequeuer. The policy may include specific entities, or classes of entities, appropriate for the queue.

6.2.1.3 Transport Layer Security

HTTP-1 message exchange within the ESInet shall be TCP with TLS as defined in IETF RFC 8446 [57]. TLS version shall be 1.3 or higher and based on the cipher suites specified in clause C.5. If TLS 1.3 is not supported, fallback to TLS 1.2 is allowed. TLS implementations shall support mutual authentication, which implies both ends have an X.509 certificate available to the other party. How a certificate is created and issued by a Certificate Authority (CA) is out of scope of the present document.

6.2.2 Bad Actor (HTTP-2)

6.2.2.1 Overview

When the downstream element identifies a source as a *bad actor*, it signals the BCF as to which source is misbehaving by sending it a request that contains a source identifier from the source parameter `sourceid` that was included in the incoming SIP message in the request body. The BCF responds by returning a status code`.` message.

Upon receiving the request, the BCF should filter out subsequent calls from that source until the attack subsides. The bad actor request/response is a webservice (refer to clause A.8) operated on the domain mentioned in the parameter.

6.2.2.2 Parameter

Table 8: BadActorRequest

Parameter	Condition	Description
BadActorSourceId	MANDATORY	sourceid as string

- Status Codes:
  - 201 Bad Actor successfully added.
  - 401 Unauthorized.
  - 432 Already reported
  - 433 No such sourceId
  - 454 Unspecified Error.

6.2.2.3 Transport Layer Security

HTTP-2 message exchange within the ESInet shall be TCP with TLS as defined in IETF RFC 8446 [57]. TLS version shall be 1.3 or higher and based on the cipher suites specified in clause C.5. If TLS 1.3 is not supported, fallback to TLS 1.2 is allowed. TLS implementations shall support mutual authentication, which implies both ends have an X.509 certificate available to the other party. How a certificate is created and issued by a Certificate Authority (CA) is out of scope of the present document.

6.3 Event Notification

6.3.1 Queue State (SIP-E1)

6.3.1.1 Overview

QueueState is an event that indicates to an upstream entity the state of a queue. The SIP NOTIFY mechanism defined in IETF RFC 6665 [51] is used to report QueueState. The event includes the URI of the queue, the current queue length, allowed maximum length and a state enumeration. ETSI Protocol Naming and Numbering Service (PNNS) has created a registry (QueueState) of allowed values with initial defined states of:

- Active: one or more entities are actively available and are currently handling calls being enqueued.
- Inactive: no entity is available or actively handling calls being enqueued.
- Disabled: The queue is disabled by management action and no calls may be enqueued.
- Full: The queue is full, and no new calls can be enqueued on it.
- Standby: the queue has one or more entities that are available to take calls, but the queue is not presently in use. When a call is enqueued, the state changes to "Active".

Race conditions exist where a dequeued call may be sent to an entity that just became congested. A call/event sent to a queue which is Inactive or Disabled, or where the current queue length is equal to or greater than the allowed maximum queue length will have an error (`486 Busy Here`) returned by the dequeuer.

An ESRP that dequeues a call, sends it to a downstream entity and receives a `486` in return shall be able to either re-enqueue the call (at the head of the line) or send it to another dequeuing entity. Note that the upstream ESRP may be configured with policy rules that will specify alternate treatment based on downstream queue state.

ESRPs normally send calls to downstream entities that indicate they are available to take calls. This availability, however, is from the downstream entities point of view. Network state may preclude an upstream entity from sending calls downstream. Normal SIP processing would eventually result in timeouts if calls were sent to an entity that never responds because the packets never arrive. Timeouts are long, however, and a more responsive mechanism is desirable to ensure rapid response to changing network conditions to route calls optimally.

If active calls are being handled, the upstream entity knows the downstream entity is connected. However, some routes are seldom used, and a mechanism shall be provided that ensures the connectedness of each entity remains known.

For this purpose, relatively frequent NOTIFYs of the QueueState event are used. Successful completion of the NOTIFY is an indication to the upstream entity that calls sent to the downstream entity should succeed. The subscription may include a "force" and/or "throttle" filter as described in IETF RFC 4660 [29] and IETF RFC 6446 [49] to control the rate of notifications.

NOTE: QueueState is not required to be implemented on simple routing proxy or when queue length is 1 and only one dequeuer is permitted.

6.3.1.2 Parameter

Event Package Name: emergency-QueueState

Event Package Parameters: None

SUBSCRIBE Bodies: Standard IETF RFC 4661 [30] + extensions filter specification may be present

Subscription Duration: Default 1 hour. One (1) minute to 24 hours is reasonable

NOTIFY Bodies: MIME type application/ emergencyCallData.QueueState+json

Table 9

Parameter	Condition	Description
queueUri	MANDATORY	SIP URI of queue
queueLength	MANDATORY	Integer indicating current number of calls on the queue
queueMaxLength	MANDATORY	Integer indicating maximum length of queue
state	MANDATORY	Enumeration of current queue state (e.g. Active/Inactive/Disabled)

Notifier Processing of SUBSCRIBE Requests:

The Notifier (i.e. the ESRP) consults the policy (QueueState) to determine if the requester is permitted to subscribe. If not, the ESRP returns `603 Decline`. The ESRP determines whether the queue is one of the queues managed by the Notifier. If not, the ESRP return `488 Not Acceptable` Here. If the request is acceptable, the Notifier returns `200 OK`. and shall immediately send a NOTIFY with the current state.

Notifier Generation of NOTIFY Requests:

When state of the queue changes (call is placed on, removed from the queue, or management action/device failure changes the "state" enumeration), a new NOTIFY is generated, adhering to the filter requests.

Subscriber Processing of NOTIFY Request:

Specific action is not required.

Handling of Forked Requests:

Forking is not expected to be used with this package.

Rate of Notification:

This package is designed for relatively high frequency of notifications. The subscriber can control the rate of notifications using the filter rate control (IETF RFC 6446 [49]). The default throttle rate is one notification per second. The default force rate is one notification per minute. The Notifier shall generate NOTIFY messages at the maximum busy second call rate to the maximum number of downstream dequeuing entities, plus at least ten (10) other subscribers.

State Agents:

Special handling is not required.

NOTE: The upstream ESRP may be configured with policy rules that will specify alternate treatment based on downstream queue state.

6.3.2 Abandoned Call (SIP-E2)

6.3.2.1 Overview

The ESRP uses the AbandonedCallEvent to notify a PSAP that a call was started, but then cancelled prior to the PSAP knowing the call occurred.

6.3.2.2 Parameter

Event Package Name: emergency-AbandonedCall

Event Package Parameters: None

SUBSCRIBE Bodies: Standard IETF RFC 4661 [30] + extensions filter specification may be present

Subscription Duration: Default one (1) hour. One (1) minute to twenty-four (24) hours is reasonable

NOTIFY Bodies: MIME type application/ emergencyCallData.AbandonedCall+json

Table 10

Parameter	Condition	Description
invite	MANDATORY	Content of INVITE message
inviteTimestamp	MANDATORY	Timestamp call was received at ESRP
cancelTimestamp	MANDATORY	Timestamp CANCEL was received at ESRP

Notifier Processing of SUBSCRIBE Requests:

The notifier consults the policy (AbandonedCall) to determine if the requester is permitted to subscribe. It returns `603 Decline` if not acceptable. If the request is acceptable, it returns `200 OK` and shall immediately send a NOTIFY with current parameters.

Notifier Generation of NOTIFY Requests:

When the ESRP receives a CANCEL for a call, and it is not certain that the downstream entity that should get that call received an INVITE for the call, a new NOTIFY is generated, adhering to the filter requests.

Subscriber Processing of NOTIFY Requests:

No specific action required.

Handling of Forked Requests:

Forking is not expected to be used with this package.

Rate of Notification:

A series of fast INVITE/CANCEL is a possible DDoS attack. The rate of notification should be limited to a provisioned value. Three (3) per second is a reasonable limit.

State Agents:

No special handling is required.

6.3.3 Security Posture (SIP-E3)

6.3.3.1 Overview

SecurityPosture is an event that represents a downstream entity's current security state. ETSI Protocol Naming and Numbering Service (PNNS) has created a registry (SecurityPosture) of allowed values with initial defined states of:

- Green - The entity is operating normally.
- Yellow - The entity is receiving suspicious activity, but can operate normally.
- Orange - The entity is receiving fraudulent calls/events, is stressed, but is able to continue most operations.
- Red - The entity is under active attack and is overwhelmed.

6.3.3.2 Parameter

Event Package Name: emergency-SecurityPosture

Event Package Parameters: None

SUBSCRIBE Bodies: Standard IETF RFC 4661 [30] + extensions filter specification may be present

Subscription Duration: Default 1 hour. One (1) minute to 24 hours is reasonable

NOTIFY Bodies: MIME type application/ emergencyCallData.SecurityPosture+json

Table 11

Parameter	Condition	Description
service	MANDATORY	
name	MANDATORY	Name of service
domain	MANDATORY	Service domain
securityPosture	MANDATORY	
posture	MANDATORY	Enumeration of current security posture from SecurityPosture registry

Notifier Processing of SUBSCRIBE Requests:

The notifier consults the policy (SecurityPosture) to determine if the requester is permitted to subscribe. It returns `603 Decline` if not acceptable. If the request is acceptable, it returns `200 OK`. and shall immediately send a NOTIFY with the current state.

Notifier Generation of NOTIFY Requests:

When the security posture of the element changes, a new NOTIFY request is generated, adhering to the filter requests.

Subscriber Processing of NOTIFY Requests:

No specific action required.

Handling of Forked Requests:

Forking is not expected to be used with this package.

Rate of Notification:

Posture state normally does not change rapidly. Changes may occur in minutes if attacks start and stop sporadically.

State Agents:

No special handling is required.

6.3.4 Element State (SIP-E4)

6.3.4.1 Overview

ElementState is an event that indicates the state of an element either automatically determined, or as determined by management. ETSI Protocol Naming and Numbering Service (PNNS) has created a registry (ElementState) of allowed values with initial defined states of:

- Normal: The element is operating normally.
- ScheduledMaintenance: The element is undergoing maintenance activities and is not processing requests.
- ServiceDisruption: The element has significant problems and is not able to process all requests.
- Overloaded: The element is completely overloaded.
- GoingDown: The element is being taken out of service.
- Down: The element is unavailable.

In addition, if the subscriber to an element is unable to contact that element, it may show the state of the element as "Unreachable".

NOTE: When an implementation provides redundant physical implementations to increase reliability, usually the set of physical boxes is treated as a single element with respect to the rest of the ESInet and there is only one element state.

6.3.4.2 Parameter

Event Package Name: emergency-ElementState

Event Package Parameters: None

SUBSCRIBE Bodies: Standard IETF RFC 4661 [30] + extensions filter specification may be present

Subscription Duration: Default 1 hour. One (1) minute to 24 hours is reasonable

NOTIFY Bodies: MIME type application/ emergencyCallData.ElementState+json

Table 12

Parameter	Condition	Description
elementId	MANDATORY	Element identifier
state	MANDATORY	Enumeration of current state from ElementState registry
reason	MANDATORY	Text containing the reason state was changed, if available

Notifier Processing of SUBSCRIBE Requests:

The notifier consults the policy (ElementState) to determine if the requester is permitted to subscribe. It returns `603 Decline` if not acceptable. If the request is acceptable, it returns `200 OK` and shall immediately send a NOTIFY with the current state. Notifiers shall implement event rate filters, as described in IETF RFC 6446 [49].

Notifier Generation of NOTIFY Requests:

When the state of the element changes, a new NOTIFY request is generated, adhering to the filter requests. Filter requests may specify a minimum notification interval. The element shall generate a NOTIFY meeting this filter, if Hspecified. This can be used as a watchdog mechanism.

Subscriber Processing of NOTIFY Requests:

No specific action required.

Handling of Forked Requests:

Forking is not expected to be used with this package.

Rate of Notification:

State normally does not change rapidly. Changes may occur in tens of seconds if the network or systems are unstable.

State Agents:

No special handling is required.

6.3.5 Service State (SIP-E5)

6.3.5.1 Overview

ServiceState is an event that indicates the state of service either automatically determined, or as determined by management. ETSI Protocol Naming and Numbering Service (PNNS) has created a registry (ServiceState) of allowed values with initial defined states of:

- Normal: The service is operating normally.
- Unmanned: (applies to PSAPs only) The PSAP has indicated that it is not currently answering calls.
- ScheduledMaintenance (down): The service is undergoing maintenance activities and is not accepting service requests.
- ScheduledMaintenance (available): The service is undergoing maintenance activities, but will respond to service requests, possibly with reduced availability.
- MajorIncidentInProgress: The element is operating normally but is handling a major incident and may be unable to accept some requests.
- PartialService: Processing some requests, but response may be delayed.
- Overloaded: The service is completely overloaded.
- GoingDown: The service is being taken out of service.
- Down: The service is unavailable.

In addition, if the subscriber to a service is unable to contact that service, it may show the state of the service as "Unreachable".

NOTE: One or more elements may implement a service. Each element would have its own element state; the service would have an independent state.

6.3.5.2 Parameter

Event Package Name: emergency-ServiceState

Event Package Parameters: None

SUBSCRIBE Bodies: Standard IETF RFC 4661 [30] + extensions filter specification may be present

Subscription Duration: Default 1 hour. One (1) minute to 24 hours is reasonable

NOTIFY Bodies: MIME type application/emergency.ServiceState+json

Table 13

Parameter	Condition	Description
service	MANDATORY	
name	MANDATORY	Name of service
domain	MANDATORY	Service domain
serviceState	MANDATORY	
state	MANDATORY	Enumeration of current state from ServiceState registry
reason	MANDATORY	Text containing the reason state was changed, if available

Notifier Processing of SUBSCRIBE Requests:

The notifier consults the policy (ServiceState) to determine if the requester is permitted to subscribe. It returns `603 Decline` if not acceptable. If the request is acceptable, it returns `200 OK` and shall immediately send a NOTIFY with the current state. Notifiers shall implement event rate filters, IETF RFC 6446 [49].

Notifier Generation of NOTIFY Requests:

When the state of the service changes, a new NOTIFY request is generated, adhering to the filter requests. Filter requests may specify a minimum notification interval. The element shall generate a NOTIFY meeting this filter, if specified. This can be used as a watchdog mechanism.

Subscriber Processing of NOTIFY Requests:

No specific action required.

Handling of Forked Requests:

Forking is not expected to be used with this package.

Rate of Notification:

State normally does not change rapidly. Changes may occur in tens of seconds if the network or systems are unstable.

State Agents:

No special handling is required.

6.4 Mapping Services

6.4.1 Find Service (LOST-1)

6.4.1.1 Overview

All SIP-based emergency calls pass location information either by value (PIDF-LO) or by reference (Location URI) plus a Service URN to an Emergency Service Routing Proxy (ESRP) to support routing of emergency calls. The ESRP passes the Service URN and location information via the LoST interface (as defined in IETF RFC 5222 [39]) to an Emergency Call Routing Function (ECRF), which determines the next hop in routing a call to the requested service. Implementation and deployment of a national LoST hierarchy is subject to national regulation.

NOTE 1: If an element using LoST receives location by reference, it dereferences the URI to obtain the value prior to querying the LoST server. The LoST server does not accept location by reference.

The ECRF (see clause 5.3) performs the mapping of the call's location information and requested Service URN to a "PSAP URI" by querying its data and then returning the URI provided. Using the returned URI and other information (time-of-day, PSAP state, etc.), the ESRP may then apply a PRF policy to determine the appropriate routing URI.

The service URN used to query the ECRF by an ESRP is obtained by provisioning of the "origination policy" of the queue that the call is received on at the ESRP (see clause 5.2). The response of the ECRF is determined by provisioning of the service boundary layers, which specify the URN they apply to. Thus, ECRFs (and ESRPs) are not hard coded with any specific URNs.

A single emergency call can be routed by one or more ESRPs within the ESInet, resulting in use of the LoST interface once per hop as well as once by the terminating PSAP.

NOTE 2: The term "PSAP URI" is used within the LoST protocol definition to refer to the URI returned from the service URN "urn:service:sos". The URI returned may not be that of a PSAP, but instead may route to a BCF or ESRP.

LoST (IETF RFC 5222 [39]) is the protocol that is used for several functions:

- Call routing: LoST is used by the ECRF as the protocol to route all emergency calls both to and within the ESInet.
- Retrieving lists of services available at a location.

The normative reference that defines the protocol is IETF RFC 5222 [39]. The text in this clause that defines LoST protocol operations should be considered informative, and any discrepancies are resolved by IETF RFC 5222 [39] text. The text below does contain limitations and specific application of LoST operations that are normative.

6.4.1.2 findService Request

The "civic" and "geodetic-2d" profiles are baseline profiles defined in IETF RFC 5222 [39] and emergency calls are expected to use only these profiles. Conformant LoST servers need not support any location profiles beyond these baseline profiles.

The LoST interface allows a geo-location to be expressed as a point or one of a number of defined "shapes" such as circle, ellipse, arcband or polygon. ECRFs **s**hall be able to handle points and all of these shapes.

The "service" element identifies the service requested by the client. Valid service names shall be `urn:service:sos` or one of its sub-services for ECRF queries used by entities or devices for emergency calls. ECRF implementations may support additional service names used internal to an ESInet dependent on the provisioning of service boundary layers in a geographical information system.

Entities inside the ESInet shall request recursion by setting the recursive attribute in the `<findService>` request to true, if required.

6.4.1.3 findService Response

An ECRF servers may operate in recursive mode or iterative mode if the server being queried is not authoritative for the location supplied.

The use of recursion by the ECRF initiates a query on behalf of the requestor that propagates through other ECRFs to an authoritative ECRF that returns the PSAP URI back through the intervening ECRFs to the requesting ECRF.

The use of iteration by the ECRF simply returns a domain name of the next ECRF to contact.

The ECRF may operate in a recursive mode or an iterative mode, depending on local provisioning and the value of the 'recursive' attribute of the `<findService>` request. All ECRF implementations shall support both recursive and iterative modes.

When the ECRF successfully processes a LoST `<findService>` message, it returns a LoST `<findServiceResponse>` message containing a `<mapping>` element that includes the "next hop" ESRP or PSAP URI in the `<uri>` element. If the ECRF cannot successfully process a LoST `<findService>` message, it returns a LoST `<errors>` message indicating the nature of the error or a LoST `<redirect>` message indicating the ECRF that can process the `<findService>` message.

The `<uri>` returned specifies either the next hop URI of the PSAP or the ESRP that is appropriate for the location sent in the query message. This shall be a globally routable URI with a `sip` scheme for `urn:service:sos` requests. Some other service URNs may return values with HTTP/HTTPS schemes. LoST servers should return SIPS and HTTPS URIs in addition to the SIP and HTTP (where appropriate) URIs.

The 'expires' attribute in the `<mapping>` element provides an ECRF with a way to control load, balancing that against the time required to completely implement a routing change when circumstances require. By increasing the expiration time, fewer queries to the server may be received if upstream LoST servers or clients implement caching.

The LoST response contains `<via>` elements in the `<path>` element that name the LoST servers visited to obtain the answer. Vias shall be returned to be compliant with IETF RFC 5222 [39] and are essential for use in error resolution and loop detection.

The `<displayName>` element of the <mapping> response is a text string that provides an indication of the serving agency(ies) for the location provided in the query. This information might be useful to PSAPs that query an ECRF.

The `<service>` element in the query identifies the service for which this mapping is valid.

The `<serviceNumber>` element in the `<mapping>` response contains the emergency services number appropriate for the location provided in the query. This allows a foreign end device to recognize a dialled emergency number.

If the ECRF is configured to allow it, a requesting entity can obtain the boundary of the service area handled by the requested service, returned in the `<serviceBoundary>` element of `<mapping>`. This is most useful for mobile devices that use geodetic coordinates since they can track their location. When they leave the service area, they can send another `<findService>` request to determine the proper service area for their new location and avoid re-querying the ECRF as long as they are within the returned boundary.

The service boundary in a <mapping> may be returned by value or by reference, or not at all, at the discretion of the server. If the server returns a service boundary reference, the client may then obtain the actual service boundary with a `<getServiceBoundary>` request. A service boundary represented by a given reference can never change, so a client only needs to retrieve the boundary value a single time.

Future mappings returned by the server and having the same service boundary may reuse the reference, eliminating the need to transmit the boundary value again. Devices handling service boundaries may be limited in processing power and battery capacity, and thus sending complex polygons should be avoided.

Devices may have to handle a polygon with more than a few points when the device is very close to an edge where the mapping will be different. Because a service boundary is not needed to initiate an emergency call, and because a complex boundary may be quite large, ECRFs shall be configured to return geodetic service boundaries by reference. Devices querying an ECRF in order to immediately initiate an emergency call should not attempt to obtain the service boundary by value.

The `<locationValidation>` element in `<findServiceResponse>` identifies which elements of the received civic address were "valid" and used for mapping, which were "invalid" and which were "unchecked" when validation is requested. Since the ECRF is not responsible for performing validation, this parameter may not be returned, subject to local implementations. If an element is unable to provide information based on the received request message, it shall return an error message as defined in clause 6.4.5.

6.4.1.4 Transport Layer Security

LOST-1 message exchange within the ESInet shall be TCP with TLS as defined in IETF RFC 8446 [57]. TLS version shall be 1.3 or higher and based on the cipher suites specified in clause C.5. If TLS 1.3 is not supported, fallback to TLS 1.2 is allowed. TLS implementations shall support mutual authentication, which implies both ends have an X.509 certificate available to the other party. How a certificate is created and issued by a Certificate Authority (CA) is out of scope of the present document.

6.4.2 Service Boundary (LOST-2)

6.4.2.1 Overview

A `<findServiceResponse>` can return a globally unique identifier in the 'serviceBoundary' attribute that can be used to retrieve the service boundary, rather than returning the boundary by value.

6.4.2.2 getServiceBoundary Request

If an element returns a service boundary by reference, it shall handle `<getServiceBoundary>` requests as defined in IETF RFC 5222 [39].

6.4.2.3 getServiceBoundary Response

The response to a `<getServiceBoundary>` request shall be constructed as defined in IETF RFC 5222 [39]. If an element is unable to provide information based on the received request message, it shall return an error message as defined in clause 6.4.5.

6.4.2.4 Transport Layer Security

LOST-2 message exchange within the ESInet shall be TCP with TLS as defined in IETF RFC 8446 [57]. TLS version shall be 1.3 or higher and based on the cipher suites specified in clause C.5. If TLS 1.3 is not supported, fallback to TLS 1.2 is allowed. TLS implementations shall support mutual authentication, which implies both ends have an X.509 certificate available to the other party. How a certificate is created and issued by a Certificate Authority (CA) is out of scope of the present document.

6.4.3 List Services (LOST-3)

6.4.3.1 Overview

A client can ask a LoST server for the list of services that it understands, primarily for diagnostic purposes. The query does not contain location information, as it simply provides an indication of which services the server can look up, not whether a particular service is offered for a particular area.

6.4.3.2 listServices Request

The response to a `<listServices>` request may depend on the credentials of the querier. A query with no `<service>` element in the request should result in all top level services being returned in the response (e.g. `urn:service:sos`). A query with `<service>` specified as `urn:service:sos` should result in all the subservices of `sos` (`urn:service:sos.police`, `urn:service:sos.fire`, …) that are available in the jurisdiction being returned in the response.

6.4.3.3 listServices Response

The response to a `<listServices>` request shall be constructed as defined in IETF RFC 5222 [39]. If an element is unable to provide information based on the received request message, it shall return an error message as defined in clause 6.4.5.

6.4.3.4 Transport Layer Security

LOST-3 message exchange within the ESInet shall be TCP with TLS as defined in IETF RFC 8446 [57]. TLS version shall be 1.3 or higher and based on the cipher suites specified in clause C.5. If TLS 1.3 is not supported, fallback to TLS 1.2 is allowed. TLS implementations shall support mutual authentication, which implies both ends have a certificate available to the other party. How a certificate is created and issued by a Certificate Authority (CA) is out of scope of the present document.

6.4.4 List Services by Location (LOST-4)

6.4.4.1 Overview

A client can ask a LoST server for the list of services it knows about for a particular area. The query contains one or more `<location>` elements and may contain the `<service>` element.

6.4.4.2 listServicesByLocation Request

The response to a <listServicesByLocation> request may depend on the credentials of the querier. A query with no <service> element in the request should result in all top level services being returned in the response (e.g. urn:service:sos). A query with <service> specified as urn:service:sos should result in all the subservices of sos (urn:service:sos.police, urn:service:sos.fire, …) that are available in the jurisdiction being returned in the response.

Entities inside the ESInet shall specify recursion by setting the recursive attribute in the <listServicesByLocation> request to true.

6.4.4.3 listServicesByLocation Response

The response to a <listServicesByLocation> request shall be constructed as defined in IETF RFC 5222 [39]. If an element is unable to provide information based on the received request message, it shall return an error message as defined in clause 6.4.5.

6.4.4.4 Transport Layer Security

LOST-4 message exchange within the ESInet shall be TCP with TLS as defined in IETF RFC 8446 [57]. TLS version shall be 1.3 or higher and based on the cipher suites specified in clause C.5. If TLS 1.3 is not supported, fallback to TLS 1.2 is allowed. TLS implementations shall support mutual authentication, which implies both ends have an X.509 certificate available to the other party. How a certificate is created and issued by a Certificate Authority (CA) is out of scope of the present document.

6.4.5 Error Responses

- <badRequest> Element: This element indicates the ECRF could not parse or otherwise understand the request sent by the requesting entity (e.g. the XML is malformed).

- <forbidden> Element: This element indicates an ECRF refused to send an answer. This generally only occurs for recursive queries, namely, if the client tried to contact the authoritative server and was refused.

- <internalError> Element: This element indicates the ECRF could not satisfy a request due to a bad configuration or some other operational and non-LoST protocol-related reason.

- <locationProfileUnrecognized> Element: None of the profiles in the request were recognized.

- <locationInvalid> Element: This element indicates the ECRF determined the geodetic or civic location is invalid (e.g. geodetic latitude or longitude value is outside the acceptable range). The only time this would normally be returned is if there was a malformed location such as a geodetic profile="geodetic-2d" and <civicAddress> element present. If there is no authoritative server for the location, that would be coded as <notFound>.

- <SRSInvalid> Element: This element indicates the ECRF does not recognize the spatial reference system (SRS) specified in the <location> element or it does not match the SRS specified in the profile attribute (e.g. not WGS84 2D, EPSG Code 4326 for profile="geodetic-2d").

NOTE: This error is not present in the IETF RFC 5222 [39] schema, has been reported as an erratum, and thus may not be implemented by all LoST servers or clients. Use of this error may be problematic.

- <loop> Element: During a recursive query, the server was about to visit a server that was already in the server list in the <path> element indicating a request loop.

- <notFound> Element: The ECRF could not find an answer to the query. This would occur if the authoritative server cannot find the location and has no applicable default route, or if no authoritative server exists.

- <serverError> Element: An answer was received from another LoST server, but it could not be parsed or otherwise understood. This error occurs only for recursive queries.

- `<serverTimeout>` Element: This element indicates the ECRF timed out waiting for a response (e.g. another ECRF for a recursive query, etc.).
- `<serviceNotImplemented>` Element: This element indicates the ECRF detected the requested service URN is not implemented and it found no substitute for it. This normally would not occur for a service beginning `urn:service:sos`.

6.5 Location Services

6.5.1 HTTP Enabled Location Delivery (HELD-1)

6.5.1.1 Overview

All elements in an ESInet that use location by reference implement HTTP Enabled Location Delivery (HELD) dereferencing protocols as defined in IETF RFC 5985 [45].

6.5.1.2 Location Request

Location requests may be used for location configuration, where a device may identify itself by a specific identity parameter, or a third-party element (BCF, ESRP or PSAP), if authorized, requests location information. Entities that request location shall support the use of device identity in HELD as defined in IETF RFC 6155 [47].

6.5.1.3 Location Response

Location in a response shall be represented by content in a PIDF-LO document (as defined in IETF RFC 4119 [22]). All geodetic data shall use WGS84 as the datum. The representation of the location object within the PIDF document shall utilize the 'tuple' element as defined in IETF RFC 4119 [22].

6.5.1.4 Error Responses

If an element is unable to provide location information based on the received request message, it shall return an error message as defined in IETF RFC 5985 [45].

6.5.1.5 Transport Layer Security

HELD-1 message exchange within the ESInet shall be TCP with TLS as defined in IETF RFC 8446 [57]. TLS version shall be 1.3 or higher and based on the cipher suites specified in clause C.5. If TLS 1.3 is not supported, fallback to TLS 1.2 is allowed. TLS implementations shall support mutual authentication, which implies both ends have an X.509 certificate available to the other party. How a certificate is created and issued by a Certificate Authority (CA) is out of scope of the present document.

6.5.2 Location Dereference (HELD-2)

An element needing location that has a HELD URI shall dereference per IETF RFC 6753 [52].

6.5.3 Location URI (HELD-3)

6.5.3.1 Overview

The SIP Presence Event Package IETF RFC 3856 [17] implementing the SIP Presence SUBSCRIBE/NOTIFY mechanism can control repeated dereferencing, especially when tracking of the caller is needed.

6.5.3.2 Subscription

Using SIP Presence, the entity desiring location shall subscribe to the SIP Presence Event Package (IETF RFC 3856 [17]) at the location URI provided.

The SUBSCRIBE shall contain an Expires header (IETF RFC 3261 [5]) which represents the subscribers requested expiration, and the 2XX response contains one that represents the server's actual expiration (which may be shorter, but not longer, than the subscriber's requested time).

6.5.3.3 Notification

Location updates shall trigger NOTIFY transactions (IETF RFC 6665 [51]) containing a PIDF document that will include the location in the Location Object (LO) part, forming the PIDF-LO. An immediate NOTIFY shall be generated upon acceptance of a subscription request.

An Expires of zero indicates a request for exactly one NOTIFY (that is the current location) with no further updates.

The subscribing element may limit how often further NOTIFYs are sent (before expiration of the subscription) using a filter (IETF RFC 4661 [30]). Rate limits (IETF RFC 6446 [49]) and Location filters (IETF RFC 6447 [50]) are useful for this application and should be supported if a SIP location URI is supplied.

6.6 Media

6.6.1 RTP Transport (RTP-1)

All media processing elements shall support media using RTP as defined in IETF RFC 3550 [13]. Each SIP session initiation message or response should describe the media the User Agent can support using the Session Description Protocol (SDP) in the body of the message as defined in IETF RFC 4566 [25]. Support of any type of media (e.g. voice, video, text) in originating networks is based on regulatory requirements or business decisions.

All media processing elements should implement media security with SRTP as defined in IETF RFC 3711 [15] and SDES as defined in IETF RFC 4568 [26]. SRTP security should be requested in all calls originated within an ESInet. RTCP as defined in IETF RFC 3550 [13] shall be, and SRTCP as defined in IETF RFC 3711 [15] should be supported within the ESInet.

PSAPs shall detect the presence of RTP streams so they can distinguish RTP failure from real silence by the caller. Elements that detect the loss of RTP should attempt to re-establish the streams by sending re-INVITE to the other party. If that fails, the device should indicate a failure and require the user (call taker in most cases) to act such as initiating disconnect.

6.6.2 RTP Types (RTP-2)

6.6.2.1 General

Besides the definitions in the following clauses, the media included in the emergency call may be varied according to the description ETSI TS 126 114 [i.4].

6.6.2.2 Audio

All audio processing entities in the ESInet shall support G.711 µ-law and a-law (Recommendation ITU-T G.711 [61]). AMR, AMR-WB, EVRC (IETF RFC 3558 [14]), EVRC-B (IETF RFC 4788 [31]), EVRC-WB [37]), and EVRC-NW (IETF RFC 6884 [55] should be supported.

6.6.2.3 Video

All video processing entities in the ESInet shall support video compression format H.264/MPEG-4 Version 10 baseline profile including levels 1-3. Further, such entities shall support both IETF RFC 5104 [35] and IETF RFC 5168 [36] for full frame refresh requests utilizing the Real-time Transport Control Protocol (RTCP) method. Video processing entities may fall back to the SIP INFO method (IETF RFC 5168 [36]) when the sender does not implement the RTCP method. In any case, elements shall attempt to maintain 30 frames per second video if offered by the sender, refer to RTP/AVPF (IETF RFC 4585 [28]).

6.6.2.4 Real-time Text

All call handling elements in the ESInet shall support the framework for Real-time Text over IP using the Session Initiation Protocol (SIP), as in IETF RFC 5194 [38], specifying the use of IETF RFC 4103 [21] for the packetization of real-time text, and enhancements of real-time text for mixing in a centralized conference model as in IETF RFC 9071 [62]. This medium may be used simultaneously with voice and/or video in calls. Refer to ETSI TS 101 470 [i.2] and ETSI TR 103 201 [i.3] and for further information on Real-Time Text and Total Conversation to Emergency Services.

6.7 Instant Messaging (IM-1)

PSAPs shall be able to receive IM as a series of individual MESSAGE transactions within and out of a SIP dialog (non-session-mode). Location shall be included in a geolocation header in the MESSAGE method as with any other emergency call. Out of dialog MESSAGEs received from the same caller within a configurable time (2-3 minutes nominally) should be considered part of the same IM session, and therefore routed to the same PSAP (and the same call taker), regardless of movement of the caller while texting.

Out of dialog MESSAGE transactions may contain a Call-Info header field value with the purpose of grouping messages into the same IM session (e.g. a namespace referring to the beginning and end of an IM session). Specification of such a namespace is out of scope of the present document.

All call handling elements within the ESInet shall support Session Initiation Protocol (SIP) Extension for Instant Messaging (IETF RFC 3428 [11]), Indication of Message Composition for Instant Messaging (IETF RFC 3994 [20]), The Message Session Relay Protocol (MSRP) (IETF RFC 4975 [32]) and Relay Extension for the Message Session Relay Protocol (MSRP) (IETF RFC 4976 [33]).

NOTE: All elements support instant messaging using the specifications in the present document. Any given origination network or device may not support instant messaging, and support of instant messaging by origination networks and devices may be subject to regulation.

6.8 Common Alerting Protocol (CAP-1)

Non-human-associated calls are non-interactive calls originated by an automated sensor-based device. Such calls contain data (e.g. sensor data). There may be streaming media (e.g. video or audio feeds) and a capability to control the device or another device. In general, there is no assumption of a human presence.

Common Alerting Protocol (CAP) [58] messages are used for events sent to, and within an ESInet. For use within ESInets, elements sending or receiving CAP messages shall have a common understanding of what kind of an event is being sent, primarily to use in routing decisions. Further definitions of CAP events or messages in general is out of scope of the present document.

Non-human-associated calls are presented to an ESInet in the same way as regular emergency calls using a SIP INVITE. If these calls only carry data (data-only emergency calls) then the considerations in clause 6.1.2.2 are applicable. This means that the SIP message contains a CAP payload. The additional data structure may provide further information about the call, caller, and location.

Non-human-associated calls are routed and handled the same as voice, video or text calls throughout the ESInet. The routing mechanisms can route non-human-associated calls differently from voice calls in the same way they can route video calls differently from voice calls. The parameters in the CAP message are available to the routing function as inputs to direct calls with specified characteristics to specific entities.

Annex A (normative): JSON Schema

A.1 QueueState

```
{
  "definitions": {},
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "https://forge.etsi.org/rep/emtel/ts-103-479/json-schema/blob/v1.2.1/queuestate.json",
  "type": "object",
  "title": "QueueState",
  "description": "QueueState event notification",
  "required": [
    "queueState"
  ],
  "properties": {
    "queueState": {
      "type": "object",
      "required": [
        "queueUri",
        "queueLength",
        "queueMaxLength",
        "state"
      ],
      "properties": {
        "queueUri": {
          "type": "string",
          "description": "The SIP URI of the queue"
        },
        "queueLength": {
          "type": "integer",
          "description": "Indicating current number of calls on the queue",
          "minimum": 0
        },
        "queueMaxLength": {
          "type": "integer",
          "description": "Integer indicating maximum length of queue",
          "minimum": 0
        },
        "state": {
          "type": "string",
          "enum": [
            "active",
            "inactive",
            "disabled",
            "full",
            "standby"
          ],
          "description": "Enumeration of current queue state (e.g., active/inactive/disabled)"
        }
      }
    }
  }
}
```

A.2 AbandonedCall

```
{
  "definitions": {},
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "https://forge.etsi.org/rep/emtel/ts-103-479/json-schema/blob/v1.2.1/abandonedcall.json",
  "type": "object",
  "title": "Abandoned call NOTIFY",
  "description": "AbandonedCall event notification",
  "required": [
    "abandonedCall"
  ],
  "properties": {
    "abandonedCall": {
      "type": "object",
      "required": [
        "invite",
        "inviteTimestamp",
        "cancelTimestamp"
      ],
      "properties": {
        "invite": {
          "type": "string",
          "description": "Content of INVITE message"
        },
        "inviteTimestamp": {
          "type": "string",
          "description": "Timestamp call was received at ESRP"
        },
        "cancelTimestamp": {
          "type": "string",
          "description": "Timestamp CANCEL was received at ESRP"
        }
      }
    }
  }
}
```

A.3 SecurityPosture

```
{
  "definitions": {},
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "https://forge.etsi.org/rep/emtel/ts-103-479/json-schema/blob/v1.2.1/securityposture.json",
  "type": "object",
  "title": "SecurityPosture",
  "description": "SecurityPosture event notification",
  "required": [
    "service",
    "securityPosture"
  ],
  "properties": {
    "service": {
      "type": "object",
      "required": [
        "name",
        "domain"
      ],
      "properties": {
        "name": {
          "type": "string"
        },
        "domain": {
          "type": "string"
        }
      }
    },
    "securityPosture": {
      "type": "object",
      "required": [
        "posture"
      ],
      "properties": {
        "posture": {
          "type": "string",
          "enum": [
            "green",
            "yellow",
            "orange",
            "red"
          ],
         "description": "Enumeration of current security posture"
        }
      }
    }
  }
}
```

A.4 ElementState

```
{
  "definitions": {},
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "https://forge.etsi.org/rep/emtel/ts-103-479/json-schema/blob/v1.2.1/elementstate.json",
  "type": "object",
  "title": "ElementState",
  "description": "ElementState event notification",
  "required": [
    "elementState"
  ],
  "properties": {
    "elementState": {
      "type": "object",
      "required": [
        "elementId",
        "state",
        "reason"
      ],
      "properties": {
        "elementId": {
          "type": "string"
        },
        "state": {
          "type": "string",
         "enum": [
           "normal",
           "sheduledMaintenance",
           "serviceDisruption",
           "overloaded",
           "goingDown",
           "down"
         ],
         "description": "Enumeration of current element state"
        },
        "reason": {
          "type": "string"
        }
      }
    }
  }
}
```

A.5 ServiceState

```
{
  "definitions": {},
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "https://forge.etsi.org/rep/emtel/ts-103-479/json-schema/blob/v1.2.1/servicestate.json",
  "type": "object",
  "title": "ServiceState",
  "description": "ServiceState event notification",
  "required": [
    "service",
    "serviceState"
  ],
  "properties": {
    "service": {
      "type": "object",
      "required": [
        "name",
        "domain"
      ],
      "properties": {
        "name": {
          "type": "string"
        },
        "domain": {
          "type": "string"
        }
      }
    },
    "serviceState": {
      "type": "object",
      "required": [
        "state",
        "reason"
      ],
      "properties": {
        "state": {
          "type": "string",
          "enum": [
            "normal",
            "unmanned",
            "sheduledMaintenance(down)",
            "sheduledMaintenance(available)",
            "majorIncidentInProgress",
            "partialService",
            "overloaded",
            "goingDown",
            "down"
          ],
          "description": "Enumeration of current service state"
        },
        "reason": {
          "type": "string",
          "description": "Text containing the reason state was changed, if available"
        }
      }
    }
  }
}
```

A.6 Dequeue Registration Request

```
{
  "definitions": {},
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "https://forge.etsi.org/rep/emtel/ts-103-479/json-schema/blob/v1.2.1/dqregrequest.json",
  "type": "object",
  "title": "DequeueRegistration Request",
  "description": "Dequeue registration is a web service whereby the registering entity becomes one
of the dequeuing entities",
  "required": [
    "queueUri",
    "dequeuerUri",
    "expirationTime"
  ],
  "properties": {
    "queueUri": {
      "type": "string",
      "description": "SIP URI of queue to register on"
    },
    "dequeuerUri": {
      "type": "string",
      "description": "SIP URI of dequeuer (where to send calls)"
    },
    "expirationTime": {
      "type": "integer",
      "description": "Requested time in seconds this registration will expire"
    },
    "dequeuePreference": {
      "type": "integer",
      "description": "Integer from 1-5 indicating queuing preference; 5 indicating highest
preference",
      "minimum": 1,
      "maximum": 5
    }
  }
}
```

A.7 Dequeue Registration Response

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "https://forge.etsi.org/rep/emtel/ts-103-479/json-schema/blob/v1.2.1/dqregresponse.json",
  "title": "DequeueRegistration Response",
  "description": "Dequeue registration is a web service whereby the registering entity becomes one
of the dequeuing entities",
  "type": "object",
  "required": [
    "expirationTime"
  ],
  "properties": {
    "expirationTime": {
      "description": "Time in seconds this registration will expire",
      "type": "integer"
    }
  }
}
```

A.8 BadActor Service

```
openapi: 3.0.1
info:
  title: Bad Actor Service
  version: "1.0"
  contact:
    name: ETSI TC EMTEL
    url: https://forge.etsi.org/rep/emtel/ts-103-479/json-schema/blob/v1.2.1/badactor.yml
servers:
  - url: http://localhost/BadActor/v1
paths:
  /BadActors:
    post:
      tags:
        - BadActorRequest
      summary: Identifies a source as a "Bad Actor"
      operationId: BadActorRequest
      requestBody:
        description: Bad actor source Id
        content:
          application/json:
            schema:
              type: string
        required: true
      responses:
        '201':
          description: Bad Actor successfully added
        '401':
          description: Unauthorized
        '432':
          description: Already reported
        '433':
          description: No such sourceId
        '454':
          description: Unspecified Error
  /Versions:
    servers:
      - url: https://api.example.com/BadActor
        description: Override base path for Versions query
    get:
      tags:
        - RetrieveVersions
      summary: Retrieves all supported versions, vendor parameter is optional.
      operationId: RetrieveVersions
      responses:
        '200':
          description: Versions found
          content:
            application/json:
              schema:
                $ref: 'i3-common.yaml#/components/schemas/VersionsArray'
```

A.9 BadActor Response

```
void
```

Annex B (informative): Organizational Descriptions

B.0 General

This clause provides a summary of the organizations described in the present document.

B.1 Certificate Authority

A Certificate Authority (CA) that issues certificates to different entities in the emergency services networks has to be created or the services of an existing CA have to be re-used. This enables proper authentication and builds the foundation for authorization. The overall level of security will be substantially improved, therefore.

Since the present document assumes a public key infrastructure the use of such a certificate authority for usage with emergency services organizations is needed. Note that a CA is responsible for managing the entire lifecycle of certificates from the creation to termination or revocation.

B.2 National, and Regional Authorities

Applicable laws, regulations and rules may need to be enhanced to support ESInet deployment. This is particularly true to provide the necessary provisions to require access network providers to share IP location information and VSPs/ASPs to transmit emergency calls to emergency services authorities.

B.3 Public Safety Computer Emergency Response Team (CERT)

To react to security breaches and other incidents the creation of a Public Safety Computer Emergency Response Team (CERT) is anticipated, and all stakeholders are obliged to make any necessary preparations to receive alerts from the CERT and to respond. It is essential that all organizations have trained staff available 24 × 7 × 365 to immediately respond to attacks and have the capability and training to be able to mitigate such attacks.

B.4 ETSI Protocol Naming and Numbering Service (PNNS)

ETSI CTI provides a Protocol Naming and Numbering Service for all ETSI Technical Bodies. Many protocols require the allocation of globally unique names or numbers to interoperate successfully. Ranges of names or numbers are often allocated to standards bodies to distribute the task of allocation, while still maintaining global uniqueness. ETSI CTI manages such name and number ranges for ETSI.

B.5 Emergency Call Service Authorities

The national/regional/local authorities are responsible for overall operation of, and the data for the emergency communication system. Such an authority:

- oversees operating the state/regional/local Emergency Service Routing Proxy (ESRP);
- provides Emergency Call Routing Function (ECRF) and Location Information Service (LIS);

- is responsible for maintaining the integrity of the data housed in the ECRF systems;
- also provides input to the definition of policies, which dictates the granularity of the routing decisions returned by the ECRF (i.e. ESRP URIs vs. PSAP URIs);
- provides data about PSAP boundaries. This data is, for example, using in LoST servers and influences routing decisions;
- is responsible to address issues caused by gaps and overlaps in these boundaries;
- ensures that BCFs are accessible from the Internet so that VSPs and ASPs can route emergency calls to them;
- is responsible to provide an authoritative GIS database containing only valid in formation, where civic addresses are used for the location validation;
- decides about the setup, and operation of the ESInet as well as PSAPs and other IT infrastructure equipment necessary to operate the IP network, interconnection points, and call routing equipment.

Annex C (informative): Parameter Registries

C.0 General

The present document requires several registries to be created and those populated with initial values. The entity that creates these values and makes them available over the Web is called ETSI Protocol Naming and Numbering Service (PNNS). ETSI PNNS ensures that the policies associated with the parameter registries are followed to avoid inconsistency in the registry and is available at https://portal.etsi.org/PNNS/Protocol-Specification-Allocation/Emergency-services.

The registry created for the present document is considered as temporary and may change in the course of the Internet Assigned Numbers Authority (IANA) registry creation process.

C.1 queueState Registry

C.1.1 General

QueueState is an event that indicates to an upstream entity the state of a queue. A registry is needed to enumerate the possible values returned.

C.1.2 Name

The name of this registry is queueState.

C.1.3 Information required to create a new value

A new entry to queueState requires an explanation of when value will be returned and how it is differentiated from other values in the registry.

C.1.4 Management Policy

A Technical Document is required to add a new entry into the registry.

C.1.5 Content

This registry contains:

- The UTF-8 "Value" of the entry.
- The UTF-8 "Purpose" of the entry and when it should be used.
- A reference (URI) to the Technical Document that defines the label.

C.1.6 Initial Values

The initial value and purposes of the registry are found in the present document.

C.2 securityPosture Registry

C.2.0 General

The SecurityPosture event returns an enumerated value of the current security posture of an agency or element. A registry is needed to enumerate the possible values returned.

C.2.1 Name

The name of this registry is securityPosture.

C.2.2 Information required to create a new value

A new entry to securityPosture requires an explanation of when value will be returned and how it is differentiated from other values in the registry.

C.2.3 Management Policy

A Technical Document is required to add a new entry into the registry.

C.2.4 Content

This registry contains:

- The UTF-8 "Value" of the entry.
- The UTF-8 "Purpose" of the entry and when it should be used.
- A reference (URI) to the Technical Document that defines the label.

C.2.5 Initial Values

The initial value and purposes of the registry are found in the present document.

C.3 elementState Registry

C.3.0 General

The elementState event returns an enumerated value of the current state of an agency or element. A registry is needed to enumerate the possible values returned.

C.3.1 Name

The name of this registry is elementState.

C.3.2 Information required to create a new value

A new entry to elementState requires an explanation of when value will be returned and how it is differentiated from other values in the registry.

C.3.3 Management Policy

A Technical Document is required to add a new entry into the registry.

C.3.4 Content

This registry contains:

- The UTF-8 "Value" of the entry.
- The UTF-8 "Purpose" of the entry and when it should be used.
- A reference (URI) to the Technical Standard that defines the label.

C.3.5 Initial Values

The initial value and purposes of the registry are found in the present document.

C.4 serviceState Registry

C.4.0 General

The serviceState event returns an enumerated value of the current state of a service. A registry is needed to enumerate the possible values returned.

C.4.1 Name

The name of this registry is serviceState.

C.4.2 Information required to create a new value

A new entry to serviceState requires an explanation of when value will be returned and how it is differentiated from other values in the registry.

C.4.3 Management Policy

A Technical Document is required to add a new entry into the registry.

C.4.4 Content

This registry contains:

- The UTF-8 "Value" of the entry.
- The UTF-8 "Purpose" of the entry and when it should be used.
- A reference (URI) to the Technical Document that defines the label.

C.4.5 Initial Values

The initial value and purposes of the registry are found in the present document.

C.5 Cipher Suites

C.5.0 General

A cipher suite is a set of algorithms that help secure a network connection. The suites typically use Transport Layer Security (TLS).

C.5.1 Recommended TLS 1.3 Cipher Suites

Cipher	Encryption	MAC
TLS_AES_128_GCM_SHA256	AESGCM(128)	AEAD
TLS_AES_256_GCM_SHA384	AESGCM(256)	AEAD
TLS_CHACHA20_POLY1305_SHA256	CHACHA20/POLY1305(256)	AEAD

C.5.2 Acceptable TLS 1.2 Cipher Suites

Cipher	Encryption	MAC
ECDHE-ECDSA-AES128-GCM-SHA256	AESGCM(128)	AEAD
ECDHE-RSA-AES128-GCM-SHA256	AESGCM(128)	AEAD
ECDHE-ECDSA-AES256-GCM-SHA384	AESGCM(256)	AEAD
ECDHE-RSA-AES256-GCM-SHA384	AESGCM(256)	AEAD
ECDHE-ECDSA-CHACHA20-POLY1305	CHACHA20/POLY1305(256)	AEAD
ECDHE-RSA-CHACHA20-POLY1305	CHACHA20/POLY1305(256)	AEAD
DHE-RSA-AES128-GCM-SHA256	AESGCM(128)	AEAD
DHE-RSA-AES256-GCM-SHA384	AESGCM(256)	AEAD

Annex D (informative): Change History

Date	Version	Information about changes
January 2019	0.1	Initial version of the first draft
April 2019	0.4	Edits and Annexes
July 2019	0.7	Namespace and Registry
October 2019	0.9	Final edits
November 2019	0.14	JSON schema
January 2023	1.1.6	Final draft submitted to EMTEL#56
January 2023	1.1.7	Final draft including agreed changes from EMTEL#56 and small edits in yellow in clause 6.1.2.2 and Table 2 as discussed during the Plugtests #5. Ready to go for approval before publication.

History

Document history		
V1.1.1	December 2019	Publication
V1.2.1	March 2023	Publication