ETSI TS 103 478 V1.2.1 (2020-03)

TECHNICAL SPECIFICATION

Emergency Communications (EMTEL); Pan-European Mobile Emergency Application

Reference

RTS/EMTEL-00048

Keywords

application, emergency

ETSI

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

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

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

Important notice

Copyright Notification

Contents

Intellectual Property Rights

Essential patents

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

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

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.

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 Pan-European Mobile Emergency Application (PEMEA) architecture provides the requirements and architecture for a solution to provide emergency application interconnection. It specifies the protocols and procedures enabling interoperable implementations of the architecture and provides extension points to enable new communication mechanisms as they evolve.

Introduction

The rise of smart devices such as smart phones, tablets and laptops has led to an explosion in communications applications. Many of these applications aim to supplement existing communications services, such as providing caller and location information for emergency calls, while others seek to provide alternative communication mechanisms such as total conversation and instant messaging for example. Many of these applications already exist in limited local capacities but lack a common framework for easy interconnection. This limitation prohibits a user's application operating in a region other than the one it was developed in and having his/her information and accurate location information passed to the PSAP serving their location. The Pan-European Mobile Emergency Application (PEMEA) architecture provides a solution to interconnect these applications.

1 Scope

The present document is divided into two parts. The first part provides the requirements and functional architecture while the second part provides the protocol and procedures for implementing the Pan-European Mobile Emergency Application (PEMEA). The first part identifies the key functional entities involved in the emergency application architecture, the interfaces between each functional entity, and the requirements on each interface. The second part defines the data exchanges, message, protocols and procedures used across each of the identified PEMEA interfaces.

It is recognized that many existing application implementations combine the functional entities identified in the present document into a single entity. The most common example of combined functional entities is the combined Application Provider (AP) and PSAP Service Provider (PSP), these are common because it is often the PSAP that writes or engages a third-party to write a local emergency application that interfaces directly with the PSAP. The present document does not seek to disallow integrated node implementations, however, it does not define how additional applications or application providers using proprietary Application Programming Interfaces (APIs) and protocols can provide PEMEA extended features, such solutions are left to the integrated node providers.

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] IETF RFC 2818: "HTTP Over TLS", May 2000.

[2] IETF RFC 2965: "HTTP State Management Mechanism", October 2000.

[3] IETF RFC 4119: "A Presence-based GEOPRIV Location Object Format", December 2005.

[4] IETF RFC 5491: "GEOPRIV Presence Information Data Format Location Object (PIDF-LO) Usage Clarification, Considerations, and Recommendations", March 2009.

[5] IETF RFC 3966: "The tel URI for telephone Number", December 2004.

[6] IETF RFC 7459: "Representation of Uncertainty and Confidence in the Presence Information Data Format Location Object (PIDF-LO)", February 2015.

[7] IETF RFC 3863: "Presence Information Data Format (PIDF)", August 2004.

[8] IETF RFC 5139: "Revised Civic Location Format for Presence Information Data Format Location Object (PIDF-LO)", February 2008.

[9] IETF RFC 6848: "Specifying Civic Address Extensions in the Presence Information Data Format Location Object (PIDF-LO)", January 2013.

[10] IANA: "Method Token Registry of Values".

NOTE: Available at http://www.iana.org/assignments/method-tokens/method-tokens.xhtml#method-tokens-1.

[11] IETF RFC 7852: "Additional Data Related to an Emergency Call", July 2016.

[12] IETF RFC 7105: "Using Device-Provided Location-Related Measurements in Location Configuration Protocols", January 2014.

[13] IANA: "Language subtag registry".

NOTE: Available at http://www.iana.org/assignments/language-subtag-registry/language-subtag-registry.

[14] ISO 639-3 (2007): "Codes for the representation of names of languages -- Part 3: Alpha-3 code for comprehensive coverage of languages".

NOTE: Available at https://www.iso.org/standard/39534.html.

[15] IETF RFC 6753: "A Location Dereference Protocol Using HTTP-Enabled Location Delivery (HELD)", October 2012.

[16] IETF RFC 5808: "Requirements for a Location-by-Reference Mechanism", May 2010.

[17] Open Mobile Alliance OMA-TS-MLP-V3-2-20110719-A: "Mobile Location Protocol 3.2" July 2011.

[18] Open Mobile Alliance OMA-TS-MLP-V3-3-1-20111117-A: "Mobile Location Protocol 3.3.1", November 2011.

[19] Open Mobile Alliance OMA-TS-MLP-V3-4-20150512-A: "Mobile Location Protocol 3.4", May 2015.

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] ETSI TS 103 479: "Emergency Communications (EMTEL); Core elements for network independent access to emergency services".

[i.2] EENA: "Pan-European Mobile Emergency Application (PEMEA) Requirements and Functional Architecture", Version 7, February 2015.

NOTE: Available at https://eena.org/wp-content/uploads/2015_12_02_PEMEA-Final.pdf.

[i.3] Void.

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

security: techniques and methods used to ensure:

- ***authentication*** of entities accessing resources or data
- ***authorization*** of authenticated entities prior to accessing or obtaining resources and/or data
- ***privacy*** of user data ensuring access only to authenticated and authorized entities

- ***secrecy*** of information transferred between two authenticated and authorized entities

trusted: identity of entity assured through an approved authentication mechanism and the entity authorized to perform the action

3.2 Symbols

Void.

3.3 Abbreviations

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

AP	Application Provider
API	Application Programming Interface
App	Application
ASP	Aggregating Service Provider
BCF	Border Control Function
BSSID	Basic Service Set Identifier
CID	Cell Identifier
ECRF	Emergency Call Routing Function
EDR	Emergency Data Received (message)
EDS	Emergency Data Send (message)
EENA	European Emergency Number Association
ESInet	Emergency Services IP Network
ESRP	Emergency Services Routing Proxy
ETSI	European Telecommunications Standards Institute
GML	Geography Markup Language
GNSS	Global Navigation Satellite System
HELD	HTTP-Enabled Location Delivery
HTTP	Hyper-Text Transfer Protocol
IETF	Internet Engineering Task Force
IMEI	International Mobile Equipment Identifier
IMSI	International Mobile Subscriber Identifier
LIF	Location Interworking Function
LIS	Location Information Server
LNG	Legacy Network Gateway
MAC	Media Access Control
MCC	Mobile Country Code
MLP	Mobile Location Protocol
MNC	Mobile Network Code
MSISDN	Mobile Service International Subscriber Dial Number
OMA	Open Mobile Alliance
oPSP	Originating PSP
OTT	Over The Top
Pa	PEMEA Application to AP interface
PEMEA	Pan-European Mobile Emergency Application
PIDF-LO	Presence Information Data Format Location Object
Pp	PEMEA PSP to PSAP interface
Pr	PEMEA PSP to ASP or ASP to PS interface
PRA	PEMEA Registration Authority
Ps	PEMEA AP to PSP interface
PSAP	Public Safety Answering Point
PSP	PSAP Service Provider
PSTN	Public Switched Telephone Network
RTT	Real-Time Text
SIP	Session Initiation Protocol
SIPS	SIP Secure
TLS	Transport Layer Security
tPSP	terminating PSP

ttl	time to live
UCF	Universal Character Set
UMTS	Universal Mobile Telecommunication System (cellular 3G)
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Universal Resource Name
UTC	Coordinated Universal Time
UTF	UCF Transformation Format
VSP	Voice Service Provider
XML	eXtensible Markup Language
XSD	XML Schema Definition

4 PEMEA architecture and functional entities

4.1 Introduction

The extensive deployment of existing mobile emergency applications and their interconnection into a Pan-European Emergency Application ecosystem has prompted the definition of entities identified in Figure 1.

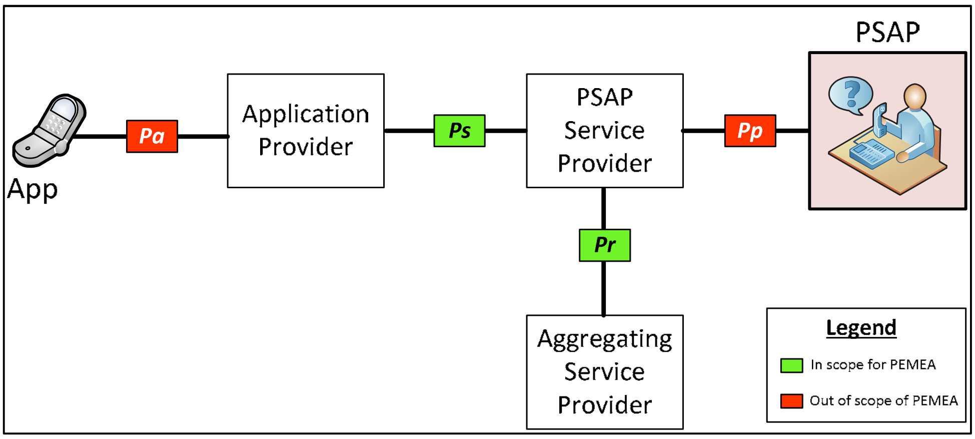


Figure 1: PEMEA Reference Architecture

In some implementations functional entities may be owned and operated by the same commercial entity, for example the Application Provider (AP) and the PSAP Service Provider (PSP) may be the same. In these cases, the external interfaces shown in the reference architecture need only apply when communicating with external entities.

4.2 Functional entities overview

4.2.0 PEMEA Registration Authority

The PEMEA Registration Authority (PRA) is the entity that contains registrations for all currently valid PEMEA entities. The PRA accepts registrations from entities that conform to the PEMEA protocol and procedures and only registered entities may send or receive messages in the PEMEA network. The PRA provides a list of these entities to all valid entities when requested to do so.

4.2.1 Application (App)

Software that runs on a smartphone or mobile computing platform that is capable of making an emergency call using mobile network operator call control machinery (3G/4G/WiFi). Simultaneous to call establishment the App sends user authentication information to an Application Provider and subsequently sends location, connectivity and other information about the caller to the Application Provider for subsequent conveyance to a PSAP.

4.2.2 Application Provider (AP)

The Application Provider (AP) is the entity that provides a mobile emergency application. It is responsible for authenticating the Application prior to accepting caller information from the App. The AP needs to format the data received from the App, possibly combining it with caller information stored in AP server, and conveying it to a PSAP Service Provider (PSP). There needs to be a trust relationship between the AP and PSP. Where the AP and PSP are not the same entity then data formats defined in the present document shall be used to convey caller information from the AP to the PSP.

In the general case, an AP has a relationship with a single PSP. However, an AP may have a relationship with more than one PSP. When this is the case it is up to the AP to determine which PSP to send the information to. How the AP makes this determination is out of scope of the present document, but the AP shall only send the information to one PSP to avoid multiple routing of the same messages through the network.

4.2.3 PSAP Service Provider (PSP)

The role of the PSAP Service Provider (PSP) is to take caller information from trusted sources and ensure that it is provided to the correct PSAP. Where the PSP directly serves the PSAP for which the information is destined, then it is referred to as the terminating-PSP (tPSP). If a PSP receives information that it knows is not for a PSAP that it directly services then it should use its knowledge of other PSPs to attempt to deliver the information to the PSP serving the correct PSAP. When this occurs it is referred to as an originating-PSP (oPSP). This situation occurs when the caller makes the call outside the area that is serviced by the AP that provided the application.

Information coming from trusted sources shall comply with the data formats and communication mechanisms defined in the present document.

Trusted information may come from one of two sources. It may come directly from an AP with which the PSP has a direct trust relationship (***Ps***). Alternatively, the information may come from an AP with which the terminating-PSP has no direct trust relationship (***Pr***). In this latter case, the trust relationship is brokered by another PSP or chain of PSPs to the terminating-PSP.

How the PSP provides or renders information to a PSAP that it directly services is out of scope of the present document.

4.2.4 Aggregating Service Provider (ASP)

The primary role of the PSP is to ensure that accurate and trusted caller information is provided to the PSAP that is terminating an emergency call. A PSP may have knowledge of immediately adjacent terminating-PSPs but requiring a PSP to have a relationship with all other PSPs so that it can direct caller information to the correct terminating-PSP is a daunting and unnecessary task. The role of the Aggregating Service Provider (ASP) is to provide this routing capability and some high-level ideas are described in Annex A.

The ASP operates as a centralized or regional entity and can determine, based on information included in the PEMEA data object, the best terminating-PSP to direct the information to. There may be more than one ASP across Europe and where this occurs meshing is expected to occur. How the meshing occurs is an operational consideration outside the scope of the present document but may be addressed by subsequent operational considerations.

4.3 Interface definitions

4.3.1 Application Interface (Pa)

This is the interface used for communication between the Application and the Application Provider. The exact nature and communication on this interface is out of scope of the present document as this is the interface that allows Application Providers to implement and support service differentiation features in their products. Whilst the implementation of this interface is not in scope of PEMEA, there are specific functions of this interface that a PEMEA-complying implementation shall provide. How these requirements are implemented is out of scope.

4.3.2 Application Provider to PSAP Service Provider Interface (Ps)

This is the interface used by the Application Provider to push caller information to the PSAP Service Provider (PSP). This is a secure interface that requires mutual authentication between the AP and the PSP and a complying AP and PSP shall implement this interface in accordance with the details in the present document when they are not the same entity.

4.3.3 PSAP Service Provider Interface to Aggregating Service Provider Interface (Pr)

This is the interface used by the PSP to route caller information to a different PSP, in which case the sending PSP becomes the origination-PSP (oPSP). The ***Pr*** interface may also be used by the PSP to receive caller information from a different PSP; in this case the receiving PSP becomes the terminating-PSP (tPSP).

This is a secure interface that requires mutual authentication between the oPSP and the tPSP or between the oPSP and the ASP and the tPSP and the ASP. A PSP that wishes to support Application roaming shall implement this interface in accordance with the details in the present document to be PEMEA compliant.

4.3.4 PSAP Service Provider to PSAP Interface (Pp)

This interface is shown for completeness but is outside the scope of the present document. The PSP may provide a simple web interface to the PSAPs it serves or it may integrate the data flows into existing PSAP systems. How this is performed will vary from PSAP to PSAP and from PSP to PSP.

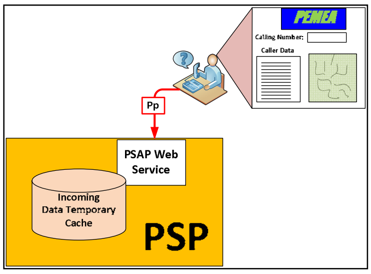


Figure 2: Basic PEMEA PSAP Integration

5 PEMEA functional entity requirements

5.1 Introduction

PEMEA needs to be a secure network and it relies heavily on trust relationships between PSAPs and the entities that they allow to provide information to them. The architecture shown in Figure 1, shows applications connect to application providers (APs) that have trust relationships with PSAP service providers (PSP) that have very strong trust relationships with PSAP. That is, PSAPs trust the PSPs to provide accurate and trustworthy information.

5.2 Application requirements

Even though the Application itself is out-of-scope of the present document, the Application has to fulfil the following requirements to be compliant with PEMEA.

AA-1: The Application shall detect when the Application is being used and initiate an emergency call.

AA-2: The Application shall authenticate itself to the AP when it sends caller information.

AA-3: At emergency call time the Application shall send the most accurate location of the device as obtained from the device's location APIs and a device timestamp.

AA-4: At emergency call time the Application shall send, if it is able to obtain it, the identity of the current point of attachment to the cellular network. At the time of writing this is the full cell-id (MCC-MNC-Cell). However as WiFi becomes more supported as an access technology for cellular operators then the BSSID of the serving WiFi entity may be used instead.

NOTE 1: It is understood that increasingly mobile operating systems are not providing applications access to this information, nevertheless the application should try to acquire it where possible as it may allow for faster routing in some circumstances.

AA-5: The Application shall, if it is able to obtain it, provide the MSISDN of the device to the AP when data is conveyed at call time.

NOTE 2: It is understood that increasingly mobile operating systems are not providing applications access to the MSISDN or IMSI of the device, nevertheless the application should try to acquire this information where possible.

5.3 Application provider requirements

AP-1: The AP shall authenticate the application prior to accepting or processing caller information.

AP-2: The AP shall procure a registered domain name and a domain certificate from a trusted certificate authority asserting ownership of the registered domain name to the AP.

AP-3: The AP shall have a trust relationship with a PSP.

AP-4: The AP shall register with the PEMEA registration authority.

AP-5: The AP shall authenticate and check the authorization of the PSP before sending any data.

AP-6: The AP shall not send any information to a PSAP that fails authentication or authorization.

AP-7: The AP shall authenticate itself to the PSP based on its domain certificate.

AP-8: The AP shall comply with the ***Ps*** interface specification to convey information to a PSP.

AP-9: The AP may provide a means for the destination PSAP to obtain application specific information from the AP.

5.4 PSAP service provider requirements

PSP-1: A PSP shall procure a registered domain name and a domain certificate from a trusted certificate authority asserting ownership of the registered domain name to the PSP.

PSP-2: The PSAP shall register the domain name with the PEMEA registration authority.

PSP-3: A PSP shall identify itself to connecting entities using its domain certificate.

PSP-4: A PSP shall authenticate and check authorization of the AP each time a connection is made.

PSP-5: A PSP shall never accept connections from an AP that fails authentication or authorization.

PSP-6: An oPSP shall not cache caller information if the information is pushed to a tPSP or to an ASP over the ***Pr*** interface.

PSP-7: A tPSP shall not cache or log caller information for longer than terminating PSAP statutes allow, and should adhere to advertised caching periods provided in any messages or data structures.

PSP-8: If a PSP is unable to determine where the caller information should be delivered then it shall return an error to the node providing it with the information.

PSP-9: An oPSP shall authenticate and authorize any ASP or tPSP before sending any information over the ***Pr*** interface.

PSP-10: A tPSP shall authenticate and authorize an oPSP or ASP each time is connects.

PSP-11: A tPSP shall never accept connections from an oPSP or an ASP that fails to authenticate.

PSP-12: A tPSP shall never process message data from an oPSP or an ASP that fails authorization.

PSP-13: A tPSP may request additional information from the originating AP providing authentication and authorization is confirmed.

5.5 Aggregating service provider requirements

ASP-1: The ASP shall procure a registered domain name and a domain certificate from a trusted certificate authority asserting ownership of the registered domain name to the ASP.

ASP-2: The ASP shall register the domain name with the PEMEA registration authority.

ASP-3: The ASP shall authenticate itself to any PSP or ASP using its registered domain certificate each time it connects.

ASP-4: The ASP shall authenticate each incoming connection and shall terminate any connection that fails authentication.

ASP-5: The ASP shall never accept data from an incoming entity that fails authorization.

ASP-6: The ASP shall never pass data to an entity that fails authorization.

ASP-7: The ASP shall return an error to the oPSP if it is unable to determine where to send the call information, or if the maximum number of hops for the message is exceeded.

ASP-8: The ASP shall never cache or log caller information.

6 PEMEA Message Element Definitions

6.1 Introduction

To keep PEMEA simple there are only three required message types:

1) emergencyDataSend, used to send information from the AP ultimately to the PSP and PSAP

2) emergencyDataReceived, used by the node receiving the information that they got it and who they are sending it on to

3) error, used to indicate that something went wrong

The following clauses define the information elements required for each of these messages.

6.2 emergencyDataSend Information Elements

Table 1: emergencyDataSend message element definitions

Element	Inclusion	Description
Time To Live	Mandatory	Defines the number of hops allowed before message delivery stops.
Receive Error Post	Recommended	When an entity receives an error message form the next hop and this element is provided, the entity receiving the error should pass the error to the address contained in this element. This capability is used to inform the AP if the data reached the destination PSAP or not.
Receive Capabilities Support	Conditional	This element shall be provided if the AP capabilities element is provided. The terminating entity sends a message to the address provided in this element containing all the capabilities contained in the AP capabilities element that it supports. This message may be empty if no capabilities are supported.
Route	Mandatory	Defines the nodes and the order through which the message has passed the nodes. This will include the first originating node sending this message.
Caller Identities	Mandatory	The mobile number of the caller. In Europe it is common for phones to support more than one SIM card, consequently this element may appear multiple times but shall appear at least once. If the App cannot obtain this from the device-API at call time, then it shall be configured when the application is installed or registered with the AP. See note.
Location information	Mandatory	The location as determined by the device.
Access Data	Conditional	The current serving mobile base station identifier or WiFi BSSID if WiFi connectivity is being used. If one of these values is available through the device-API then it is provided, where the device-API does not support obtaining any of these values then this field may be omitted.
Application Provider Information	Mandatory	Details on how to contact the Application Provider, including phone, helpdesk and email contacts.
Caller Information	Mandatory	Contains information about the caller.
AP capabilities	Optional	Anything additional that the AP wishes to provide to the PSAP, such as a URI for obtaining additional information or future PEMEA extensions.
NOTE: SIMless devices are not considered in the present document but may be an area of further study.		

The purpose of the Route element is to describe the path that a particular set of data took through the PEMEA network. It captures each node that message passes through and the time at which it passed through that node. In this way it serves two purposes, it avoids circular routing through the network and it provides a means to determine where a routing error may have occurred.

Table 2: Route element definition

Element	Inclusion	Description
Sequence Number	Mandatory	A unique sequence number generated by the AP when it creates the emergencyDataSend message.
Hop	Mandatory	This is a complex element that defines each node (hop) in the PEMEA signalling. There shall be at least one hop, but there may be as many as are required to get the message to the destination PSAP or until the ttl reaches zero, which ever happens first.

The Hop element defines the entity through which the PEMEA message passed, the time that the entity passed, the link number in the chain of nodes, and decrement the ttl.

Table 3: Hop element definition

Element	Inclusion	Description
Time Stamp	Mandatory	The time that this hop was added to the route element.
Position	Mandatory	The number of nodes through which the message has passed prior to this node. The AP hop will have a Position value of zero.
Node	Mandatory	The URI of the node to which this hop is attributed.

6.3 emergencyDataReceived Information Elements

Table 4: PEMEA emergencyDataReceived information element definitions

Element	Inclusion	Description
Time Stamp	Mandatory	The time that this message was sent by the receiving node to the originating hop immediately preceding it.
Route	Mandatory	Defines the nodes and the order through which the message has passed the nodes. This will include the node sending this message.
Delivery	Mandatory	The node to which the emergencyDataSend information has passed on to.

6.4 error Information Elements

Table 5: PEMEA error information element definitions

Element	Inclusion	Description
Time Stamp	Mandatory	The time that this message was sent by the receiving node to the originating hop immediately preceding it.
Reason	Mandatory	A token representing the kind of problem that occurred.
Route	Mandatory	Defines the nodes and the order through which the message has passed the nodes. This will include the node sending this message.
Message	Optional	A text message providing more information to the user as to what occurred.

6.5 PEMEA Confirmation Messages

The EDS, EDR and error messages described in the previous clauses are all hop-by-hop messages exchanged between adjacent nodes in order to get the caller information to the correct PSAP. PEMEA provides for extensions to enable the PSAP to request further information from the caller via the AP by including a capabilities element in the EDS message.

The additional capabilities are invoked by the PSAP as supported or required and may include periodic polling for information, such as requesting an updated location, or it may enable streaming media, such as video to further augment the information available to the PSAP call-taker.

Support for these capabilities consumes resources in the AP, so it is useful to know which, if any, of the additional capabilities the receiving PSAP can use. It is also good to know if the EDS message actually delivered to the PSAP.

The error case is covered by the receive error post element in the EDS described in clause 6.1.

The PSAP supported capabilities functionality is supported by a capabilities message sent from the PSAP to the AP.

Table 6: PSAP Capability Support Message

Element	Inclusion	Description
AP Capabilities	Mandatory	This is the list of capabilities provided by the AP that the PSAP supports. The list may be empty if the PSAP does not support any of the proffered capabilities. This message is posted to the URI provided in the Received Capabilities Support element of the EDS.

7 PEMEA Message Flows

7.1 Introduction

The *Pa* and *Pp* message flows are out of scope of the present document, so this clause only describes the *Ps* and *Pr* message flows. Establishment of the connections is assumed to occur prior to message exchanges occurring. Connection establishment procedures are covered in the detailed protocol clause.

7.2 Ps message flows

7.2.1 Ps message flow description

The *Ps* interface is the interface between the AP and PSP. The connection originates from the AP making the AP the client and the PSP the server. The AP puts its data into an emergencyDataSend message (the precise format for this message is defined in a detailed protocol clause) and uses HTTP to deliver the message to the PSP.

The AP may not know if the user is roaming or if the PSP to which the AP has a direct association servicing the PSAP that requires the data. The AP, by setting parameters in the emergencyDataSend can control how the message is routed based on AP or user policy. How this policy is set in the AP or by the user is outside the scope of the present document.

7.2.2 Ps basic flow

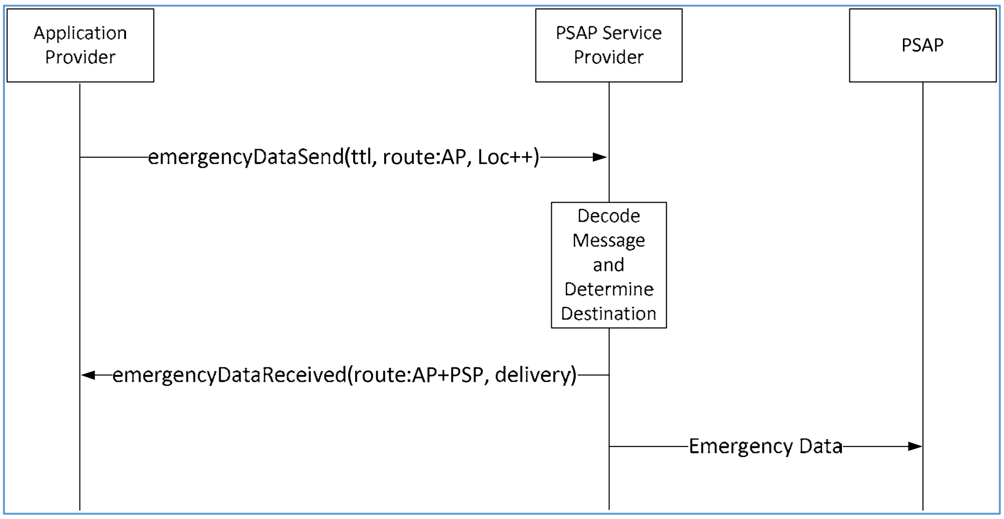

Application Provider
PSAP Service Provider
PSAP
emergencyDataSend(ttl, route:AP, Loc++)
Decode Message and Determine Destination
emergencyDataReceived(route:AP+PSP, delivery)
Emergency Data

Figure 3: Ps basic message flow

In this flow the AP:

1) creates an emergencyDataSend message:
   - sets the time to live (ttl) to the number of hops the AP will allow before the message is dropped. The minimum value that the AP may set the ttl value to is 1
   - adds its identity information to the route element
   - assembles and appends the location and other information
   - sends the message to the PSP

The PSP:

1) receives the message from the AP and decodes it
2) examines the location and determines it is for the local PSAP
3) creates an emergencyDataReceived message
4) copies the route element from the emergencyDataSend message, adds its identity information to this element, then inserts this element in the emergencyDataReceived message
5) adds the PSAP identity information into the delivery element of the emergencyDataReceived message
6) logs the data from the route and delivery elements
7) sends the message to the AP and closes the connection

8) sends the data to the PSAP

7.2.3 Ps error flow

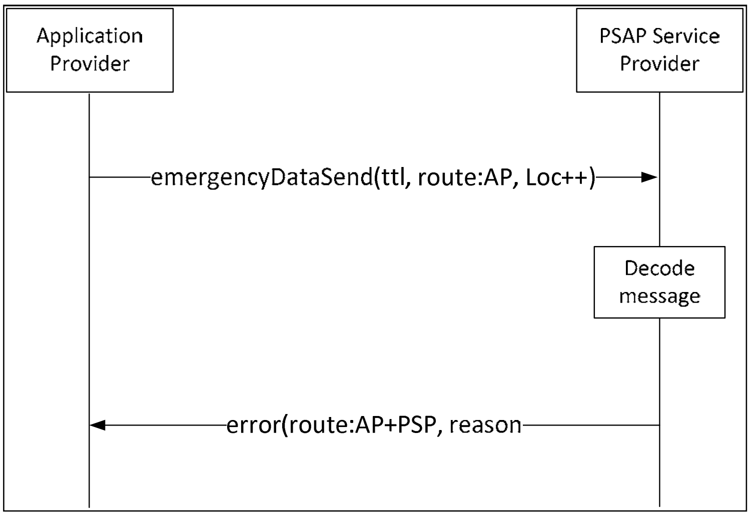

Application Provider
PSAP Service Provider
emergencyDataSend(ttl, route:AP, Loc++)
Decode message
error(route:AP+PSP, reason

Figure 4: Ps error flow

In this flow the AP:

1) creates an emergencyDataSend message:
    - sets the time to live (ttl) to the number of hops the AP will allow before the message is dropped
    - adds its identity information to the route element
    - assembles and appends the location and other information
    - sends the message to the PSP

The PSP:

1) receives the message from the AP and:
    a) fails to decode it or
    b) cannot determine a next hop or
    c) the ttl does not permit retransmission of the message
    d) determines that the next hop is already present in the route element
2) creates an emergencyDataError message
3) copies the route element from the emergencyDataSend message, adds its identity information to this element, then inserts this element in the emergencyDataReceived message
4) adds the reason for the error into the reason element of the emergencyDataError message
5) logs the data from the route and reason elements
6) sends the message to the AP and closes the connection

7.2.4 Ps routing flow

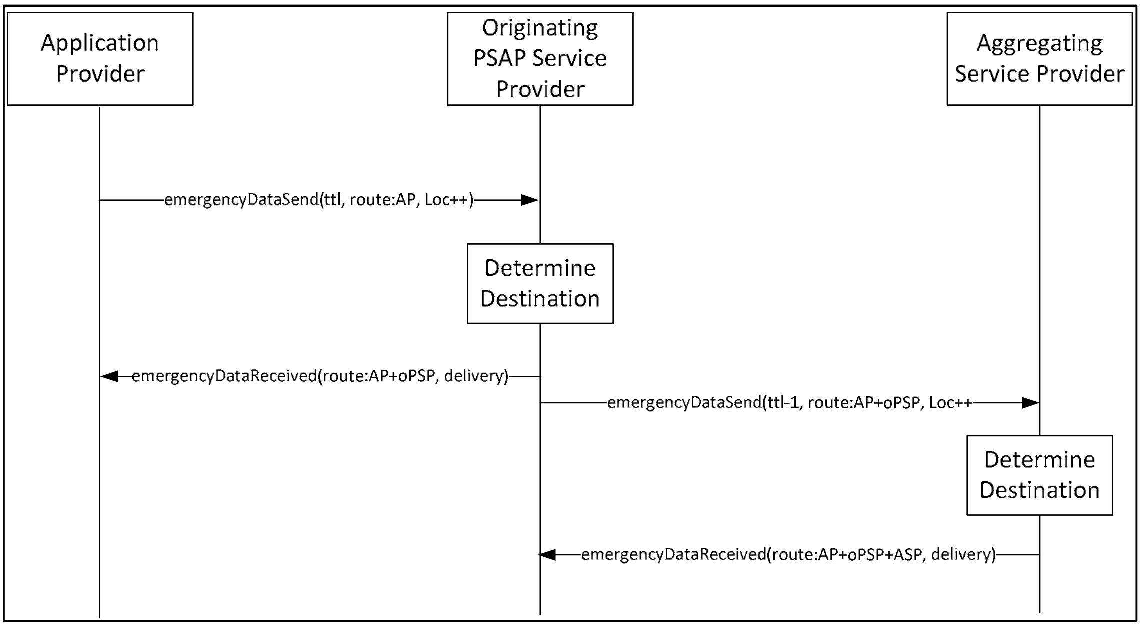

Application Provider
Originating PSAP Service Provider
Aggregating Service Provider
emergencyDataSend(ttl, route:AP, Loc++)
Determine Destination
emergencyDataReceived(route:AP+oPSP, delivery)
emergencyDataSend(ttl-1, route:AP+oPSP, Loc++
Determine Destination
emergencyDataReceived(route:AP+oPSP+ASP, delivery)

Figure 5: Ps routing flow

In this flow the AP:

1) creates an emergencyDataSend message:
   - sets the time to live (ttl) to the number of hops the AP will allow before the message is dropped
   - adds its identity information to the route element
   - assembles and appends the location and other information
   - sends the message to the PSP

The oPSP:

1) receives the message from the AP and decodes it
2) examines the location and determines the next hop for the message to take in order to reach the correct PSAP is an ASP
3) verifies that the next hop is not already present in the route element
4) creates an emergencyDataReceived message
5) copies the route element from the emergencyDataSend message, adds its identity information to this element, then inserts this element in the emergencyDataReceived message
6) adds the ASP identity information into the delivery element of the emergencyDataReceived message
7) logs the data from the route and delivery elements
8) sends the message to the AP and closes the connection
9) decrements the ttl value in the emergencyDataSend message
10) adds its identity information to the route element of the emergencyDataSend message
11) opens a secure connection to the ASP
12) sends the emergencyDataSend message to the ASP

The ASP:

1) receives the message from the oPSP and decodes it
2) examines the location and determines the next hop for the message to take to reach the correct PSAP
3) verifies that the next hop is not already present in the route element
4) creates an emergencyDataReceived message
5) copies the route element from the emergencyDataSend message, adds its identity information to this element, then inserts this element in the emergencyDataReceived message
6) adds the next hop identity information into the delivery element of the emergencyDataReceived message
7) logs the data from the route and delivery elements
8) sends the message to the oPSP and closes the connection

7.3 Pr message flows

7.3.1 Pr message flow description

The *Pr* interface is the interface between the PSP and the ASP. The connection may be originated by either the PSP or the ASP depending on whether the PSP is forwarding the data or terminating the data.

The sender (either the oPSP or the ASP) puts its data into an emergencyDataSend message (the format for this message is defined in a subsequent clause) and uses HTTP to deliver the message to the receiver (either the PSP or the ASP).

7.3.2 Pr terminating-PSP basic flow

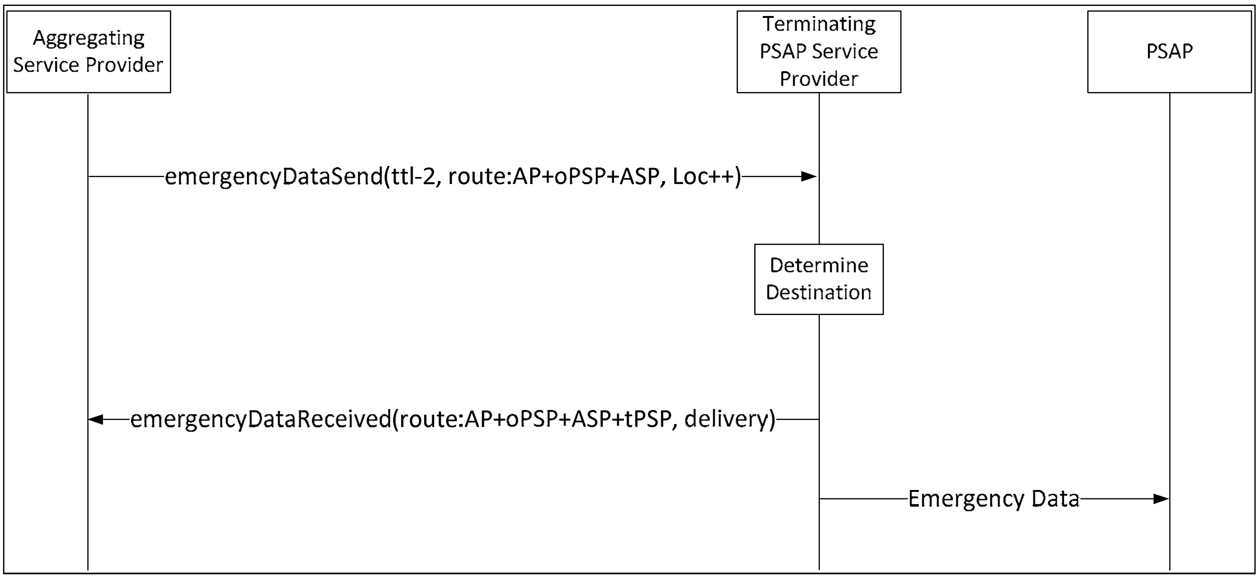

Aggregating Service Provider
Terminating PSAP Service Provider
PSAP
emergencyDataSend(ttl-2, route:AP+oPSP+ASP, Loc++)
Determine Destination
emergencyDataReceived(route:AP+oPSP+ASP+tPSP, delivery)
Emergency Data

Figure 6: Pr basic message flow

In this flow, the AP has sent an emergencyDataSend message to a PSP but that PSP did not serve the destination PSAP so it sent the message on to an ASP. This flow illustrates the ASP directing the emergencyDataSend message to a terminating PSP.

In this flow the ASP:

1) receives an emergencyDataSend message from an oPSP and determines the next hop for the message
2) checks the contents of the route element to ensure that the next hop has not already been visited

3) if the route element is clear it responds to the oPSP with an emergencyDataReceived message and closes the connect

4) copies the received emergencyDataSend message

5) decrements the time to live (ttl) value from the received message

6) adds its identity information to the route element

7) sends the message to the tPSP

The tPSP:

1) receives the message from the ASP and decodes it

2) examines the location and determines it is for the local PSAP

3) creates an emergencyDataReceived message

4) copies the route element from the emergencyDataSend message, adds its identity information to this element, then inserts this element in the emergencyDataReceived message

5) adds the PSAP identity information into the delivery element of the emergencyDataReceived message

6) logs the data from the route and delivery elements

7) sends the message to the ASP and closes the connection

8) sends the data to the PSAP

7.3.3 Pr error flow

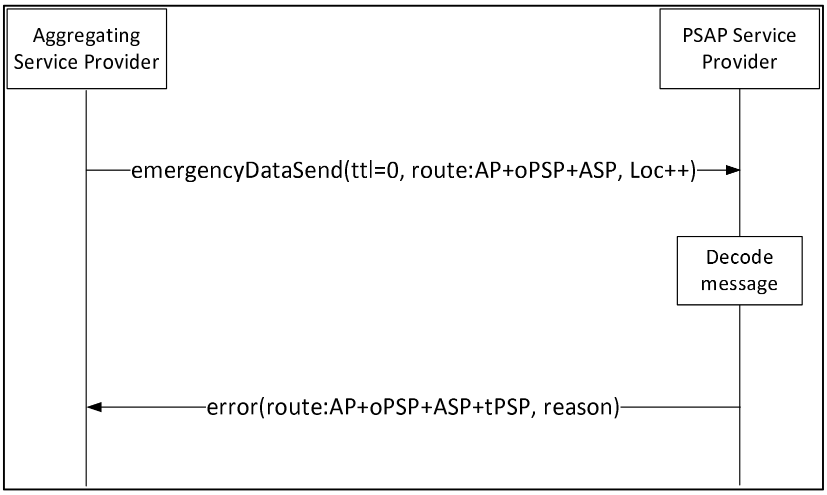

Aggregating Service Provider
PSAP Service Provider
emergencyDataSend(ttl=0, route:AP+oPSP+ASP, Loc++)
Decode message
error(route:AP+oPSP+ASP+tPSP, reason)

Figure 7: Pr error flow

In this flow the ASP:

1) receives the emergencyDataSend message from the oPSP

2) determines the correct tPSP and sends the corresponding response to the oPSP before closing the connection

3) decrements the time to live (ttl) in the received emergencyDataSend message

4) adds its identity information to the route element

5) sends the message to the terminating PSP and closes the connection

The tPSP:

1) receives the message from the ASP and:

   a) fails to decode it or

   b) cannot determine a next hop or

   c) the ttl does not permit retransmission of the message, as shown in Figure 7 (except directly to the PSAP) or

   d) the next hop is already present in the route element

2) creates an emergencyDataError message

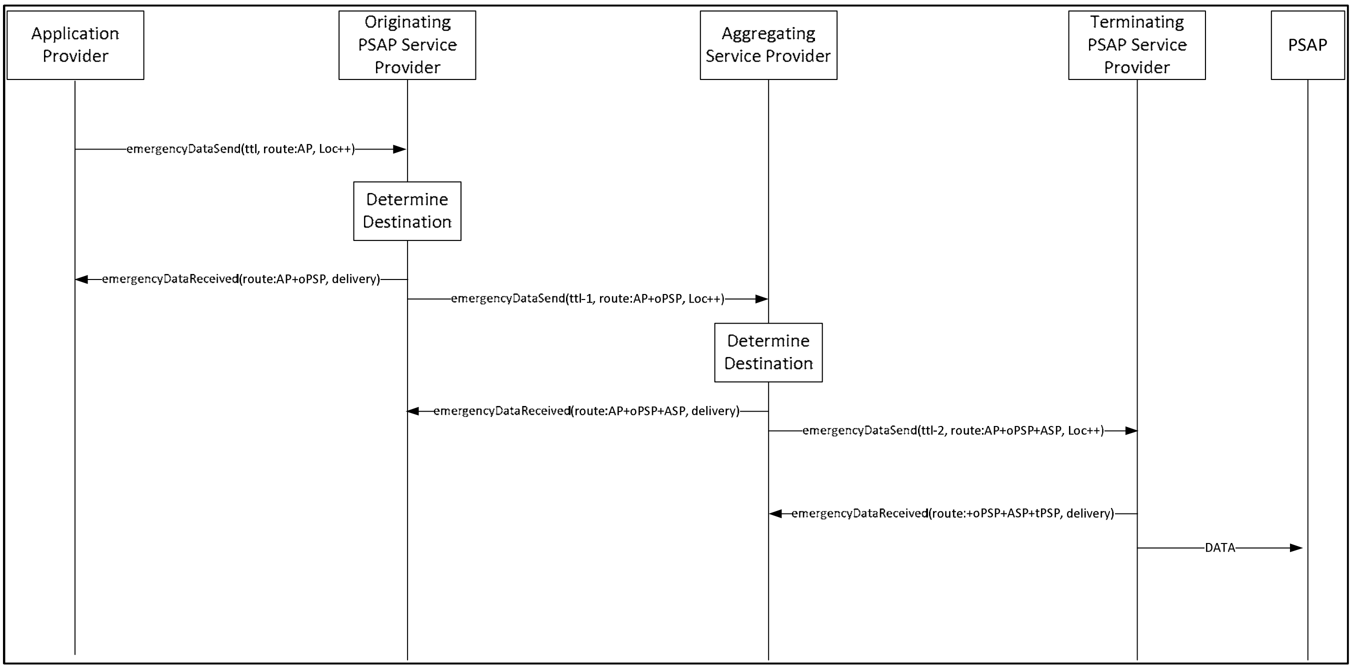
3) copies the route element from the emergencyDataSend message, adds its identity information to this element, then inserts this element in the emergencyDataReceived message

4) adds the reason for the error into the reason element of the emergencyDataError message

5) logs the data from the route and reason elements

6) sends the message to the ASP and closes the connection

7.3.4 Pr end to end routing flow



Application Provider
Originating PSAP Service Provider
Aggregating Service Provider
Terminating PSAP Service Provider
PSAP
emergencyDataSend(ttl, route:AP, Loc++)
Determine Destination
emergencyDataReceived(route:AP+oPSP, delivery)
emergencyDataSend(ttl-1, route:AP+oPSP, Loc++)
Determine Destination
emergencyDataReceived(route:AP+oPSP+ASP, delivery)
emergencyDataSend(ttl-2, route:AP+oPSP+ASP, Loc++)
emergencyDataReceived(route:+oPSP+ASP+tPSP, delivery)
DATA

Figure 8: Ps routing flow

In this flow the AP:

1) creates an emergencyDataSend message

2) sets the time to live (ttl) to the number of hops the AP will allow before the message is dropped

3) adds its identity information to the route element

4) assembles and appends the location and other data

5) sends the message to the oPSP

The oPSP:

1) receives the message from the AP and decodes it
2) examines the location and determines the next hop for the message to take in order to reach the correct PSAP is an ASP
3) verifies that the next hop is not already present in the route element
4) creates an emergencyDataReceived message
5) copies the route element from the emergencyDataSend message, adds its identity information to this element, then inserts this element in the emergencyDataReceived message
6) adds the ASP identity information into the delivery element of the emergencyDataReceived message
7) logs the data from the route and delivery elements
8) sends the message to the AP and closes the connection
9) decrements the ttl value in the emergencyDataSend message
10) adds its identity information to the route element of the emergencyDataSend message
11) opens a secure connection to the ASP
12) sends the emergencyDataSend message to the ASP

The ASP:

1) receives the message from the oPSP and decodes it
2) examines the location and determines the next hop for the message to take to reach the correct PSAP
3) verifies that the next hop is not already present in the route element
4) creates an emergencyDataReceived message
5) copies the route element from the emergencyDataSend message, adds its identity information to this element, then inserts this element in the emergencyDataReceived message
6) adds the next hop identity information into the delivery element of the emergencyDataReceived message
7) logs the data from the route and delivery elements
8) sends the emergencyDataReceived message to the tPSP and closes the connection
9) decrements the ttl value in the emergencyDataSend message
10) adds its identity information to the route element of the emergencyDataSend message
11) opens a secure connect to the tPSP
12) sends the emergencyDataSend message to tPSP

tPSP (Termination PSP):

1) receives the message from the ASP and decodes it
2) examines the location and determines it is for the local PSAP
3) creates an emergencyDataReceived message
4) copies the route element from the emergencyDataSend message, adds its identity information to this element, then inserts this element in the emergencyDataReceived message
5) adds the PSAP identity information into the delivery element of the emergencyDataReceived message
6) logs the data from the route and delivery elements

7) sends the message to the ASP and closes the connection

8) sends the data to the PSAP

8 PEMEA alignment with ETSI TS 103 479

8.1 General alignment

PEMEA was intended to provide a stepping stone to the Core element for network independent access to emergency services ETSI TS 103 479 [i.1], and it does this by reusing many of the same data structure used in the IETF and 3GPP documents as well as ETSI TS 103 479 [i.1]. The core elements document [i.1] however, only defines what happens once the emergency call reaches the gateway to the Emergency Services Intranet (ESInet), it does not define how the call gets to the gateway in the first place. PEMEA can align with ETSI TS 103 479 [i.1] by addressing this issue and allowing calls to provide much of the ancillary data provided through PEMEA. Clauses 8.2 and 8.3 provide descriptions for a call that is SIP-based and traverses the Border Control Function (BCF) to enter the ESInet, and the legacy case where the call enters the ESInet via a Legacy Network Gateway (LNG). The information contained in these clauses is informative only.

8.2 PEMEA to border control function

In this case the device is using SIP to make an emergency call. In order for the call to reach the correct ESInet it needs to first reach the BCF at the gateway to the ESInet. In an Over The Top (OTT) environment there is no direct association between the access network and the call service provider, so routing information is required to get the SIP INVITE from the calling device to the BCF. In the absence of a hierarchy of publically accessible Emergency Call Routing Functions (ECRFs), this problem becomes hard. The PEMEA routing function by contrast is distributed in nature, using hop-by-hop routing to reach the final destination. Adding an option to the PEMEA Emergency Data Send (EDS) message for the calling device to indicate that it would like the termination-PSP to provide a SIP URI enables the calling device to use the PEMEA network to provide routing information to the terminating-PSP.

This requires the terminating-PSP to be associated with the BCF at the gateway to the ESInet. The AP needs to indicate to the PSAP/PSP that it requires a SIP URI for the App to send to the SIP INVITE, the means to do this is provided in Table 10.

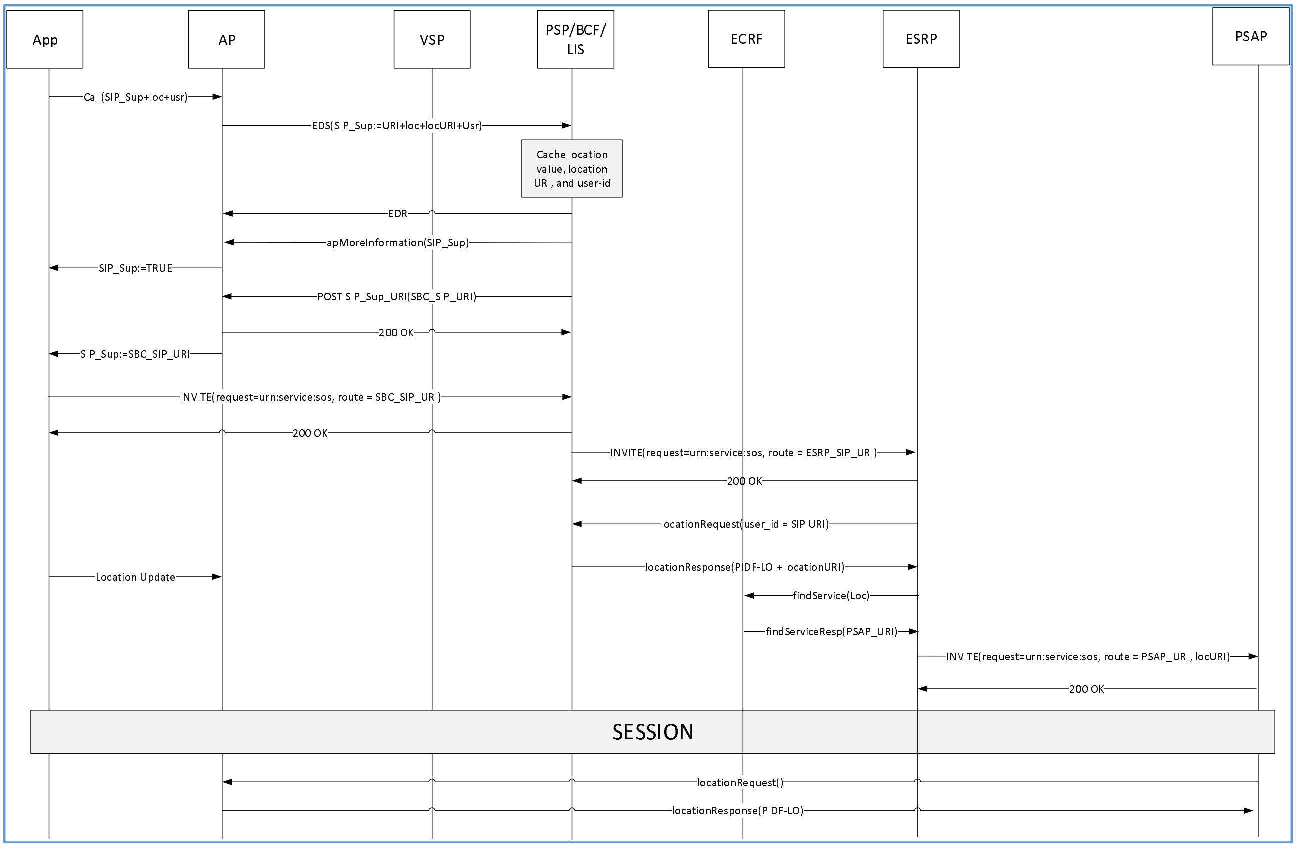

App
AP
VSP
PSP/BCF/ LIS
ECRF
ESRP
PSAP
Call(SIP_Sup+loc+usr)
EDS(SIP_Sup:=URI+loc+locURI+Usr)
Cache location value, location URI, and user-id
EDR
apMoreInformation(SIP_Sup)
SIP_Sup:=TRUE
POST SIP_Sup_URI(SBC_SIP_URI)
200 OK
SIP_Sup:=SBC_SIP_URI
INVITE(request=urn:service:sos, route = SBC_SIP_URI)
200 OK
INVITE(request=urn:service:sos, route = ESRP_SIP_URI)
200 OK
locationRequest(user_id = SIP URI)
locationResponse(PIDF-LO + locationURI)
Location Update
findService(Loc)
findServiceResp(PSAP_URI)
INVITE(request=urn:service:sos, route = PSAP_URI, locURI)
200 OK
SESSION
locationRequest()
locationResponse(PIDF-LO)

Figure 9: PEMEA alignment with ETSI TS 103 479 [i.1] for SIP-originated calls

1) The Application indicates to the AP that the user is making an emergency call and provides the usual location and user data information, as well as a request for a SIP URI and indicates that it supports location updates.

2) The AP bundles this information into an EDS and sends it to its local PSP and the EDS is then routed through the PEMEA network to the terminating PSP which is coupled with the BCF and ESINet Location Information Server (LIS). An EDR is sent for each PEMEA hop to indicate successful receipt of the EDS.

3) The PSP caches the user identity, user information reference, and location URI.

4) PSP indicates to the AP that it supports location updates and it can provide a SIP URI.

5) The AP indicates to the Application that a SIP URI is coming.

6) The PSP then posts to the AP the SIP URI of the BCF. That is, the gateway to the correct ESINet.

7) The AP sends the BCF SIP URI to the Application.

8) The Application initiates a SIP call via its VSP to the BCF.

9) The BCF terminates the call and sends the Application a 200 OK.

10) The BCF sends the SIP INVITE to the emergency services routing proxy (ESRP).

11) The ESRP needs location information before it can determine the final destination PSAP, it gets this information by making a HELD location request to the internal LIS that is coupled with the terminating PSP. The location request includes the SIP URI in the FROM header field of the SIP INVITE.

12) The LIS/PSP responds with the PIDF-LO and the locationURI from the EDS.

13) The ESRP takes the location value and queries the ECRF for the destination PSAP URI.

14) The ECRF uses the location to determine the correct PSAP and returns this to the ESRP.

15) The ESRP is also able to apply policy routing to the returned URI, such as language support. It then puts the locationURI in the Geolocation header field of the INVITE message, addresses it to the selected PSAP and sends the INVITE to the PSAP.

16) The PSAP acknowledges receipt of the INVITE and then proceeds to establish a media session with the Application (this will always go through the BCF).

17) In the meantime the Application is able to send location updates to the AP.

18) The PSAP is able to use the locationURI provided in the SIP signalling to request location updates from the AP.

8.3 PEMEA to Legacy Network Gateway

In this scenario, the call is originated through the Public Switch Telephone Network (PSTN) and needs to go through an LNG to enter the ESInet. The LNG requires a Location Interworking Function (LIF) in order to provide the location of the caller for further routing through the ESInet to the correct PSAP. The terminating-PSP is coupled with the LNG/LIF, so when the call arrives at the LNG, the information contained in the EDS is associated with the call allowing the call to be routed to the most appropriate PSAP. Further to this a location URI can be provided allowing the PSAP to get location updates.

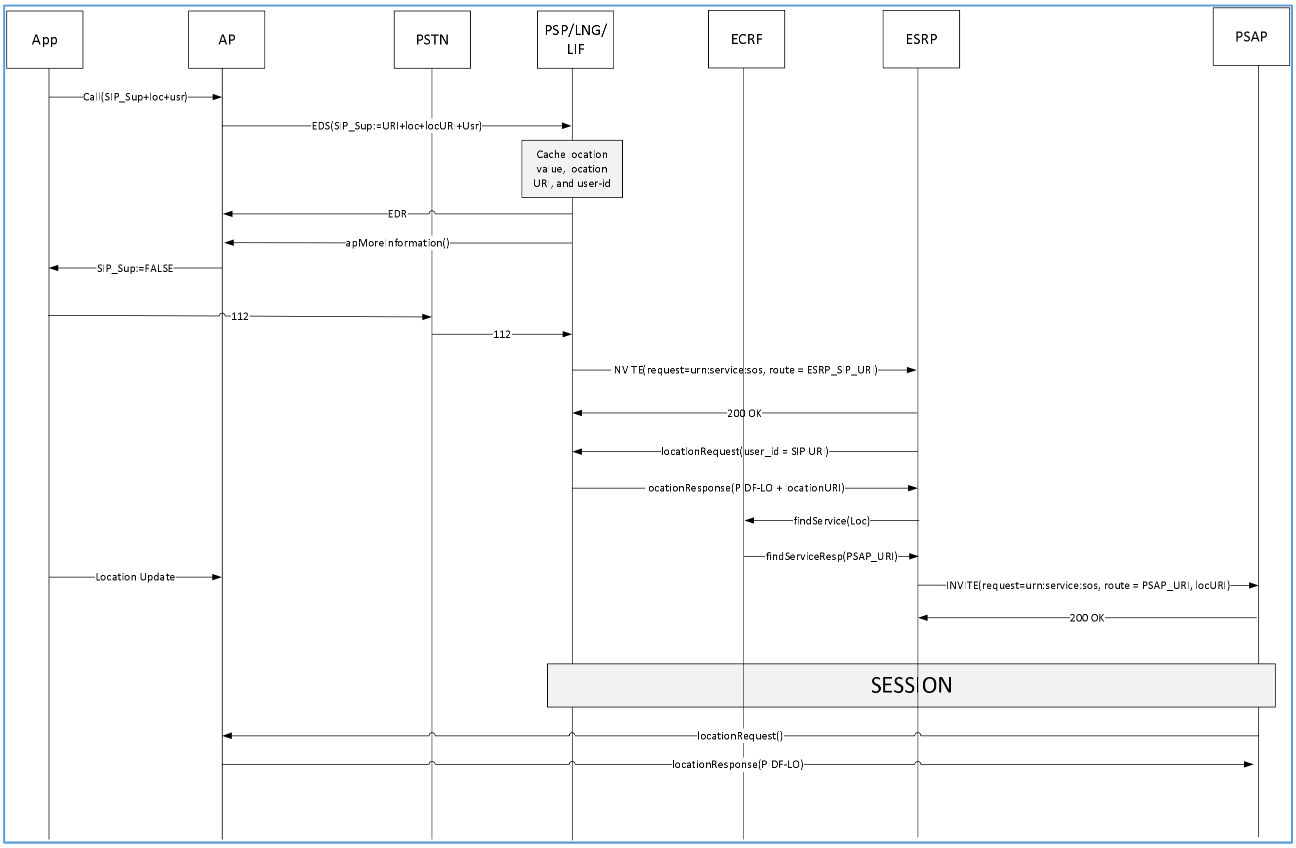

App
AP
PSTN
PSP/LNG/ LIF
ECRF
ESRP
PSAP
Call(SIP_Sup+loc+usr)
EDS(SIP_Sup:=URI+loc+locURI+Usr)
Cache location value, location URI, and user-id
EDR
apMoreInformation()
SIP_Sup:=FALSE
112
112
INVITE(request=urn:service:sos, route = ESRP_SIP_URI)
200 OK
locationRequest(user_id = SIP URI)
locationResponse(PIDF-LO + locationURI)
findService(Loc)
findServiceResp(PSAP_URI)
Location Update
INVITE(request=urn:service:sos, route = PSAP_URI, locURI)
200 OK
SESSION
locationRequest()
locationResponse(PIDF-LO)

Figure 10: PEMEA alignment with ETSI TS 103 479 [i.1] for legacy calls

1) The Application indicates to the AP that the user is making an emergency call and provides the usual location and user data information, as well as a request for a SIP URI and indicates that it supports location updates.

2) The AP bundles this information into an EDS and sends it to its local PSP and the EDS is then routed through the PEMEA network to the terminating PSP which is coupled with the LNG and LIF. An EDR is sent for each PEMEA hop to indicate successful receipt of the EDS.

3) The PSP caches the user identity, user information reference, and location URI.

4) PSP indicates to the AP that it supports location updates but that it cannot provide a SIP URI.

5) AP tells the Application that it cannot use SIP to make the call because there is no SIP PSAP or BCF address available.

6) The Application dials 112 which gets routed via the Public Switch Telephone Network (PSTN) to a legacy network gateway.

7) The LNG turns the call from circuit-based to SIP and sends it to the Emergency Service Routing Proxy (ESRP).

8) The ESRP needs location information before it can determine the final destination PSAP, it gets this information by making a HELD location request to the internal LIF that is coupled with the terminating PSP. The location request includes the SIP URI in the FROM header field of the SIP INVITE, which will be the telephone number of the caller.

9) The LIF/PSP responds with the PIDF-LO and the locationURI from the EDS.

10) The ESRP takes the location value and queries the ECRF for the destination PSAP URI.

11) The ECRF uses the location to determine the correct PSAP and returns this to the ESRP.

12) The ESRP is also able to apply policy routing to the returned URI, such as language support. It then puts the locationURI in the Geolocation header field of the INVITE message, addresses it to the selected PSAP and sends the INVITE to the PSAP.

13) The PSAP acknowledges receipt of the INVITE and then proceeds to establish a media session with the LNG, that is subsequently turned in a PSTN circuit so that the PSAP call taker can talk with the user.

14) In the meantime, the Application is able to send location updates to the AP.

15) The PSAP is able to use the locationURI provided in the SIP signalling to request location updates from the AP.

16) The PSAP also has access to a lot of user information that would otherwise not be available in a call of this type.

9 Message transportation and processing

9.1 HTTP usage

All messages are conveyed as XML over secure HTTP.

The HTTP connection is established using mutual authentication in accordance with IETF RFC 2818 [1].

Cookies as defined in IETF RFC 2965 [2], shall not be used.

Providers shall use the HTTP POST method to deliver emergencyDataSend messages in the body of the HTTP message. The sending entity shall include a Host header when initiating a connection. No sending entity shall accept a redirect response and shall cease to attempt message delivery in the event that one is received.

emergencyDataReceived and application error messages are returned using HTTP 200 responses.

The HTTP connection shall be closed by the server once the response is sent.

Table 7: HTTP Request Headers for PEMEA

Header	Value	Description
Request method	POST	Defines the HTTP method being used when passing information to the server
Host	psp.example.com:47273	The address of the host being sort.
Connection	close	Each originating message shall establish a new connection to the server
Accept	application/xml	
Content-Type	application/xml; charset=utf-8	The content type of the message
Cache-control	private	
Content-Length		

Table 8: HTTP Response Headers for PEMEA

Header	Value	Description
Response	HTTP/1.1 200 OK	
Server	Example PSP	
Date	Tue, 02 Feb 2016 20:45:30 GMT	
Expires	Tue, 02 Feb 2016 20:45:30 GMT	
Cache-control	Private	
Content-Type	application/xml; charset=utf-8	The content type of the message
Content-Length		

9.2 Authenticating and authorizing PEMEA entities

All entities, APs, PSPs, PSAPs and ASPs need to have a PEMEA identifier (PEMEA-ID) and this is provided by the PEMEA registry when an application is approved. The PEMEA registry provides a list of each entity, the entity type and entity's registered domain name. All PEMEA entities should load this information into their servers periodically. The recommended refresh period is outlined in the operational document.

Each entity is required to obtain and maintain a domain name certificate stemming from a well-known root CA and this shall be the domain name provided to the PEMEA registry at the time of registration.

The client entity shall assert itself to the serving entity when attempting to access a resource. This is done by using the client entity's domain certificate. The serving entity similarly asserts its identity to the requesting entity. The serving entity validates the requesting entity's certificate against and matches the domain name to a registered PEMEA-entity, if it does not match, then the request for data shall be denied and an HTTP 403 "Forbidden" response sent to the requesting entity.

Resources accessed by certain URI types, such as the reach-back URIs defined in clause 10.3.12 and the onCapSupportPost URI described in clause 11.1.4 shall only be accessible to a PSAP or PSP.

9.3 PEMEA Securing a PSAP Retrieving Data By Reference or a reach-back URI

Clause 6.4 describes a means by which an AP may advertise a means for a PSAP to acquire additional information related to a call or caller from the AP. In the present document, this mechanism is implemented using URIs.

Information about the AP, and caller information is conveyed in IETF Additional-Data structures defined in IETF RFC 7105 [12], with a general preference to information being conveyed by value. However, it may not be legal in some jurisdictions to send private caller information (contained in the SubscriberData structure) to any entity but the receiving PSAP or PSP. In this case SubscriberData shall be sent by reference and only provided to a validated PSAP or PSP querying for the information.

The AP advertises its capabilities to the destination PSAP through reach-back URIs specified in the apMoreInformation information elements. The AP shall not accept any requests to invoke information element URIs until after it has received an onCapSupportPost message from the terminating PSP or PSAP. Furthermore, only the capabilities responded to in the onCapSupportPost message shall be supported by the AP from that point on for the duration of the call, and the AP shall reject URI invocation requests from any node except for the node that sent the onCapSupportPost message to it.

9.4 PEMEA XML Processing Rules

The PEMEA messages are specified as XML elements and are designed with explicit extension points. These extension points exist for two main reasons.

Firstly, PEMEA needs to be capable of transferring intra-country messages between providers and agencies, and some countries and agencies have local requirements over and above those addressed in the basic PEMEA data set. These extension points allow this information to be included by an AP or a PSP in all PEMEA exchanges thereby simplifying implementation.

Secondly, it is envisaged that PEMEA will require general enhancements and extensions over its lifetime. However, like all wide-network deployments, it is impossible to upgrade all nodes at once, or across national boundaries to enforce upgrades. As a consequence the schema extension points designed into the protocol elements provide a means for this evolution of PEMEA functionality.

The standard processing rules for XML are that if an element is not understood or recognized then it is ignored. The interpretation of this edict is clear when the receiving node is a terminating element, that is, a tPSP. The interpretation of the edict is less clear for intermediary nodes since messages are interpreted, altered and then passed on to the next node in the chain. Interpreting a message may result in unrecognized extensions being "ignored" and subsequently thrown away when the message is altered and passed on. It is a fundamental requirement for implementations of the present document NOT TO DISCARD any information from messages received, processed and subsequently passed on to other PEMEA nodes. That is, all valid XML objects received SHALL be passed to subsequent PEMEA nodes.

10 PEMEA XML structures and messages

10.1 Ps Introduction

This clause is broken into a series of parts, with each part describing important types and structures. It is broken down like this to try and help the reader understand how the messages are put together. Where deemed helpful in a clause an XML fragment showing how to use the structure is included. In clause 16 is the schema in its entirety. Each XML fragment has been validated against its base schema for correctness.

The PEMEA protocol messages are defined as XML documents that SHALL be encoded in UTF-8.

10.2 Timestamps

PEMEA is specified using XML and timestamps are defined as XML dateTime types. While this definition does allow for the specification of timezone, all timeStamps contained in a route element shall be specified in UTC time. This ensures that if different entities in the message flow are in different timezone it is easy to determine when certain events occurred without the need for timezone conversion.

This restriction does not apply to conveyed entities such as timestamps inside the PIDF-LO, though UTC time is preferable.

10.3 General types

10.3.1 pemea:posIntType

Type	Values	Description
pemea:posIntType	Zero and positive integers	N/A

10.3.2 pemea:nodeType

Type	Parameter	Param-Type	Presence	Description
pemea:nodeType	position	pemea:posIntType	Mandatory	A number representing the position of the node in the routing chain. The node initiating the message shall have a position of 0
	timeStamp	xs:dateTime	Mandatory	The date and time that the message was sent specified in UTC time: 2015-05-16T16:46:32Z
	node	xs:anyURI	Mandatory	The URI used to send messages to the node: https://psp.oonagal.zz:9001

10.3.3 pemea:hopsType

Type	Parameter	Param-Type	Presence	Description
pemea:hopsType	hop	pemea:nodeType	Mandatory	List of nodes that the message has traversed.

```
<hops>
   <hop position="0" timeStamp="2016-01-14T19:43:00.001Z">
      <node>https://cooAP.example.com.be:2001/pemea/</node>
   </hop>
   <hop position="1" timeStamp="2016-01-14T19:43:00.098Z">
      <node>https://orig.psp.example.com:2001/pemea/</node>
   </hop>
</hops>
```

The position attribute is used to specify the order in which the <hop> occurred in the signalling chain. A document shall be deemed invalid if it contains two or more hops with the same position attribute values. Receiving such a message shall result in the receiver sending an error to the sender and terminating any further forwarding of the emergencyDataSend (EDS) message. See Table 12 for error reason codes.

10.3.4 pemea:routeType

Type	Parameter	Param-Type	Presence	Description
pemea:routeType	msgSeq	xs:token	Mandatory	Message identifier provided by the initial originator of the emergencyDataSend message. In the present document this value is set by the AP.
	hops	pemea:hopsType	Mandatory	The hops parameter contains the identity of each node through which a message has passed.

```
<route msgSeq="CoolAP-7496" >
   <hops>
      <hop position="0" timeStamp="2016-01-14T19:43:00.001Z">
         <node>https://cooAP.example.com.be:2001/pemea/</node>
      </hop>
      <hop position="1" timeStamp="2016-01-14T19:43:00.098Z">
         <node>https://orig.psp.example.com:2001/pemea/</node>
      </hop>
   </hops>
</route>
```

10.3.5 pemea:destinationType

Parameter	Type	Description
pemea:destinationType	xs:enumeration	One of the following values: • PSAP • PSP • ASP

10.3.6 pemea:destinationNodeType

Type	Param-Type	Presence	Description
pemea:destinationNodeType	xs:token	Conditional	If a destination node does not have a URI then it may describe itself using a name. Only a PSAP may use this option.
	xs:anyURI	Conditional	If the destination node is identifiable by a URI then it shall use this form.

10.3.7 pemea:deliveryType

Type	Parameter	Param-Type	Presence	Description
pemea:deliveryType	destType	pemea:destinationType	Mandatory	Type of node the data was delivered: • PSAP • PSP • ASP See note.
	--	pemea:destinationNodeType	Mandatory	The name or address of the entity to which the message/data was delivered.
NOTE: A PSP can deliver data to a PSAP, ASP or to another PSP.				

```
<!—Example One Token -->
<delivery destType="PSAP">PSAP connected to oPSP</delivery>

<!—Example Two URI -->
<delivery destType="PSAP">https://psap.opsp.example.com:2001/pemea/</delivery>
```

10.3.8 pemea:typeOfCallerIdType

Type	Param-Type	Presence	Description
pemea:typeOfCallerIdType	xs:token	Conditional	If the caller identifier is not something that can be expressed as a URI, then it may use a token instead.
	xs:anyURI	Conditional	This is the normal form of caller identifier and any identifier that can be expressed as a URI shall use this form.

10.3.9 pemea:callerIdType

Type	Parameter	Param-Type	Presence	Description
pemea:callerIdType	typeOfId	xs:token	Mandatory	This is the type of identifier being used, in terms of what communications application the identifier is applicable to. Since these will grow over time, a registry of valid values is established in the present document.
	--	pemea:typeOfCallerIdType	Mandatory	This is the actual value of the caller id.

Table 9: Caller-Id Token Types

Value	Type	Description
msisdn	tel uri	An MSISDN of the caller expressed as a tel uri
skypeName	xs.token	The identifier used by the caller when using Skype^{TM}
WhatsAppId	tel uri	The identifier used by the caller when using WhatsApp^{TM}

10.3.10 pemea:callerIdListType

Type	Parameter	Param-Type	Presence	Description
pemea:callerIdListType	callerId	pemea:callerIdType	Mandatory	List of possible caller Ids

```
<callerIds>
   <callerId typeOfId="msisdn">tel:+44-555-555-1234</callerId>
   <callerId typeOfId="msisdn">tel:+34-555-222-6789</callerId>
   <callerId typeOfId="skypeName">winterb</callerId>
</callerIds>
```

10.3.11 pemea:informationType

The informationType is the means by which the AP "More Information" capability described in Annex D of the PEMEA Requirements and Functional Architecture document [i.2] is implemented. All information/resources are accessed in the AP using "reach-back" URIs provided by the AP to the destination PSAP. A PSAP not understanding or implementing one or any of these capabilities shall ignore the URIs.

A registry for the types of information is established and the initial entries are in Table 10. Negotiation mechanisms for the streaming capabilities are for further study.

A registry of protocols for use with the types of information when more than one protocol type may be available for the same URI scheme is established and initial entries are in Table 11.

Use of these URIs is subject to the description provided in clause 8.1.

Type	Parameter	Param-Type	Presence	Description
pemea:informatIonType	typeOfInfo	xs:token	Mandatory	Type of additional information being provided. Allowed values to be defined in a registry, initial values are in Table 10.
	protocol	xs:token	Optional	The protocol that the AP will accept for the service defined in the typeOfInfo attribute. For example, the AP may support HELD and MLP for location updates and both provide HTTP URIs.
	Value	xs:anyURI	Mandatory	The URI through which the resource can be contacted.

Table 10: AP Information Type Registry

Value	Description
Location_Update	Provides the most up to date location available to the PSAP. If not protocol attribute is specified then the request is treated as a HELD location request per the HELD de-reference specification [15]. Other possible protocols are specified in Table 11.
Web	General Webpage content.
RTT	URI through which an RTT media stream can be negotiated with the caller. For further Study.
Audio	URI through which an audio media stream can be negotiated with the caller. For further Study.
Video	URI through which a video media stream can be negotiated with the caller. For further Study.
Audio_Video	URI through which an audio and video media stream can be negotiated with the caller. For further Study.
IM	URI through which text message can be exchanged with the caller. For further Study.
Medical	URI through which the PSAP can acquire caller medical records (format to be determined). For further Study.
SIP_Request	URI through which the PSAP/PSP can provide a SIP URI to the Application. Required for NG112 alignment.

Table 11: AP Information Type Protocol Registry

Info type Value	Protocol Token	Description
Location_Update	HELD_Deref	Location requested using a HELD location request per the HELD de-reference specification [15].
	MLP_3.2	Mobile Location Protocol Version 3.2 [17].
	MLP_3.3	Mobile Location Protocol Version 3.3 [18].
	MLP_3.4	Mobile Location Protocol Version 3.4 [19].
SIP_Request	sip	Requesting a PSAP/PSP SIP URI to which the device can send an INVITE.
	sips	Requesting a PSAP/PSP SIPS URI to which the device can send an INVITE.

10.3.12 pemea:apMoreInfoType

This is the container into which any additional AP information URI are placed. These URIs are referred to as "reach-back" URIs as they enable a tPSP or PSAP to "reach-back" to the originating AP for more information to establish media sessions. Presence of the apMoreInfoType container in an emergency data send message is optional, however, if it is present then at least one information element shall be provided.

Type	Parameter	Param-Type	Presence	Description
pemea:apMoreInfoType	information	pemea:informationType	Mandatory	At least one element shall be provided. Each element represents a capability that the AP supports and is prepared to make available to the PSAP.

```
<apMoreInformation>
   <information typeOfInfo="IM">https://coolap.example.com.be:2002/im?id=CoolAP-7496</information>
   <information typeOfInfo="Video">https://coolap.example.com.be:2007/vid?id=CoolAP-
7496</information>
   <information typeOfInfo="Web">https://coolap.example.com.be:2022/Web?id=CoolAP-7496</information>
   <information typeOfInfo="Location_Update"
                protcol="HELD">
                https://coolap.example.com.be:2096/Web?id=CoolAP-7496
   </information>
</apMoreInformation>
```

10.3.13 pemea:accessDataType

10.3.13.1 pemea:accessDataType structure

Defines the inclusion of the network and WiFi types from HELD Measurements, IETF RFC 7105 [12].

This is defined a choice, so only one element may be present.

Type	Parameter	Param-Type	Presence	Description
pemea:accessDataType	network	cell:network (refers to urn:ietf:params:xml:ns:geopriv:lm:cell)	Conditional	Is present if wifi is not.
	wifi	wifi:wifi (refers to urn:ietf:params:xml:ns:geopriv:lm:wifi)	Conditional	Is present if network is not.

10.3.13.2 network

IETF RFC 7105 [12], clause 5.4 defines the cellular elements with the XML schema defined in clause 8.7 of IETF RFC 7105 [12]. The schema defines a number of different cell representations.

Element	Parameter	PEMEA Presence	Type	Description
network	mcc	Conditional	element	Mobile Country Code. A three digit code that represents the country in which the serving cell is located. This is obtained using an API in the mobile device. It shall be provided if the nid element is not used. It shall be absent if the nid element is present.
	mnc	Conditional	element	Mobile Network Code. A two or three digit number representing the operator that owns the serving cell. This value is obtained using an API in the mobile device. It shall be provided if the nid element is not used. It shall be absent if the nid element is present.
	nid	Conditional	element	This is a non-negative integer representing the complete cell-id, MCC+MNC+CID. It shall not be included if the mcc and mnc elements are present.

Example:

```
<network xmlns="urn:ietf:params:xml:ns:geopriv:lm:cell">
   <mcc>214</mcc>
   <mnc>01</mnc>
</network>
```

10.3.13.3 wifi

The elements for this clause are specified in IETF RFC 7105 [12], clause 5.3.

Element	Parameter	PEMEA Presence	Type	Description
wifi	ap	Mandatory	element	This element shall be present if the wifi element is included.

The schema supports multiple access points (APs), however, only the serving AP is required.

Element	Parameter	PEMEA Presence	Type	Description
ap	serving	Mandatory	boolean	This indicates that the access point (AP) is the serving AP and shall be set to true.
	bssid	Mandatory	element	This is the base service set identifier (bssid) is the 48 bit MAC address of the access point.

```
<wifi:wifi>
   <wifi:ap serving="true">
      <wifi:bssid>AB-CD-EF-AB-CD-EF</wifi:bssid>
   </wifi:ap>
</wifi:wifi>
```

10.3.14 pemea:accessData

Contains information the App provides about the access network it is attached to. It explicitly includes the pemea:accessDataType, but also allows any other types to be added as required in the future by use of an xml extension point.

Type	Parameter	Param-Type	Presence	Description
pemea:accessData	network	accessData	Conditional	Present if provided. Full details provided in IETF RFC 7105 [12].
	wifi	accessData	Conditional	Present if provided. Full details provided in IETF RFC 7105 [12].

```
<accessData>
   <accessDataType>
      <cell:network>
         <cell:mcc>255</cell:mcc>
         <cell:mnc>023</cell:mnc>
      </cell:network>
   </accessDataType>
</accessData>
```

10.3.15 pemea:msgInfoType

This is the container into which a free text message can be placed. It allows the specification of the language in which the message is written.

Type	Parameter	Param-Type	Presence	Description
pemea:msgInfoType	lang	xs:lang	Mandatory	The language in which the message is written.
	value	xs:token	Mandatory	The message being provided.

```
<msgInfoType lang="en">This is the message that goes here</msgInfoType>
```

11 PEMEA Message Definition

11.1 emergencyDataSend Message

11.1.1 emergencyDataSend message structure

The emergencyDataSend (EDS) message is the means by which data is conveyed from the AP to the destination PSAP, all intermediary nodes transfer this type of message. Example does not show additional data.

Parameter	Presence	Type	Description
ttl	Mandatory	pemea:posIntType	Indicates the maximum number of hops that the message may take before it is discarded.
onErrorPost	Conditional	xs:anyURI	Provides a URI at the AP for receiving error messages. If this attribute is present any entity receiving an error message in response to having sent an EDS message shall send the same error message to the URI specified in the attribute. Use of this attribute is recommended to assist with identifying network issues. Use of this attribute is mandatory if the apMoreInformation element is present. Refer to clause 11.1.3 or more details.

Parameter	Presence	Type	Description
onCapSupportPost	Conditional	xs:anyURI	Provides a URI at the AP that a terminating PSAP or PSP shall post to on receipt of an EDS. If this attribute is present then the terminating PSAP or PSP shall post to this URI on receipt of an EDS. Use of this attribute by the AP in an EDS is mandatory if the apMoreInformation element is present. Refer to clause 11.1.4 for more details.
route	Mandatory	pemea:routeType	Indicates the path that the message has followed up to this point.
callerIds	Mandatory	pemea:callerIdListType	The list of possible caller identifiers that may have been used to initiate the call. At the time of writing this is only expected to consist of a list containing one or more MSISDNs thereby supporting multi-sim devices.
apMoreInformation	Optional	pemea:apMoreInfoType	The list of URIs that the AP supports for the PSAP to communicate with or get more information about the caller. Only the AP may add the apMoreInformation element to the EDS message. No entities other than the AP shall add information elements to the apMoreInformation element. No elements shall remove or change information elements in the apMoreInformation element.
accessData	Conditional	pemea:accessDataType	The list of access information that the App is able to provide at call time.
pidfLo	Mandatory		Contains all of the information associated with the caller.

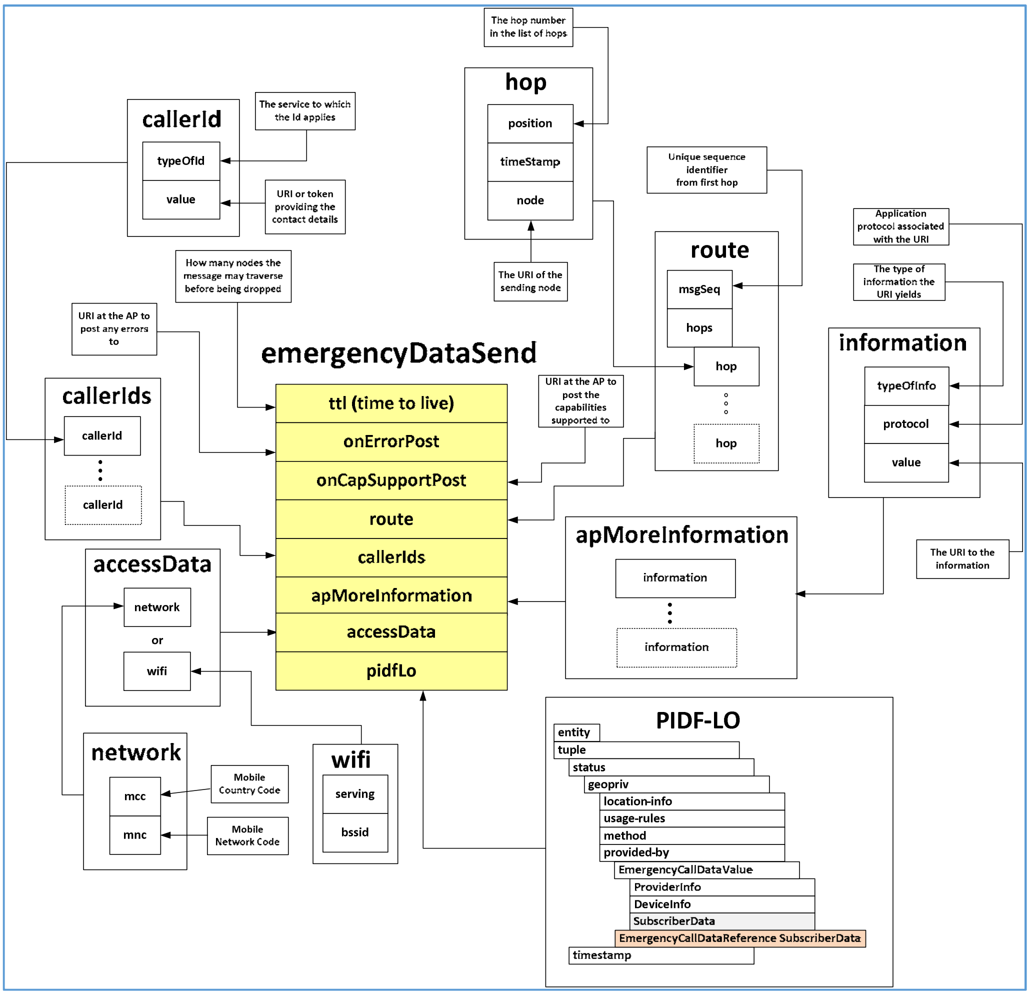


Figure 11: emergencyDataSend message structure

11.1.2 emergencyDataSend example

```
<emergencyDataSend xmlns="urn:pemea:apps:xml:ns:pemea:base"
                   xmlns:pdf="urn:ietf:params:xml:ns:pidf"
                     xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
                   xmlns:gml="http://www.opengis.net/gml"
                   xmlns:gs="http://www.opengis.net/pidflo/1.0"
                   xmlns:cell="urn:ietf:params:xml:ns:geopriv:lm:cell"
                   xmlns:con="urn:ietf:params:xml:ns:geopriv:conf"
         ttl="5"
         onErrorPost="https://cooAP.example.com.be:2001/pemea/error/CoolAP-7496"
         onCapSupporPost="https://cooAP.example.com.be:2001/pemea/cap/CoolAP-7496">
  <route msgSeq="CoolAP-7496" >
    <hops>
      <hop position="0" timeStamp="2016-02-01T19:43:00+11:00" >
        <node>https://cooAP.example.com.be:2001/pemea/ </node>
      </hop>
    </hops>
  </route>
  <callerIds>
    <callerId typeOfId="msisdn">tel:+44-555-555-1234</callerId>
    <callerId typeOfId="msisdn">tel:+34-555-222-6789</callerId>
    <callerId typeOfId="skypeName">winterb</callerId>
  </callerIds>
  <apMoreInformation>
    <information typeOfInfo="IM">https://coolap.example.com.be:2002/im?id=CoolAP-7496</information>
    <information typeOfInfo="Video">https://coolap.example.com.be:2007/vid?id=CoolAP-
7496</information>
    <information typeOfInfo="Web">https://coolap.example.com.be:2009/Web?id=CoolAP-
7496</information>
    <information typeOfInfo="Location_Update">
                 https://coolap.example.com.be:2045/loc?id=CoolAP-7496
    </information>
  </apMoreInformation>
  <accessData>
    <cell:network>
      <cell:mcc>255</cell:mcc>
      <cell:mnc>023</cell:mnc>
    </cell:network>
  </accessData>
  <pdf:presence entity=" tel:+44-555-555-1234">
    <pdf:tuple id="circle">
      <pdf:status>
        <gp:geopriv>
          <gp:location-info>
            <gs:Circle srsName="urn:ogc:def:crs:EPSG::4326">
              <gml:pos>42.5463 -73.2512</gml:pos>
              <gs:radius uom="urn:ogc:def:uom:EPSG::9001">
                   30.0
              </gs:radius>
            </gs:Circle>
            <con:confidence pdf="normal">95</con:confidence>
          </gp:location-info>
          <gp:usage-rules/>
          <gp:method>GNSS</gp:method>
        </gp:geopriv>
      </pdf:status>
    </pdf:tuple>
  </pdf:presence>
</emergencyDataSend>
```

11.1.3 onErrorPost usage details

The onErrorPost attribute is a URI provided by the AP to which error messages related to the delivery of an EDS message are sent. The URI is specified as an HTTPS URI and the same authentication and authorization procedures defined in clause 9.2 shall apply.

Any PEMEA entity receiving a PEMEA error in response to sending an EDS, where the EDS message contained an onErrorPost attribute and URI, shall post this same PEMEA error to the URI contained in the onErrorPost attribute.

If an entity sending an EDS, where the EDS message contains an onErrorPost attribute and URI, receives an HTTP error form the destination server, then the sending entity shall construct a PEMEA error message and post it to the URI contained in the onErrorPost attribute. The reason token in the error message shall be set to "httpError", and the route in the PEMEA error message shall be the same as the route that was in the EDS that was being sent to the rejecting node. The message element should contain the type of HTTP error received.

If the URI is invalid, or an error is received from the server addressed by the URI then the PEMEA entity may log the error but does not need to take any further action.

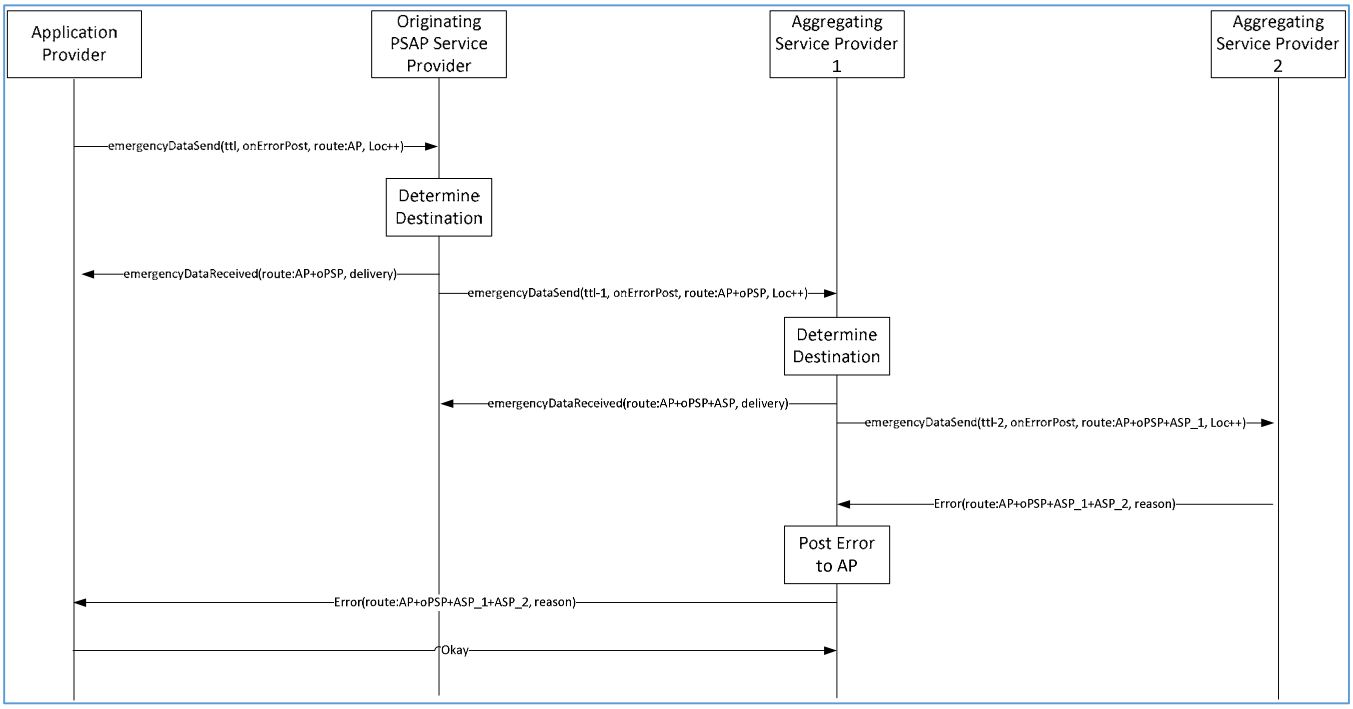

Application Provider
Originating PSAP Service Provider
Aggregating Service Provider 1
Aggregating Service Provider 2
emergencyDataSend(ttl, onErrorPost, route:AP, Loc++)
Determine Destination
emergencyDataReceived(route:AP+oPSP, delivery)
emergencyDataSend(ttl-1, onErrorPost, route:AP+oPSP, Loc++)
Determine Destination
emergencyDataReceived(route:AP+oPSP+ASP, delivery)
emergencyDataSend(ttl-2, onErrorPost, route:AP+oPSP+ASP_1, Loc++)
Error(route:AP+oPSP+ASP_1+ASP_2, reason)
Post Error to AP
Error(route:AP+oPSP+ASP_1+ASP_2, reason)
Okay

Figure 12: onErrorPost message flow

11.1.4 onCapSupportPost usage details

The onCapSupportPost attribute is a URI provided by the AP to which a terminating PSAP or PSP indicates which proffered capabilities contained in the information elements of the apMoreInformation element the terminating entity supports.

The URI is specified as an HTTPS URI and the same authentication and authorization procedures defined in clause 9.2 shall apply. If the AP determines that the posting entity is not a PSAP or PSP then the AP shall respond with an HTTP 403 "Forbidden" response.

If the onCapSupportPost attribute is provided in the EDS and the EDS does not contain an apMoreInformation element then the terminating PSAP or PSP shall post an empty message to the provided URI. This mechanism signifies receipt of the EDS message by a terminating entity to the AP.

If the onCapSupportPost attribute is provide in the EDS message and an apMoreInformation element is present in the EDS but the terminating PSAP or PSP does not support any of the information elements then the terminating PSAP or PSP shall post an empty message to the provided URI. This mechanism signifies receipt of the EDS message by a terminating entity to the AP and that terminating entity does not support any of the information elements expressed in the apMoreInformation element.

If the onCapSupportPost attribute is provided in the EDS message and an apMoreInformation element is present in the EDS and the terminating PSAP or PSP supports one or more of the information elements then the terminating PSAP or PSP shall:

1) construct an apMoreInformation element containing all information elements received in the EDS that are supported by the PSAP or PSP;

2) place the constructed apMoreInformation element into the body of an HTTP POST message;

3) POST the HTTP message to the URI contained in the onCapSupportPost attribute.

If a terminating PSAP or PSP were to receive the EDS from the example in clause 11.1.2 and the PSAP/PSP could only support Video and Location_Updates then the PSAP/PSP would post a message to the onCapSupportPost URI with the following body:

```
<?xml version="1.0" encoding="UTF-8"?>
<apMoreInformation xmlns="urn:pemea:apps:xml:ns:pemea:base">
    <information typeOfInfo="Video"/>
    <information typeOfInfo="Location_Update" protocol="HELD"/>
</apMoreInformation>
```

If the posting entity receives an error from the AP in response to the post, then the posting entity may log the error. No further action is required on the part of the posting entity.

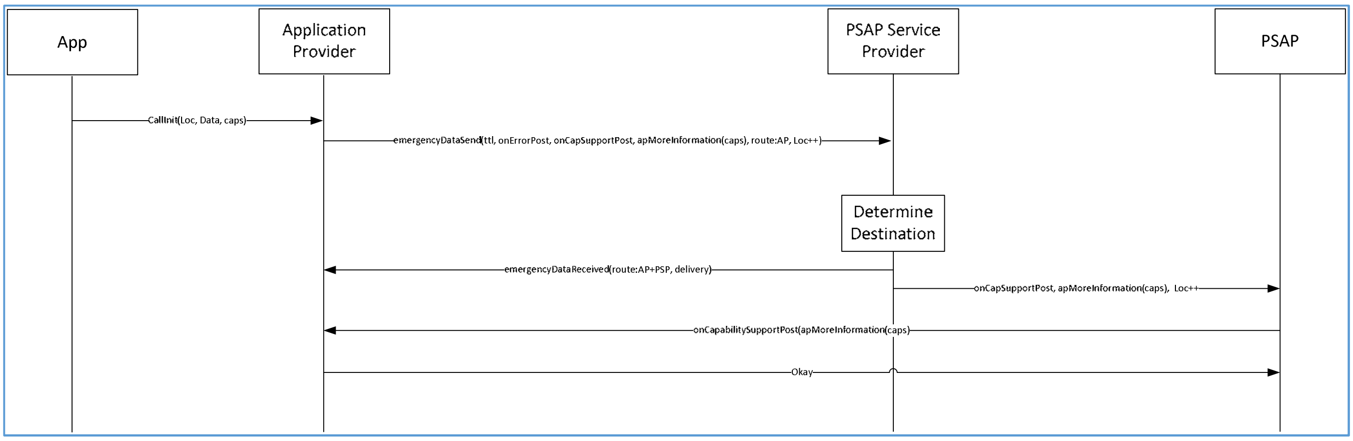

App
Application Provider
PSAP Service Provider
PSAP
CallInit(Loc, Data, caps)
emergencyDataSend(ttl, onErrorPost, onCapSupportPost, apMoreInformation(caps), route:AP, Loc++)
Determine Destination
emergencyDataReceived(route:AP+PSP, delivery)
onCapSupportPost, apMoreInformation(caps), Loc++
onCapabilitySupportPost(apMoreInformation(caps)
Okay

Figure 13: onCapSupportPost message sequence

11.2 emergencyDataReceived message

11.2.1 emergencyDataReceived message structure

Parameter	Presence	Type	Description
timeStamp	Mandatory	xs:timeStamp	Time stamp that the message is sent specified in UTC.
route	Mandatory	pemea:routeType	Indicates the path that the message has followed up to this point.
delivery	Mandatory	pemea:deliveryType	The node to which the data has just been sent.

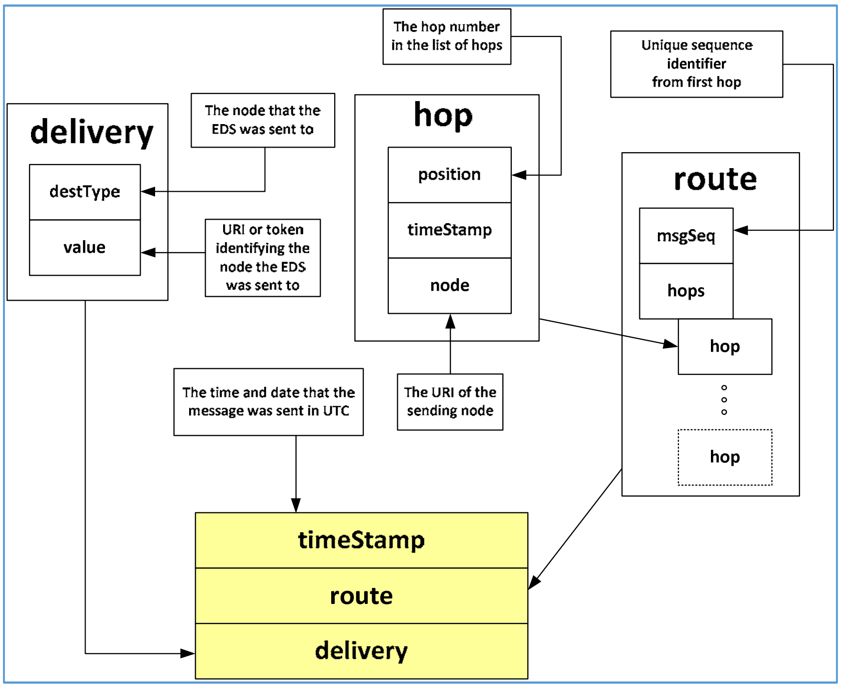

The hop number in the list of hops
Unique sequence identifier from first hop
delivery
destType
value
The node that the EDS was sent to
URI or token identifying the node the EDS was sent to
hop
position
timeStamp
node
route
msgSeq
hops
hop
hop
The time and date that the message was sent in UTC
The URI of the sending node
timeStamp
route
delivery

Figure 14: emergencyDataReceived message structure

11.2.2 emergencyDataReceived example

```
<emergencyDataReceived xmlns="urn:pemea:apps:xml:ns:pemea:base" timeStamp="2016-01-
14T19:43:01.521Z">
   <route msgSeq="CoolAP-7496" >
      <hops>
         <hop position="0" timeStamp="2016-01-14T19:43:00.001Z">
            <node>https://cooAP.example.com.be:2001/pemea/</node>
         </hop>
         <hop position="1" timeStamp="2016-01-14T19:43:00.098Z">
            <node>https://orig.psp.example.com:2001/pemea/</node>
         </hop>
      </hops>
   </route>
   <delivery destType="PSAP">https://psap.opsp.example.com:2001/pemea/</delivery>
</emergencyDataReceived>
```

11.3 error message

Parameter	Presence	Type	Description
timeStamp	Mandatory	xs:timeStamp	Time stamp that the message is sent specified in UTC.
reason	Mandatory	xs:token	The reason that the error was generated based on a token value defined in Table 10.
message	Optional	pemea:msgInfoType	The human readable text message describing the problem. Note that the language is a mandatory attribute.
route	Mandatory	pemea:routeType	Indicates the path that the message has followed up to this point.

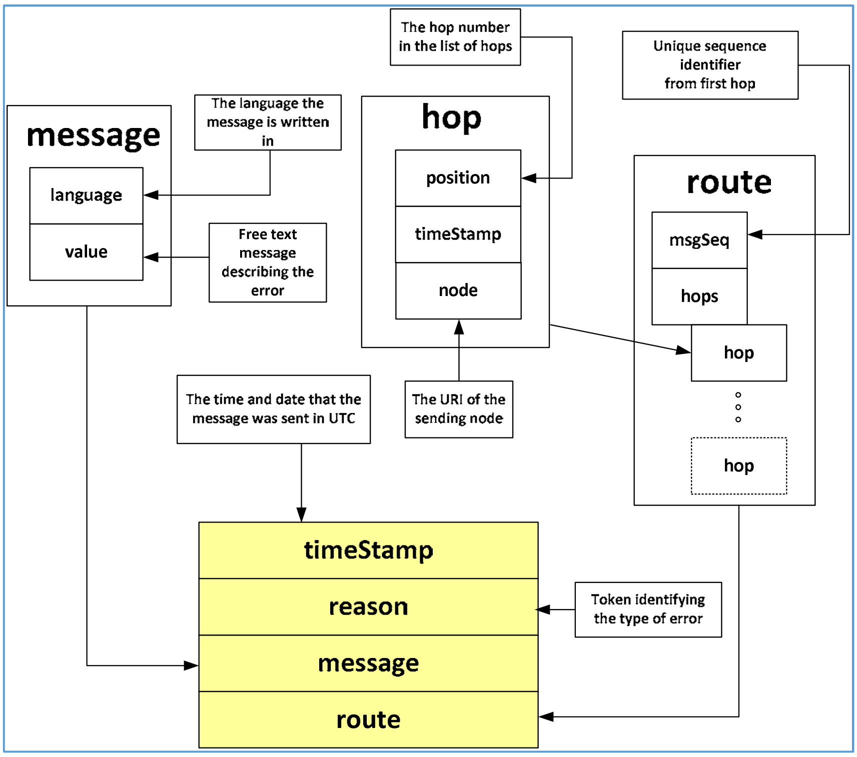

The hop number in the list of hops
Unique sequence identifier from first hop
The language the message is written in
message
language
value
Free text message describing the error
hop
position
timeStamp
node
route
msgSeq
hops
hop
hop
The time and date that the message was sent in UTC
The URI of the sending node
timeStamp
reason
message
route
Token identifying the type of error

Figure 15: Error message structure

Table 12: Error reasonToken values

Value	Description
ttlExhausted	The time to live value reached zero and the message was not delivered to a PSAP.
noRoute	The entity currently responsible for routing the message does not have a relationship with any entity that can receive it.
badMessage	The message could not be understood by the receiving entity (normally this message will be sent by the home PSP).
circularRoute	The entity currently trying to route the message has identified that the next hop it would send the message to is already in the route element.
duplicateHopPosition	The route element contained two or more hops with the same position attribute value.
httpError	The entity trying to send the EDS message encountered an HTTP error from the next hop. The error type should be contained in the message element.

```
<error xmlns="urn:pemea:apps:xml:ns:pemea:base" timeStamp="2016-01-14T19:43:01.521Z">
   <reason>noRoute</reason>
   <message lang="en">Cannot find route for location provided</message>
   <route msgSeq="CoolAP-7496" >
      <hops>
         <hop position="0" timeStamp="2016-01-14T19:43:00.001Z">
            <node>https://cooAP.example.com.be:2001/pemea/</node>
         </hop>
         <hop position="1" timeStamp="2016-01-14T19:43:00.098Z">
            <node>https://asp1.example.com:2001/pemea/</node>
         </hop>
      </hops>
   </route>
</error>
```

12 PEMEA PIDF-LO Profiling

12.1 Rationale

The Presence Information Data Format (PIDF) [7] Location Object (LO) [3] is a highly extensible data structure. IETF schemas are not repeated in the present document, however references are provided to the documents where these schemas are specified. Note that SubscriberData may be sent either by value or by reference depending on jurisdictional constraints.

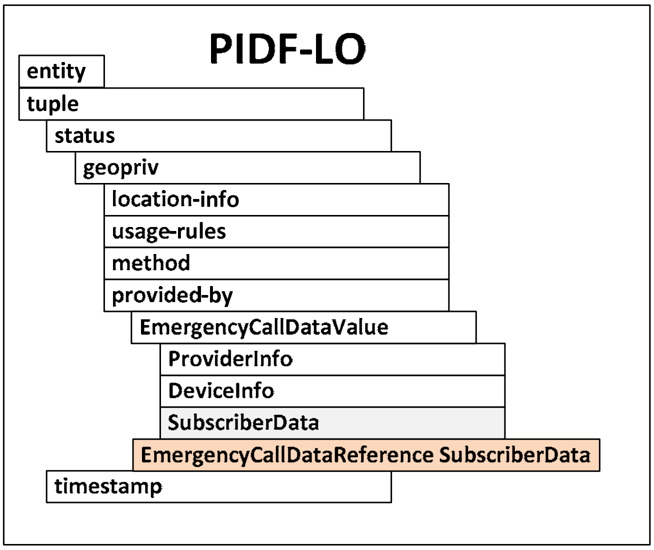


Figure 16: PEMEA PIDF-LO Structure

12.2 entity

Parameter	Presence	Type	Value	Description
entity	Mandatory	xs:anyURI	uri	This value shall be expressed as a URI and should be a URI from the callerIds list if a URI is provided as a URI.

12.3 tuple

Element	Parameter	PEMEA Presence	Type	Value	Description
tuple	id	Mandatory	xs:ID	Unique token	Used to identify the tuple within the PIDF-LO document. See note.
	status	Mandatory	element	--	Container for the geopriv object.
NOTE: IETF RFC 5491 [4] allows geopriv elements to be contained in a <person> or <device> element in addition to a <tuple> element. PEMEA shall only support geopriv elements being provided in a <tuple> element.					

12.4 status

Element	Parameter	PEMEA Presence	Type	Value	Description
status	geopriv	Mandatory	element	--	Container for location and user information

12.5 geopriv

12.5.1 geopriv element profile

Element	Parameter	PEMEA Presence	Type	Value	Description
geopriv	Location-info	Mandatory	element	--	Container for location and user information.
	usage-rules	Mandatory	element	--	Container for usage rules.
	method	Mandatory	element	From IANA [10]	The value for the method should be based on the values from the IANA registry [10].
	provided-by	Mandatory	element	--	Used to provide all supplementary user data.

12.5.2 location-info

12.5.2.1 location-info profile

The location-info element as defined in the PIDF-LO [3] is open to any XML namespace. PEMEA constrains the location types to those described in the PIDF-LO profile specification IETF RFC 5491 [4].

Element	Parameter	PEMEA Presence	Type	Description
Location-info	gml:Point	Conditional	element	A 2 dimensional or 3 dimensional point in space form the GML specification.
	gml:Polygon	Conditional	element	Construction for defining a 2D polygon from the GML specification.
	gs:Circle	Conditional	element	Construct for defining a circle from the GeoShape specification.
	gs:Ellipse	Conditional	element	Construct for defining an ellipse from the GeoShape specification.
	gs:ArcBand	Conditional	element	Construct for defining an arcband from the GeoShape specification.
	gs:Sphere	Conditional	element	Construct for defining a sphere from the GeoShape specification.
	gs:Ellipsoid	Conditional	element	Construct for defining an ellipsoid from the GeoShape specification.
	gs:Prism	Conditional	element	Construct for defining an extruded polygon (prism) from the GeoShape specification.
	confidence	Conditional	element	Defines the confidence level in the location information being provided.
	civicAddress	Conditional	element	Construct for defining civic address elements from IETF RFC 5139 [8] and civic extension complying with IETF RFC 6848 [9].

12.5.2.2 Confidence

IETF RFC 7459 [6] introduced an explicit means of representing confidence and uncertainty into locations provided in a PIDF-LO. Inclusion of this parameter is mandatory with all shape-types with the exception of the point types. If the location API of the device provided a confidence then this value can be explicitly provided, if the API does not provide this value then including the confidence with a value of "unknown" ensures that the receiving entity does not assume a value of 95 % as stipulated in IETF RFC 5491 [4].

Element	Parameter	Default Value	Description
confidence	pdf	unknown	Probability Density Function. For GNSS solutions this shall be set to "normal", for other determination mechanisms it should be set to "unknown".
	Element value	--	This is a value between 0 and 100 representing the confidence in the location being provided.

12.5.3 usage-rules

Element	Parameter	PEMEA Presence	Type	Default Value	Description
usage-rules	retransmission-allowed	Mandatory	element	yes	This value should be set to "yes" whenever the application user may be roaming. See note.
	retention-expiry	Mandatory	xs:dateTime	--	This value is set by the AP based on user policy. If no user policy is provided then the AP should set the value to 1 hour after the time the call for help is initiated by the user.
NOTE: If this parameter is set to "no", then the ttl value is ignored by the PSP and data from the AP can only be delivered from the PSP to a directly-connected PSAP. The EDS shall not be passed to a second PSP or an ASP.					

12.5.4 method

The method parameter is an element in the geopriv structure that is used to describe how the location information was determined. Often, this will be provided by the location API in the device, sometimes it is not. If is provided then one of values in the IANA registry [10] should be used. If the method is not provided or is unknown then "*Unknown*" should be used. While this value is defined in IANA, it informs downstream entities that the way in which location was determined is not known.

12.5.5 provided-by

Element	Parameter	PEMEA Presence	Scheme	Description
provided-by	provided-by	Mandatory	Additional-Data	This element contains the provided-by element taken the from the "additional-data" scheme in IETF RFC 7852 [11].

12.6 timestamp

This occurs after the status stanza is closed. It expresses the time that the PIDF-LO was created and in the context of PEMEA is represents the date and time that the AP received the location and other caller information associated with the current emergency.

12.7 PIDF-LO example

The following XML fragment is of a PIDF-LO. The Additional-Data information is not included for brevity.

```
<presence
    xmlns="urn:ietf:params:xml:ns:pidf"
    xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
    xmlns:gml="http://www.opengis.net/gml"
    xmlns:gs="http://www.opengis.net/pidflo/1.0"
    xmlns:con="urn:ietf:params:xml:ns:geopriv:conf"

    entity="tel:+44-555-555-1234">
       <tuple id="circle">
         <status>
           <gp:geopriv>
             <gp:location-info>
               <gs:Circle srsName="urn:ogc:def:crs:EPSG::4326">
                 <gml:pos>42.5463 -73.2512</gml:pos>
                 <gs:radius uom="urn:ogc:def:uom:EPSG::9001">
                   30.0
                 </gs:radius>
               </gs:Circle>
               <con:confidence pdf="normal">95</con:confidence>
             </gp:location-info>
             <gp:usage-rules/>
             <gp:method>GNSS</gp:method>
           </gp:geopriv>
         </status>
```

```
        </tuple>
</presence>
```

13 PEMEA Additional-Data Profiling

13.1 Rationale

The additional-data specification IETF RFC 7852 [11] was explicitly designed to provide containers for additional information about an emergency call, including who is making the call, what kind of device and network they are using and who the various providers are. The type and nature of the information being conveyed requires the data structures to be very flexible. PEMEA requires that the structures are used in a specific way so the structures are profiled for use in the PEMEA data exchanges, ensuring that all entities are able to interpret information provided different sources.

13.2 Additional-Data :- provided-by

PEMEA prefers to pass additional data by-value, clause 9.3 describes situations where this may not be allowed. Information relating to the AP shall always be passed by-value, however, caller information shall only be sent by reference. Entity authorization is described in clause 9.2.

Element	Parameter	PEMEA Presence	Type	Description
provided-by	EmergencyCallDataValue	Mandatory	EmergencyCallDataValueType	This element contains the provided-by element defined in the Additional-Data document IETF RFC 7852 [11].
	EmergencyCallDataReference	Conditional	ByRefType	Caller information is sensitive and shall be sent by reference. This element is defined in the Additional-Data document IETF RFC 7852 [11].

13.3 EmergencyCallDataValue

Element	Parameter	PEMEA Presence	Type	Description
EmergencyCallDataValue	EmergencyCallData.ProviderInfo	Mandatory	ProviderInfoType	This information is filled in by the AP.
	EmergencyCallData.DeviceInfo	Optional	DeviceInfoType	This information may be provided by the application to the AP either at call time or ahead of call time.

13.4 EmergencyCallData.ProviderInfo

13.4.1 EmergencyCallData.ProviderInfo profile

Inclusion of this element is mandatory. It conveys information for contacting the AP that is originating an EDS message.

A full example is of an XML EmergencyCallData.ProviderInfo structure that is compliant with the present document is provided in clause 13.4.3.

Element	Parameter	PEMEA Presence	Type	Description
EmergencyCallData.ProviderInfo	DataProviderReference	Mandatory	xs:token	This information is provided by the AP and shall be the same for all DataProviderReference elements in all additional data blocks in the same emergency DataSend message.
	DataProviderString	Mandatory	xs:string	A plain text string containing the name of the Application Provider. The Additional-Data specification IETF RFC 7852 [11] does provide for setting a language for this field. The value of the first Language element shall be used as the language to interpret this field.
	ProviderID	Mandatory	xs:string	Value provided by PRA after protocol certification.
	ProviderIDSeries	Mandatory	xs:string	This shall have a value of PRA.
	TypeOfProvider	Mandatory	xs:string	This value shall be set to "Application Provider".
	ContactURI	Mandatory	xs:anyURI	This value shall contain the telephone number of the AP expressed as a tel uri [5].
	Language	Mandatory	Restricted xs:string	There shall be one, but the AP may support more than one spoken language. The list of acceptable abbreviation can be found in [13].
	DataProviderContact	Mandatory	vcard	This supplies more information about the application provider. Only one vcard is allowed DataProviderContact in PEMEA.

13.4.2 DataProviderContact :- vcard

13.4.2.1 DataProviderContact :- vcard profile

The vcard is used extensively in the additional data specification. This is a very flexible and potentially unstructured information container. This table attempts to add structure to the data provider contact information. The information in this table shall be provided, other fields may be provided.

Element	Field being represented	PEMEA Presence	vcard element	Description
vcard	Organization's full name	Mandatory	org	Contains the full name of the AP
	Organization's street address	Mandatory	adr	Street address of the AP
	Email	Mandatory	<email><text>	General support email address shall be provided
	Public website (URL)	Mandatory	<url><url>	Public email address for the AP

13.4.2.2 DataProviderContact :- org

Element	Element	PEMEA Presence	Type	Description
org	parameters	Mandatory	element	Holds the parameter that defines the language that the organization name is provided in.
	text	Mandatory	xs:string	Contains the full name of the AP.

Element	Element	PEMEA Presence	Value	Description
parameters	language	Mandatory	--	
	language-tag	Mandatory	Any abbreviation from [13].	PEMEA requires the language that the organization name is specified in. This shall be an abbreviation and the allowable abbreviations are specified in [13].

```
<vcard xmlns="urn:ietf:params:xml:ns:vcard-4.0">
   <org>
      <parameters>
         <language>
            <language-tag>en</language-tag>
         </language>
      </parameters>
      <text>Really Application Provider</text>
   </org>
</vcard>
```

13.4.2.3 DataProviderContact :- adr

Element	Element	PEMEA Presence	Type	Description
adr	parameters	Mandatory	element	Block defining parameters that describe the type of address.
	pobox	Optional	xs:string	Post office box of the AP if applicable. See note.
	ext	Optional	xs:string	Not used. See note.
	street	Mandatory	xs:string	Street address, street name, number, suite and floor if applicable.
	locality	Mandatory	xs:string	Municipality, city or suburb. May be empty if not applicable.
	region	Mandatory	xs:string	State or Province. May be empty if not applicable.
	code	Mandatory	xs:string	The post code of the AP.
	country	Mandatory	xs:string	Country name. Shall be provided.
NOTE: The contents for this parameter are optional in PEMEA, however, the vCard schema requires the presence of this element even if it is empty.				

Element	Element	PEMEA Presence	Value	Description
parameters	language	Mandatory	--	
	language-tag	Mandatory	Any abbreviation from [13].	PEMEA requires the language that the address is specified in. This shall be an abbreviation and the allowable abbreviations are specified in [13].

```
<vcard xmlns="urn:ietf:params:xml:ns:vcard-4.0">
   <adr>
      <parameters>
         <language>
            <language-tag>fr</language-tag>
         </language>
      </parameters>
     <pobox>77222</pobox>
     <ext/>
     <street>Avenue de la Toison d'Or, 79 - 3rd Floor</street>
     <locality>Brussels</locality>
     <region/>
     <code>1060</code>
     <country>Belgium</country>
   </adr>
</vcard>
```

13.4.2.4 DataProviderContact :- email

Element	Element	PEMEA Presence	Type	Description
email	text	Mandatory	xs:string	The primary email address of the Application Provider.

```
<vcard xmlns="urn:ietf:params:xml:ns:vcard-4.0">
   <email><text>jw@pemea.org</text></email>
</vcard>
```

13.4.2.5 DataProviderContact :- URL

Element	Element	PEMEA Presence	Type	Description
url	uri	Mandatory	xs:anyURI	The public website of the Application Provider

```
<vcard xmlns="urn:ietf:params:xml:ns:vcard-4.0">
   <url>
      <uri>http://www.pemea.help</uri>
   </url>
</vcard>
```

13.4.3 EmergencyCallData.ProviderInfo:- Complete Example

This clause provides an example of a complete PEMEA compliant EmergencyCallData.ProviderInfo structure.

```
<EmergencyCallData.ProviderInfo xmlns="urn:ietf:params:xml:ns:EmergencyCallData:ProviderInfo"
                                xmlns:xc="urn:ietf:params:xml:ns:vcard-4.0">
   <DataProviderReference>CoolAP-0xFF4568262458</DataProviderReference>
   <DataProviderString>Cool Application Provider</DataProviderString>
   <ProviderID>urn:pemea:pemea:ap:ID0x123FEDAC</ProviderID>
   <ProviderIDSeries>pemea</ProviderIDSeries>
   <TypeOfProvider>Application Provider</TypeOfProvider>
   <ContactURI>tel:+32-2534-9789</ContactURI>
   <Language>en</Language>
   <Language>fr</Language>
   <DataProviderContact>
      <xc:vcard>
         <xc:org>
            <xc:parameters>
               <xc:language>
                  <xc:language-tag>en</xc:language-tag>
               </xc:language>
            </xc:parameters>
            <xc:text>Really Application Provider</xc:text>
         </xc:org>
         <xc:adr>
            <xc:parameters>
               <xc:language>
                  <xc:language-tag>fr</xc:language-tag>
               </xc:language>
            </xc:parameters>
            <xc:pobox>77222</xc:pobox>
            <xc:ext/>
            <xc:street>Avenue de la Toison d'Or, 79 - 3rd Floor</xc:street>
            <xc:locality>Brussels</xc:locality>
            <xc:region/>
            <xc:code>1060</xc:code>
            <xc:country>Belgium</xc:country>
         </xc:adr>
         <xc:email>
            <xc:text>support@pemea.org</xc:text>
         </xc:email>
         <xc:url>
            <xc:uri>http://www.pemea.help</xc:uri>
         </xc:url>
      </xc:vcard>
   </DataProviderContact>
</EmergencyCallData.ProviderInfo>
```

13.5 EmergencyCallData.DeviceInfo

Inclusion of this element is optional.

Much of the information included in this element is unknown to the Application Provider and it need only be provided if the App provides it at call time. See IETF RFC 7852 [11] for examples.

Element	Parameter	PEMEA Presence	Type	Description
EmergencyCallData.DeviceInfo	DataProviderReference	Mandatory	xs:token	This information is provided by the AP and shall be the same for all DataProviderReference elements in all additional data blocks in the same emergency DataSend message.
	DeviceClassification	Mandatory	xs:string	This shall be set to "smart-phone-app".
	DeviceManufacturer	Optional	xs:string	This should be provided if available.
	DeviceModelNr	Optional	xs:string	This should be provided if available.
	UniqueDeviceID	Mandatory	xs:string	Two of these shall be provided. See table below.
	DeviceSpecificType	Conditional	xs:string	The information in this field represents the name of the device application being used and the version number of the application.

Element	Parameter	Value	Description
UniqueDeviceID	TypeOfDeviceID	"IMSI"	This is an attribute on the UniqueDeviceID element and specifies how to interpret the value.
	Element value	--	The value of the element shall be the IMSI of the device.
UniqueDeviceID	TypeOfDeviceID	"IMEI"	This is an attribute on the UniqueDeviceID element and specifies how to interpret the value.
	Element value	--	The value of the element shall be the IMEI of the device.

13.6 EmergencyCallData.SubscriberData

13.6.1 EmergencyCallData.SubscriberData profile

Inclusion of this element is mandatory. Future extension to the protocol usage may move its inclusion to conditional based on EDS origination.

This clause defines the elements that make up the EmergencyCallData.SubscriberData structure for use in PEMEA. A full example of an EmergencyCallData.SubscriberData element compliant with the present document is provided in clause 13.6.3.

Element	Parameter	PEMEA Presence	Type	Description
EmergencyCallData.SubscriberData	privacyRequested	Mandatory	xs:boolean	This is an attribute of the EmergencyCallData.SubscriberData element. Normally set to "false", if set to "true" adherence is determined by the destination jurisdiction not the originating jurisdiction.
	DataProviderReference	Mandatory	xs:token	This information is provided by the AP and may be any unique token.
	SubscriberData	Mandatory	vcard	This supplies more information about the caller. Only one vcard is allowed SubscriberData in PEMEA.

13.6.2 SubscriberData :- vcard

13.6.2.1 SubscriberData :- vcard profile

The vcard is used extensively in the additional data specification. This is a very flexible and potentially unstructured information container. This table attempts to add structure to the caller information provided in the SubscriberData vcard. The information in this table shall be provided, other fields may be provided.

Element	Field being represented	PEMEA Presence	vcard element	Description
vcard	Full name of caller	Mandatory	n	Contains the full name of the caller.
	Home address	Conditional	adr	The home address of the caller if available. Not all people are of fixed abode, so this data may not be provided.
	Language	Mandatory	lang	Contains the languages that the caller can speak. This should include all verbal and non-verbal languages (for example sign language).
	Gender	Optional	gender	Gender of the caller.
	Age	Optional	bday	Birthday of the caller allowing age of caller to be determined.
	Other Contacts	Optional		Defines other ways that the caller may be contacted.
	Telephone numbers	Optional	tel	Additional telephone numbers associated with the caller.
	Email Address	Optional	email	An email address for the caller.
	Emergency Family Contacts	Conditional	related	Next of kin or family member to contact if required. This information is sent if the application user has provided it.

13.6.2.2 SubscriberData :- Caller's name

Element	Element	PEMEA Presence	Type	Description
n	parameter	Mandatory	element	Used to specify the language that the name is written in
	surname	Mandatory	xs:string	Last name of the caller
	given	Mandatory	xs:string	Given/first name of the caller
	additional	Optional	xs:string	Any other names the caller may have if available
	prefix	Mandatory	xs:string	Prefix salutation of the caller e.g. Mr, Ms, Dr
	suffix	optional	xs:string	Any name suffixes that the caller may have
NOTE: The contents for this parameter are optional in PEMEA, however, the vCard schema requires the presence of this element even if it is empty.				

Element	Element	PEMEA Presence	Value	Description
parameters	language	Mandatory	--	
	language-tag	Mandatory	Any abbreviation from [13].	PEMEA requires the language that the caller's name is specified in. This shall be an abbreviation and the allowable abbreviations are specified in [13].

```
<vcard xmlns="urn:ietf:params:xml:ns:vcard-4.0">
   <n>
      <parameters>
         <language>
            <language-tag>en</language-tag>
         </language>
      </parameters>
      <surname>Smith</surname>
      <given>George</given>
      <additional>Lawrence</additional>
      <prefix>Mr</prefix>
      <suffix/>
   </n>
```

```
</vcard>
```

13.6.2.3 SubscriberData :- home address

Element	Element	PEMEA Presence	Type	Description
adr	parameters	Mandatory	element	Block defining parameters that describe the type of address.
	pobox	Optional	xs:string	Use is not recommended. See note.
	ext	Optional	xs:string	Not used. See note.
	street	Mandatory	xs:string	Street address, street name, number, suite and floor if applicable.
	locality	Mandatory	xs:string	Municipality, city or suburb. May be empty if not applicable.
	region	Mandatory	xs:string	State or Province. May be empty if not applicable.
	code	Mandatory	xs:string	The post code the caller.
	country	Mandatory	xs:string	Country name. Shall be provided.
NOTE: The contents for this parameter are optional in PEMEA, however, the vCard schema requires the presence of this element even if it is empty.				

Element	Element	PEMEA Presence	Value	Description
parameters	language	Mandatory	--	
	language-tag	Mandatory	Any abbreviation from [13]	PEMEA requires the language that the address is specified in. This shall be an abbreviation and the allowable abbreviations are specified in [13].
	type	Mandatory	--	Defines the type of element that contains the information that describes the address.
	text	Mandatory	home	PEMEA only allows a value of "home" to be specified in this field when relating to SubscriberData.

```
<vcard xmlns="urn:ietf:params:xml:ns:vcard-4.0">
   <adr>
      <parameters>
         <language>
            <language-tag>fr</language-tag>
         </language>
         <type><text>home</text></type>
      </parameters>
      <pobox/>
      <ext/>
      <street>Avenue de la Toison d'Or, 79 - 3rd Floor</street>
      <locality>Brussels</locality>
      <region/>
      <code>1060</code>
      <country>Belgium</country>
   </adr>
</vcard>
```

13.6.2.4 SubscriberData :- language

There shall be at least one of these elements but there may be more than one.

Element	Element	PEMEA Presence	Type	Description
lang	parameters	Mandatory	element	Used to assist in defining the order of preference if the caller can speak more than one language.
	Language-tag	Mandatory	xs:string	Name of the language used by the caller, this may be spoken or unspoken. List of the allowable abbreviations are contained in [13] or [14].

Element	Element	PEMEA Presence	Value	Description
parameters	pref	Mandatory	--	Defines the parameter is specifying a preference.
	integer	Mandatory	1 - 20	Specifies the preference towards the specific language. The smaller the number the higher the preference.

```
<vcard xmlns="urn:ietf:params:xml:ns:vcard-4.0">
   <lang>
      <parameters>
         <pref><integer>1</integer></pref>
      </parameters>
      <language-tag>en</language-tag>
   </lang>
   <lang>
      <parameters>
         <pref><integer>2</integer></pref>
      </parameters>
      <language-tag>fr</language-tag>
   </lang>
</vcard>
```

13.6.2.5 SubscriberData :- gender

Element	Element	PEMEA Presence	Type	Description
gender	sex	Mandatory	xs:string	Indicates the gender of the caller. M, F, O, N, U

```
<vcard xmlns="urn:ietf:params:xml:ns:vcard-4.0">
   <gender><sex>M</sex></gender>
</vcard>
```

13.6.2.6 SubscriberData :- bday

Element	Element	PEMEA Presence	Value	Description
bday	date	Mandatory	YYYYMMDD	If a value is provided it shall be provided in the form YYYYMMDD. The purpose of the value is to prove the age of the caller.

```
<vcard xmlns="urn:ietf:params:xml:ns:vcard-4.0">
   <bday><date>19670722</date></bday>
</vcard>
```

13.6.2.7 SubscriberData :- tel

Inclusion of this element is optional but recommended.

This clause defines how to specify additional telephone contacts for the caller. These may be repeats of the contacts provided in the callerIds specified in clause 10.3.10, however, the numbers in the tel fields are designed for display to the PSAP, while those in the callerId fields are designed as keys through which the PSAP can access this additional data.

Element	Element	PEMEA Presence	Value	Description
tel	parameters	Mandatory	--	Contains the type of telephony device.
	text	Mandatory	anyURI	If this is a telephone number then it shall be expressed as a tel URI [5]. Text and video types may be expressed using other URI forms.

Element	Element	PEMEA Presence	Value	Description
parameters	pref	Optional	-	Specifies a preference in terms of numbers to try and contact the caller on. Note this may not specify the calling number as the first choice.
	type	Mandatory	--	Defines the parameter is specifying a preference.

Element	Element	PEMEA Presence	Value	Description
pref	integer	Conditional	1 - 100	This field shall contain a value if the pref element is present. It specifies the preference in terms on numbers to call to reach the caller, with 1 being the most preferred option.
type	text	Mandatory	work home cell video text	Only the types listed in the value column are allowed in PEMEA. Multiple entries for each type are allowed.

```
<vcard xmlns="urn:ietf:params:xml:ns:vcard-4.0">
   <tel>
      <parameters>
         <pref><integer>1</integer></pref>
         <type><text>home</text></type>
      </parameters>
      <text>tel:+32-2534-9789</text>
   </tel>
   <tel>
      <parameters>
         <pref><integer>2</integer></pref>
         <type><text>cell</text></type>
      </parameters>
      <text>tel:+32-4352-9789</text>
   </tel>
   <tel>
      <parameters>
         <pref><integer>3</integer></pref>
         <type><text>cell</text></type>
      </parameters>
      <text>tel:+44-3425-9789</text>
   </tel>
</vcard>
```

13.6.2.8 SubscriberData :- email

Inclusion of this element is optional.

This element provides email contacts for the caller. No preference or types are specified for the email address.

Element	Element	PEMEA Presence	Type	Description
email	text	Conditional	xs:string	Specifying an email is completely optional. If an email element exists, then the text element with a correctly formed email address shall also be present.

```
<vcard xmlns="urn:ietf:params:xml:ns:vcard-4.0">
   <email><text>jw@pemea.org</text></email>
</vcard>
```

13.6.2.9 SubscriberData :- Emergency Family Contacts

Inclusion of this element is optional.

The vCard does not explicitly provide fields for this information. To support the representation of this information the vCard *related* construct is used. This type has a set of tokens explicitly defined in the vCard schema, however, only the types in Table 13 are valid within the scope of PEMEA. This will ensure that any receiving entity can parse the data and render it to the PSAP staff and first responders.

The vCard schema restricts the related element so that only one contact means can be provided per instance. This means that if the caller wishes to specify home, work and mobile numbers for his/her wife/husband/partner, then a new related structure needs to be added for each.

Element	Element	PEMEA Presence	Type	Description
related	parameters	Mandatory	element	The parameter element holds the information that relates the caller to the contact, this aspect is mandatory for PEMEA. It may in addition optional include a relationship between the contact and the contact details, e.g. home or work.
	uri	Mandatory	element	This is the means of contacting the emergency contact person. Only one contact mechanism is allowed per related element in the vCard schema.

Element	Element	PEMEA Presence	Value	Description
parameters	type	Mandatory	--	Contains the text element that indicates the relationship between the caller and the contact. In addition it may optional also contain the relationship between the contact and the contact details.
	text	Mandatory	See Table 13	The relationship between the caller and the contact shall be present.
	text	Optional	See Table 14	This element is optional. If present, it contains the relationship between the contact and the contact details.

The entries in Table 13 contains the allowed the values for the description of the relationship between the caller and contact. This list is a reduced set of the defined tokens in the vCard schema that is not extensible. The descriptions are representative examples only and not expected to be exhaustive. Values other than those provided in Table 13 are not valid in PEMEA and should be ignored by a receiving entity.

Table 13: Emergency contact relationship to caller

Value	Description
spouse	Wife, husband or partner
parent	Father, mother, step-parents, grand-parents, adoptive and foster parents
child	Son, daughter, adopted child, step-child, grand-child
sibling	Brother, sister, step-sibling, half-sibling
kin	Uncle, aunt, cousin
emergency	Caller does not specify the relationship this is just who to call in an emergency
neighbor	Lives close to the caller
friend	A friend of the caller
co-resident	Shares accommodation with the caller, flat-mate, house-mate
co-worker	Works with the caller

The entries in Table 14 contain the allowed values used to express the relationship between the contact and the contact details. Only the values in Table 14 are considered valid and any other values should be ignored by a receiving entity.

Table 14: Relationship between contact and contact details

Value	Description
home	Details provided are for reaching the contact at home
work	Details provided are for reaching the contact at work
contact	Details provided are for reaching the contact in general, for example mobile phone

```
<vcard xmlns="urn:ietf:params:xml:ns:vcard-4.0">
   <related>
      <parameters>
         <type>
            <text>spouse</text>
            <text>home</text>
         </type>
      </parameters>
      <uri>tel:+32-2534-9789</uri>
   </related>
   <related>
      <parameters>
         <type>
            <text>spouse</text>
            <text>contact</text>
         </type>
      </parameters>
      <uri>tel:+32-7777-9789</uri>
   </related>
</vcard>
```

13.6.3 EmergencyCallData.SubscriberData :- Complete Example

This clause provides an example of a complete PEMEA compliant EmergencyCallData.SubscriberData structure.

```
<EmergencyCallData.SubscriberInfo
   xmlns="urn:ietf:params:xml:ns:EmergencyCallData:SubscriberInfo"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:xc="urn:ietf:params:xml:ns:vcard-4.0"
   xsi:schemaLocation="urn:ietf:params:xml:ns:EmergencyCallData:SubscriberInfo SubscriberInfo.xsd"
   privacyRequested="true">
   <DataProviderReference>CoolAP-0xFF4568262458</DataProviderReference>
   <SubscriberData>
      <xc:vcard>
          <xc:n>
            <xc:parameters>
              <xc:language>
                <xc:language-tag>en</xc:language-tag>
              </xc:language>
            </xc:parameters>
            <xc:surname>Smith</xc:surname>
            <xc:given>George</xc:given>
            <xc:additional>Lawrence</xc:additional>
            <xc:prefix>Mr</xc:prefix>
            <xc:suffix/>
         </xc:n>
         <xc:adr>
            <xc:parameters>
               <xc:language>
                  <xc:language-tag>fr</xc:language-tag>
               </xc:language>
            </xc:parameters>
            <xc:pobox>77222</xc:pobox>
            <xc:ext/>
            <xc:street>Avenue de la Toison d'Or, 79 - 3rd Floor</xc:street>
            <xc:locality>Brussels</xc:locality>
            <xc:region/>
            <xc:code>1060</xc:code>
            <xc:country>Belgium</xc:country>
         </xc:adr>
         <xc:lang>
            <xc:parameters>
               <xc:pref><xc:integer>1</xc:integer></xc:pref>
             </xc:parameters>
             <xc:language-tag>en</xc:language-tag>
         </xc:lang>
         <xc:lang>
```

```
            <xc:parameters>
               <xc:pref><xc:integer>2</xc:integer></xc:pref>
            </xc:parameters>
            <xc:language-tag>fr</xc:language-tag>
         </xc:lang>
         <xc:gender><xc:sex>M</xc:sex></xc:gender>
         <xc:tel>
            <xc:parameters>
               <xc:pref><xc:integer>1</xc:integer></xc:pref>
               <xc:type><xc:text>home</xc:text></xc:type>
            </xc:parameters>
            <xc:text>tel:+32-2534-9789</xc:text>
         </xc:tel>
         <xc:tel>
            <xc:parameters>
               <xc:pref><xc:integer>2</xc:integer></xc:pref>
               <xc:type><xc:text>cell</xc:text></xc:type>
            </xc:parameters>
            <xc:text>tel:+32-4352-9789</xc:text>
         </xc:tel>
         <xc:tel>
            <xc:parameters>
               <xc:pref><xc:integer>3</xc:integer></xc:pref>
               <xc:type><xc:text>cell</xc:text></xc:type>
            </xc:parameters>
            <xc:text>tel:+44-3425-9789</xc:text>
         </xc:tel>
         <xc:email>
            <xc:text>support@pemea.org</xc:text>
         </xc:email>
         <xc:related>
            <xc:parameters>
               <xc:type>
                  <xc:text>spouse</xc:text>
                  <xc:text>home</xc:text>
               </xc:type>
            </xc:parameters>
            <xc:uri>tel:+32-2534-9789</xc:uri>
         </xc:related>
         <xc:related>
            <xc:parameters>
               <xc:type>
                  <xc:text>spouse</xc:text>
                  <xc:text>contact</xc:text>
               </xc:type>
            </xc:parameters>
            <xc:uri>tel:+32-7777-9789</xc:uri>
        </xc:related>
      </xc:vcard>
   </SubscriberData>
</EmergencyCallData.SubscriberInfo>
```

13.7 Additional-Data :- EmergencyCallDataReference

The only valid additional-data element that shall be sent by reference is for caller information, contained in an EmergencyCallData.SubscriberInfo.

Element	Parameter	PEMEA Presence	Type	Description
EmergencyCallData Reference	purpose	Mandatory	xs:token	This value shall be set to "EmergencyCallData.SubscriberInfo".
	ref	Mandatory	xs:anyURI	The URI shall use the https scheme. The URI shall be constructed so as not to expose the "identity" of the caller and the URI construction techniques described in clause 3.4 of IETF RFC 5808 [16] apply. The authentication procedures outlined in clause 9.2 shall be followed.

```
<EmergencyCallDataReference xmlns="urn:ietf:params:xml:ns:EmergencyCallData"
                            purpose="EmergencyCallData.SubscriberInfo"
                            ref="https://coolAP.pemea.org:7865/duheuh38x894nxe3iu3iu"/>
```

The PSAP/PSP shall use an HTTP GET to retrieve the EmergencyCallData.SubscriberInfo data block. The MIME type in the content header field of the request shall be set to "application/EmergencyCallData.SubscriberInfo+xml".

The EmergencyCallData.SubscriberInfo data structure is described in detail in clause 13.6.

13.8 provided-by : Complete Examples

The provided-by element is defined inside the PIDF-LO schema, it is therefore necessary to show the examples inside the geopriv structure in order to provide an example that validates.

```
<geopriv xmlns="urn:ietf:params:xml:ns:pidf:geopriv10"
         xmlns:ecd="urn:ietf:params:xml:ns:EmergencyCallData"
         xmlns:pi="urn:ietf:params:xml:ns:EmergencyCallData:ProviderInfo"
         xmlns:xc="urn:ietf:params:xml:ns:vcard-4.0">
   <location-info/>
   <usage-rules/>
   <provided-by>
      <ecd:EmergencyCallDataValue>
         <pi:EmergencyCallData.ProviderInfo>
            <pi:DataProviderReference>xhjjshjsdhjsdh</pi:DataProviderReference>
            <pi:DataProviderString>Cool Application Provider</pi:DataProviderString>
            <pi:ProviderID>urn:pemea:pemea:ap:ID0x123FEDAC</pi:ProviderID>
            <pi:ProviderIDSeries>PEMEA</pi:ProviderIDSeries>
            <pi:TypeOfProvider>Application Provider</pi:TypeOfProvider>
            <pi:ContactURI>tel:+32-2534-9789</pi:ContactURI>
            <pi:Language>fr</pi:Language>
            <pi:DataProviderContact>
              <xc:vcard>
                <xc:org>
                  <xc:parameters>
                    <xc:language>
                      <xc:language-tag>en</xc:language-tag>
                    </xc:language>
                 </xc:parameters>
                 <xc:text>Really Application Provider</xc:text>
               </xc:org>
               <xc:adr>
                 <xc:parameters>
                   <xc:language>
                     <xc:language-tag>fr</xc:language-tag>
                   </xc:language>
                 </xc:parameters>
                 <xc:pobox>77222</xc:pobox>
                 <xc:ext/>
                 <xc:street>Avenue de la Toison d'Or, 79 - 3rd Floor</xc:street>
                 <xc:locality>Brussels</xc:locality>
                 <xc:region/>
                 <xc:code>1060</xc:code>
                 <xc:country>Belgium</xc:country>
               </xc:adr>
               <xc:email>
                  <xc:text>support@pemea.org</xc:text>
               </xc:email>
               <xc:url>
                 <xc:uri>http://www.pemea.help</xc:uri>
               </xc:url>
                </xc:vcard>
              </pi:DataProviderContact>
           </pi:EmergencyCallData.ProviderInfo>
        </ecd:EmergencyCallDataValue>
     <ecd:EmergencyCallDataReference
                 purpose="EmergencyCallData.SubscriberInfo"
                 ref="https://coolAP.pemea.org:7865/duheuh38x894nxe3iu3iu"/>
   </provided-by>
</geopriv>
```

14 Operating Procedures

14.1 Application Provider Operating Procedures

14.1.1 AP sending an EDS to the PSP

This clause defines the procedures for sending an emergencyDataSend message to a PSP.

- Generate a unique message sequence (msgSeq) for the EDS.
- Set the ttl value:
    - The value shall be set to 1 if the caller has indicated to the AP that they do not wish to roam beyond their home PSP area.
    - The value shall be set to a minimum of 3 if the caller has indicated to the AP that they do wish be able to roam beyond their home region.
    - A value of 5 is recommended, as this allows for 2 ASPs in the message path.
    - Shall not exceed a value of 10.
- Create a URI for:
    - onErrorPost so that the AP is notified of any delivery errors (refer to clause 11.1.3).
    - Retrieval of SubscriberInfo (if ttl > 1).
    - For each <information> element reach-back URI type supported by the AP and by the App.
    - onCapSupportPost so that the AP is notified of EDS delivery to a terminating entity, and which <information> capabilities the terminating entity supports (refer to clause 11.1.4).
- Create a route:
    - Add the message msgSeq.
    - Add a <hop> to the <hops>:
        - Set position to zero.
        - Set timeStamp to the current time.
        - Set the node to the URL of the AP.
- Create the callerIDs element based on the caller-ids registered by the user.
- Create the PIDF-LO:
    - Set the entity to the first callerId URI type.
    - Convert the location provided by the App to a GeoShape.
    - Add the method if available.
    - Add a provided-by element:
        - Create an EmergencyDataValue element:
            - Create a ProviderInfo element and add it to the EmergencyDataValue element.
            - If ttl = 1, create a SubscriberInfo element and add it to the EmergencyDataValue element.
            - If ttl >1, create an EmergencyCallReference element and add the SubscriberInfo URI to it.

  - Set the timeStamp to the current time.

- Create an ApMoreInformation element containing all of the information URI determined previously.
- Create an EDS:
  - Set the timeStamp to the current time.
  - Set the ttl value.
  - Set the onErrorPost URI if used.
  - Set the onCapSupportPost URI if used.
  - Add the route element.
  - Add the callerIDs.
  - Add the apMoreInformation.
  - Add the PIDF-LO.
- Log the key components of the EDS.
- Send the EDS to the PSP.
- Log the response from the PSP.

14.1.2 AP reach-back URI queries

This condition occurs when the AP has included an onCapSupportPost URI and an apMoreInformation element in the EDS. The PSAP uses the provided reach-back URIs to request further information about the call/caller from the AP:

- The PSAP or tPSP receives the EDS message containing an onCapSupportPost URI and posts to this URI which of the proffered capabilities it supports (this may be an empty post refer to clause 11.1.4).
- If the post to the onCapSupportPost URI contains an apMoreInformation element then the AP:
  - Receives an apMoreInformation element from the terminating entity via the URI contained in the onCapSupportPost attribute of the EDS.
  - Has a list of PSAPs and PSPs and their associated domains (this is an operational procedure).
  - Has core CA root certificates contained in a local trust store.
  - Is listening on a port assigned to one of the URIs sent in the apMoreInformation element in the EDS message (refer to clause 10.3.12).
  - Receives client-certificate from the PSAP and validates it against a root CA certificate from the local trust store:
    - If the validation fails then the request is denied with an HTTP 403 "Forbidden" response. The source of the request as well as the URI used is logged.
    - If the requesting entity validates against a root CA certificate, then the domain is checked against the PSAP or PSP list described above.
    - If the domain of the requesting entity is not in the list then the request is denied with an HTTP 403 "Forbidden" response. The source of the request as well as the URI used is logged.
  - If all authentication and validation succeed then specific protocol exchange procedures are invoked for the feature being requested.

14.1.3 Call termination (ending) and URI invalidation

- AP shall invalidate URIs associated with caller data after a fixed period of time from a specific time X:
    - Unless the App explicitly notifies the AP that the call is still in progress:
        - Location updates from the App to the AP are one example of how this may be achieved.
    - The time period shall not exceed more than 1 hour from time X.
- The initial time X is set when the AP receives notification from the App that a call has been initiated.
- Time X is set to the current time each time the App notifies that AP that the call is still in progress.
- Start time + time X shall not exceed 24 hours.
- Once the duration has expired, any attempt to access a reach-back URI by a requesting entity shall result in an HTTP 404 "Not Found" response being returned.

14.2 PSAP Service Provider Operating Procedures

14.2.1 PSP receiving an EDS message over Ps

This clause provides the PSP procedures when receiving an emergencyDataSend message from an AP:

- The AP is authenticated:
    - If the AP fails authentication then an HTTP 403 "Forbidden" response is returned.
- If the *ttl* value is less than one then an error is returned to the AP with a reason code of *ttlExhausted.*
- If the *ttl* value is set to one and the PSP determines that routing the EDS requires the use of an ASP, then the PSP shall return an error to the AP with a reason code of ttlExhausted and indicate a message of "ASP required" in the preferred language of the PSP.
- The PSP uses its internal routing functions to determine the destination or next hop of the EDS message.
- If the PSP determines that the final destination is for a PSAP that it directly serves then:
    - The PSP shall send an EDR message to the AP and:
        - Set the timestamp to be the time that the message is constructed for sending to the AP.
        - Add a <hop> to the <hops> element in the <route> element.
        - The new hop shall have position="1".
        - The new hop shall have a value of the URI of the PSP.
        - The destination element shall have a destType="PSAP" and the value shall be a URI identifying the PSAP is one is available, otherwise a name describing the PSAP.
        - Return the EDR to the AP and terminate the session.
    - The PSP shall make the data available to the PSAP.
    - If the EDS contains an onCapSupportPost URI and the PSAP cannot support initiating a post to the onCapSupportPost URI then the PSAP may perform this task on behalf of the PSAP. The PSP shall:
        - If no apMoreInformation element is present in the EDS then the PSP shall post an empty message to the onCapSupportPost URI.

        - If an apMoreInformation element is present in the EDS then the PSP shall:
            - Determine which capabilities the PSAP does support (how the PSP knows which capabilities the PSAP supports is out of scope of the present document) of the proffered AP capabilities.
            - Construct an apMoreInformation element containing all of the capabilities common to the AP and the PSAP.
            - Post the apMoreInformation element to the URI contained in the onCapSupportPost URI.

- If the PSP determines that the final destination is not a PSAP that it directly serves then:
    - If the PSP does not have a relationship with a routing node it shall return an error to the AP:
        - Set the timestamp to the time that the message is constructed for sending to the AP.
        - Set the reason="noRoute".
        - Set the message if desirable.
        - Add a <hop> to the <hops> element in the <route> element.
        - The new hop shall have a timeStamp value set to the time that the hop is added.
        - The new hop shall have position="1".
        - The new hop shall have a value of the URI of the PSP.
        - Return the error to the AP and terminate the session.
    - If the PSP does have a relationship with a routing node it:
        - Sends an EDR to the AP:
            - Set the timestamp to be the time that the message is constructed for sending to the AP.
            - Add a <hop> to the <hops> element in the <route> element.
            - The new hop shall have a timeStamp value set to the time that the hop is added.
            - The new hop shall have position="1".
            - The new hop shall have a value of the URI of the PSP.
            - The destination element shall have a destType="ASP" and the value shall be a URI identifying the ASP to which the EDS is being sent.
            - Return the EDR to the AP and terminate the session with the AP.

14.2.2 PSP sending an EDS message over Pr

The PSP has received an EDS from an AP:

- Send an EDS to an ASP:
    - Decrement the ttl value in the EDS received from the AP.
    - Add a <hop> to the <hops> element in the <route> element.
    - The new hop shall have a timeStamp value set to the time that the hop is added.
    - The new hop shall have position="1".
    - The new hop shall have a value of the URI of the PSP.
    - The modified ttl and route elements replace the ttl and route fields in the EDS received from the AP and this message is sent to the ASP.

  - The PSP logs the response from the ASP.

- If the PSP receives an error message from the ASP and the EDS message contains an "onErrorPost" URI attribute then the PSP shall post the error message received from the ASP to the URI specified by the onErrorPost attribute (refer to clause 11.1.3).

14.2.3 PSP receiving an EDS message over Pr

This clause provides the PSP procedures when receiving an emergencyDataSend message from an ASP:

- The ASP is authenticated.
- If the ASP fails authentication then an HTTP 403 "Forbidden" response is returned.
- If the *ttl* value is less than one then an error is returned to the ASP with a reason code of *ttlExhausted.*
- If the *ttl* value is set to one and the PSP determines that routing the EDS requires the use of another ASP, then the PSP shall return an error to the ASP with a reason code of ttlExhausted and indicate a message of "ASP required" in the preferred language of the PSP.
- If the PSP determines that the final destination is for a PSAP that it directly serves then:
  - The PSP shall send an EDR message to the ASP and:
    - Set the timestamp to be the time that the message is constructed for sending to the ASP.
    - Add a <hop> to the <hops> element in the <route> element.
    - The new hop position of the next number in the sequence.
    - The new hop shall have a value of the URI of the PSP.
    - The destination element shall have a destType="PSAP" and the value shall be a URI identifying the PSAP is one is available, otherwise a name describing the PSAP.
    - Return the EDR to the ASP and terminate the session.
  - The PSP shall make the data available to the PSAP.
  - If the EDS contains an onCapSupportPost URI and the PSAP cannot support initiating a post to the onCapSupportPost URI then the PSAP may perform this task on behalf of the PSAP. The PSP shall:
    - If no apMoreInformation element is present in the EDS then the PSP shall post an empty message to the onCapSupportPost URI.
    - If an apMoreInformation element is present in the EDS then the PSP shall:
      - Determine which capabilities the PSAP does support (how the PSP knows which capabilities the PSAP supports is out of scope of the present document) of the proffered AP capabilities.
      - Construct an apMoreInformation element containing all of the capabilities common to the AP and the PSAP.
    - Post the apMoreInformation element to the URI contained in the onCapSupportPost URI.
- If the PSP determines that the final destination is not a PSAP that it directly serves then:
  - If the PSP does not have a relationship with a routing node then it shall return an error to the ASP:
    - Set the timestamp to the time that the message is constructed for sending to the ASP.
    - Set the reason="noRoute".
    - Set the message if desirable.
    - Add a <hop> to the <hops> element in the <route> element.

        - ▪ The new hop shall have a timeStamp value set to the time that the hop is added.
        - ▪ The new hop position of the next number in the sequence.
        - ▪ The new hop shall have a value of the URI of the PSP.
        - ▪ Return the error to the ASP and terminate the session.
    - If the PSP does have a relationship with a routing node then it will:
        - ▪ Send an EDR to the ASP:
            - Set the timestamp to be the time that the message is constructed for sending to the AP.
            - Add a <hop> to the <hops> element in the <route> element.
            - The new hop shall have a timeStamp value set to the time that the hop is added.
            - The new hop position of the next number in the sequence.
            - The new hop shall have a value of the URI of the PSP.
            - The destination element shall have a destType="ASP" and the value shall be a URI identifying the ASP to which the EDS is being sent.
- Return the EDR to the ASP and terminate the session with the ASP.

14.3 Aggregating Service Provider Operating Procedures

14.3.1 Overview of Pr

The aggregating service provider has two distinct interfaces, a receiving Pr interface on which it receives EDS messages from an oPSP or ASP, and a sending Pr interface on which it sends EDS messages to a tPSP or subsequent ASP.

14.3.2 ASP receiving an EDS message over Pr

This clause the procedures for an ASP when receiving an emergencyDataSend message over the Pr interface:

- If the *ttl* value is less than two then an error is returned to the sender with a reason code of *ttlExhausted,* a message may be included in the preferred language of the ASP.
- If the ASP does not have a relationship with the tPSP or a routing node to get there then it shall return an error to the oPSP:
    - Sets the timestamp to the time that the message is constructed for sending to the oPSP.
    - Sets the reason="noRoute".
    - Sets the message if desirable.
    - Adds a <hop> to the <hops> element in the <route> element.
    - The new hop shall have a timeStamp value set to the time that the hop is added.
    - The new hop position of the next number in the sequence.
    - The new hop shall have a value of the URI of the ASP.
    - Returns the error to the oPSP and terminate the session.
- If the ASP does have a relationship with the tPSP or a routing node to get there then it:
    - Sends an EDR to the oPSP.
    - Sets the timestamp to be the time that the message is constructed for sending to the oPSP.

  - Adds a <hop> to the <hops> element in the <route> element.
  - The new hop shall have a timeStamp value set to the time that the hop is added.
  - The new hop position of the next number in the sequence.
  - The new hop shall have a value of the URI of the ASP.
  - The destination element shall have a destType="tPSP" or "ASP" depending on the type of the destination and the value shall be a URI identifying the tPSP/ASP to which the EDS is being sent.
  - Returns the EDR to the oPSP and terminate the session with the oPSP.

14.3.3 ASP sending an EDS message over Pr

The ASP has received an EDS from an oPSP or ASP:

- Sends an EDS to a tPSP or ASP:
  - Decrement the ttl value in the EDS received from the oPSP.
  - Add a <hop> to the <hops> element in the <route> element.
  - The new hop shall have a timeStamp value set to the time that the hop is added.
  - The new hop position of the next number in the sequence.
  - The new hop shall have a value of the URI of the ASP.
  - The modified ttl and route elements replace the ttl and route fields in the EDS received from the oPSP and the message is sent to the tPSP/ASP.
  - The ASP logs the response from the tPSP/ASP.
- If the ASP receives an error message from the tPSP and the EDS message contains an "onErrorPost" URI attribute then the ASP shall post the error message received from the tPSP to the URI specified by the onErrorPost attribute.

15 Example message Flows

15.1 Description

The following flows provide the messages from an AP to the oPSP and from the oPSP to an ASP and from the ASP to a tPSP.

15.2 AP to PSP EDS

```
<emergencyDataSend xmlns="urn:pemea:apps:xml:ns:pemea:base"
                   xmlns:pdf="urn:ietf:params:xml:ns:pidf"
            xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
                   xmlns:gml="http://www.opengis.net/gml"
                   xmlns:gs="http://www.opengis.net/pidflo/1.0"
                   xmlns:con="urn:ietf:params:xml:ns:geopriv:conf"
                   xmlns:pi="urn:ietf:params:xml:ns:EmergencyCallData:ProviderInfo"
                   xmlns:ecd="urn:ietf:params:xml:ns:EmergencyCallData"
                   xmlns:xc="urn:ietf:params:xml:ns:vcard-4.0"
                   xmlns:cell="urn:ietf:params:xml:ns:geopriv:lm:cell"
                   ttl="5"
                   onErrorPost="https://cooAP.example.com.be:2001/pemea/error/CoolAP-7496"
                   onCapSupporPost="https://cooAP.example.com.be:2001/pemea/cap/CoolAP-7496">
  <route msgSeq="CoolAP-7496" >
    <hops>
      <hop position="0" timeStamp="2016-02-02T18:14:001Z">
        <node>https://cooAP.example.com.be:2001/pemea/</node>
      </hop>
    </hops>
  </route>
  <callerIds>
    <callerId typeOfId="msisdn">tel:+44-555-555-1234</callerId>
    <callerId typeOfId="msisdn">tel:+34-555-222-6789</callerId>
    <callerId typeOfId="skypeName">winterb</callerId>
  </callerIds>
  <apMoreInformation>
    <information typeOfInfo="Location_Update" protocol="HELD">
                 https://coolap.example.com.be:2096/Web?id=CoolAP-7496
    </information>
  </apMoreInformation>
  <accessData>
    <cell:network>
      <cell:mcc>253</cell:mcc>
      <cell:mnc>002</cell:mnc>
    </cell:network>
  </accessData>
  <pdf:presence entity="tel:+44-555-555-1234">
    <pdf:tuple id="circle">
      <pdf:status>
        <gp:geopriv>
          <gp:location-info>
            <gs:Circle srsName="urn:ogc:def:crs:EPSG::4326">
              <gml:pos>42.5463 -73.2512</gml:pos>
              <gs:radius uom="urn:ogc:def:uom:EPSG::9001">
                   30.0
              </gs:radius>
            </gs:Circle>
            <con:confidence pdf="normal">95</con:confidence>
          </gp:location-info>
          <gp:usage-rules/>
          <gp:method>GNSS</gp:method>
          <gp:provided-by>
            <ecd:EmergencyCallDataValue>
              <pi:EmergencyCallData.ProviderInfo>
                <pi:DataProviderReference>xhjjshjsdhjsdh</pi:DataProviderReference>
                <pi:DataProviderString>Cool Application Provider</pi:DataProviderString>
         <pi:ProviderID>urn:pemea:pemea:ap:ID0x123FEDAC</pi:ProviderID>
         <pi:ProviderIDSeries>PEMEA</pi:ProviderIDSeries>
         <pi:TypeOfProvider>Application Provider</pi:TypeOfProvider>
         <pi:ContactURI>tel:+32-2534-9789</pi:ContactURI>
                <pi:Language>fr</pi:Language>
                <pi:DataProviderContact>
                  <xc:vcard>
                    <xc:org>
                      <xc:parameters>
                        <xc:language>
                          <xc:language-tag>en</xc:language-tag>
                        </xc:language>
                      </xc:parameters>
                      <xc:text>Really Application Provider</xc:text>
                    </xc:org>
                    <xc:adr>
                      <xc:parameters>
                         <xc:language>
                          <xc:language-tag>fr</xc:language-tag>
                        </xc:language>
                      </xc:parameters>
                      <xc:pobox>77222</xc:pobox>
                      <xc:ext/>
                      <xc:street>Avenue de la Toison d'Or, 79 - 3rd Floor</xc:street>
                      <xc:locality>Brussels</xc:locality>
                      <xc:region/>
                      <xc:code>1060</xc:code>
                      <xc:country>Belgium</xc:country>
                    </xc:adr>
                    <xc:email>
                      <xc:text>support@pemea.org</xc:text>
                    </xc:email>
                    <xc:url>
                      <xc:uri>http://www.pemea.help</xc:uri>
                    </xc:url>
                  </xc:vcard>
                </pi:DataProviderContact>
              </pi:EmergencyCallData.ProviderInfo>
            </ecd:EmergencyCallDataValue>
```

```
            <ecd:EmergencyCallDataReference purpose="EmergencyCallData.SubscriberInfo"

ref="https://coolAP.pemea.org:7865/duheuh38x894nxe3iu3iu"/>
          </gp:provided-by>
        </gp:geopriv>
      </pdf:status>
    </pdf:tuple>
  </pdf:presence>
</emergencyDataSend>
```

15.3 oPSP to AP EDR

The oPSP determines that it needs the services of an ASP. It adds its address to the route, and marks the destination as being an ASP, along with the address of the ASP.

```
<emergencyDataReceived xmlns="urn:pemea:apps:xml:ns:pemea:base"
                       xmlns:xs="http://www.w3.org/2001/XMLSchema"
                       xmlns:pemea="urn:pemea:apps:xml:ns:pemea:base"
         timeStamp="2016-02-02T18:14:00.521Z">
  <route msgSeq="CoolAP-7496" >
    <hops>
      <hop position="0" timeStamp="2016-02-02T18:14:00.001Z">
         <node>https://cooAP.example.com.be:2001/pemea/</node>
      </hop>
      <hop position="1" timeStamp="2016-02-02T18:14:00.521Z">
        <node>https://orig.psp.example.com:2134/pemea/</node>
      </hop>
    </hops>
  </route>
  <delivery destType="ASP">https://asp.example.com:2195/pemea/</delivery>
</emergencyDataReceived>
```

15.4 oPSP to ASP EDS

oPSP decrements ttl and adds its address to the route element.

```
<emergencyDataSend xmlns="urn:pemea:apps:xml:ns:pemea:base"
                   xmlns:pdf="urn:ietf:params:xml:ns:pidf"
            xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
                   xmlns:gml="http://www.opengis.net/gml"
                   xmlns:gs="http://www.opengis.net/pidflo/1.0"
                   xmlns:con="urn:ietf:params:xml:ns:geopriv:conf"
                   xmlns:pi="urn:ietf:params:xml:ns:EmergencyCallData:ProviderInfo"
                   xmlns:ecd="urn:ietf:params:xml:ns:EmergencyCallData"
                   xmlns:xc="urn:ietf:params:xml:ns:vcard-4.0"
                   xmlns:cell="urn:ietf:params:xml:ns:geopriv:lm:cell"
                   ttl="4"
                   onErrorPost="https://cooAP.example.com.be:2001/pemea/error/CoolAP-7496"
                   onCapSupporPost="https://cooAP.example.com.be:2001/pemea/cap/CoolAP-7496">
  <route msgSeq="CoolAP-7496" >
    <hops>
      <hop position="0" timeStamp="2016-02-02T18:14:001Z ">
        <node>https://cooAP.example.com.be:2001/pemea/</node>
      </hop>
      <hop position="1" timeStamp="2016-01-02T18:14:521Z" >
        <node>https://opsp.example.com.be:2134/pemea/</node>
      </hop>
    </hops>
  </route>
  <callerIds>
    <callerId typeOfId="msisdn">tel:+44-555-555-1234</callerId>
    <callerId typeOfId="msisdn">tel:+34-555-222-6789</callerId>
    <callerId typeOfId="skypeName">winterb</callerId>
  </callerIds>
  <apMoreInformation>
    <information typeOfInfo="Location_Update" protocol="HELD">
          https://coolap.example.com.be:2096/Web?id=CoolAP-7496
    </information>
  </apMoreInformation>
  <accessData>
    <cell:network>
      <cell:mcc>253</cell:mcc>
      <cell:mnc>002</cell:mnc>
    </cell:network>
  </accessData>
  <pdf:presence entity="tel:+44-555-555-1234">
    <pdf:tuple id="circle">
      <pdf:status>
        <gp:geopriv>
          <gp:location-info>
            <gs:Circle srsName="urn:ogc:def:crs:EPSG::4326">
              <gml:pos>42.5463 -73.2512</gml:pos>
              <gs:radius uom="urn:ogc:def:uom:EPSG::9001">
                   30.0
              </gs:radius>
            </gs:Circle>
            <con:confidence pdf="normal">95</con:confidence>
          </gp:location-info>
          <gp:usage-rules/>
          <gp:method>GNSS</gp:method>
          <gp:provided-by>
            <ecd:EmergencyCallDataValue>
              <pi:EmergencyCallData.ProviderInfo>
                <pi:DataProviderReference>xhjjshjsdhjsdh</pi:DataProviderReference>
                <pi:DataProviderString>Cool Application Provider</pi:DataProviderString>
                <pi:ProviderID>urn:pemea:pemea:ap:ID0x123FEDAC</pi:ProviderID>
         <pi:ProviderIDSeries>PEMEA</pi:ProviderIDSeries>
         <pi:TypeOfProvider>Application Provider</pi:TypeOfProvider>
         <pi:ContactURI>tel:+32-2534-9789</pi:ContactURI>
                <pi:Language>fr</pi:Language>
                <pi:DataProviderContact>
                  <xc:vcard>
                    <xc:org>
                      <xc:parameters>
                        <xc:language>
                          <xc:language-tag>en</xc:language-tag>
                        </xc:language>
                      </xc:parameters>
                      <xc:text>Really Application Provider</xc:text>
                    </xc:org>
                    <xc:adr>
                      <xc:parameters>
                        <xc:language>
                          <xc:language-tag>fr</xc:language-tag>
                        </xc:language>
                      </xc:parameters>
                      <xc:pobox>77222</xc:pobox>
                      <xc:ext/>
                      <xc:street>Avenue de la Toison d'Or, 79 - 3rd Floor</xc:street>
                      <xc:locality>Brussels</xc:locality>
                      <xc:region/>
                      <xc:code>1060</xc:code>
                      <xc:country>Belgium</xc:country>
                    </xc:adr>
                    <xc:email>
                      <xc:text>support@pemea.org</xc:text>
                    </xc:email>
                    <xc:url>
                      <xc:uri>http://www.pemea.help</xc:uri>
                    </xc:url>
                  </xc:vcard>
                </pi:DataProviderContact>
              </pi:EmergencyCallData.ProviderInfo>
            </ecd:EmergencyCallDataValue>
            <ecd:EmergencyCallDataReference purpose="EmergencyCallData.SubscriberInfo"
                                        ref="https://coolAP.pemea.org:7865/duheuh38x894nxe3iu3iu"/>
          </gp:provided-by>
        </gp:geopriv>
      </pdf:status>
    </pdf:tuple>
  </pdf:presence>
</emergencyDataSend>
```

15.5 ASP to oPSP EDR

The ASP determines the tPSP. It adds its address to the route, and marks the destination as being a PSP, along with the address of the tPSP.

```
<emergencyDataReceived xmlns="urn:pemea:apps:xml:ns:pemea:base"
                       xmlns:xs="http://www.w3.org/2001/XMLSchema"
                       xmlns:pemea="urn:pemea:apps:xml:ns:pemea:base"
       timeStamp="2016-02-02T18:14:00.980Z">
<route msgSeq="CoolAP-7496" >
  <hops>
    <hop position="0" timeStamp="2016-02-02T18:14:00.001Z">
      <node>https://cooAP.example.com.be:2001/pemea/</node>
    </hop>
    <hop position="1" timeStamp="2016-02-02T18:14:00.521Z">
      <node>https://orig.psp.example.com:2134/pemea/</node>
    </hop>
    <hop position="2" timeStamp="2016-02-02T18:14:00.980Z">
      <node>https://asp.example.com:2195/pemea/</node>
    </hop>
  </hops>
  </route>
  <delivery destType="PSP">https://term.psp.example.com:3297/pemea/</delivery>
</emergencyDataReceived>
```

15.6 ASP to tPSP EDS

The ASP determines the tPSP, decrements the ttl, adds its address to the route and sends the EDS to the tPSP.

```
<emergencyDataSend xmlns="urn:pemea:apps:xml:ns:pemea:base"
                   xmlns:pdf="urn:ietf:params:xml:ns:pidf"
                       xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
                   xmlns:gml="http://www.opengis.net/gml"
                   xmlns:gs="http://www.opengis.net/pidflo/1.0"
                   xmlns:con="urn:ietf:params:xml:ns:geopriv:conf"
                   xmlns:pi="urn:ietf:params:xml:ns:EmergencyCallData:ProviderInfo"
                   xmlns:ecd="urn:ietf:params:xml:ns:EmergencyCallData"
                   xmlns:xc="urn:ietf:params:xml:ns:vcard-4.0"
                   xmlns:cell="urn:ietf:params:xml:ns:geopriv:lm:cell"
                   ttl="3"
                   onErrorPost="https://cooAP.example.com.be:2001/pemea/error/CoolAP-7496"
                   onCapSupporPost="https://cooAP.example.com.be:2001/pemea/cap/CoolAP-7496">
  <route msgSeq="CoolAP-7496" >
    <hops>
      <hop position="0" timeStamp="2016-02-02T18:14:00.001Z">
        <node>https://cooAP.example.com.be:2001/pemea/</node>
      </hop>
      <hop position="1" timeStamp="2016-02-02T18:14:00.521Z" >
        <node>https://opsp.example.com.be:2134/pemea/</node>
      </hop>
      <hop position="2" timeStamp="2016-02-02T18:14:00.980Z" >
        <node>https://asp.example.com.be:2195/pemea/</node>
      </hop>
    </hops>
  </route>
  <callerIds>
    <callerId typeOfId="msisdn">tel:+44-555-555-1234</callerId>
    <callerId typeOfId="msisdn">tel:+34-555-222-6789</callerId>
    <callerId typeOfId="skypeName">winterb</callerId>
  </callerIds>
  <apMoreInformation>
    <information typeOfInfo="Location_Update" protocol="HELD">
          https://coolap.example.com.be:2096/Web?id=CoolAP-7496
    </information>
  </apMoreInformation>
  <accessData>
    <cell:network>
      <cell:mcc>253</cell:mcc>
      <cell:mnc>002</cell:mnc>
    </cell:network>
  </accessData>
  <pdf:presence entity="tel:+44-555-555-1234">
    <pdf:tuple id="circle">
      <pdf:status>
        <gp:geopriv>
          <gp:location-info>
            <gs:Circle srsName="urn:ogc:def:crs:EPSG::4326">
              <gml:pos>42.5463 -73.2512</gml:pos>
              <gs:radius uom="urn:ogc:def:uom:EPSG::9001">
                   30.0
              </gs:radius>
            </gs:Circle>
            <con:confidence pdf="normal">95</con:confidence>
          </gp:location-info>
          <gp:usage-rules/>
          <gp:method>GNSS</gp:method>
          <gp:provided-by>
            <ecd:EmergencyCallDataValue>
              <pi:EmergencyCallData.ProviderInfo>
                <pi:DataProviderReference>xhjjshjsdhjsdh</pi:DataProviderReference>
                <pi:DataProviderString>Cool Application Provider</pi:DataProviderString>
                <pi:ProviderID>urn:pemea:pemea:ap:ID0x123FEDAC</pi:ProviderID>
         <pi:ProviderIDSeries>PEMEA</pi:ProviderIDSeries>
         <pi:TypeOfProvider>Application Provider</pi:TypeOfProvider>
         <pi:ContactURI>tel:+32-2534-9789</pi:ContactURI>
                <pi:Language>fr</pi:Language>
                <pi:DataProviderContact>
                  <xc:vcard>
                    <xc:org>
                      <xc:parameters>
                        <xc:language>
                          <xc:language-tag>en</xc:language-tag>
                        </xc:language>
                      </xc:parameters>
                      <xc:text>Really Application Provider</xc:text>
                    </xc:org>
                    <xc:adr>
                      <xc:parameters>
                        <xc:language>
                          <xc:language-tag>fr</xc:language-tag>
                        </xc:language>
                      </xc:parameters>
                      <xc:pobox>77222</xc:pobox>
                      <xc:ext/>
                      <xc:street>Avenue de la Toison d'Or, 79 - 3rd Floor</xc:street>
                      <xc:locality>Brussels</xc:locality>
                      <xc:region/>
                      <xc:code>1060</xc:code>
                      <xc:country>Belgium</xc:country>
                    </xc:adr>
                    <xc:email>
                      <xc:text>support@pemea.org</xc:text>
                    </xc:email>
                    <xc:url>
                      <xc:uri>http://www.pemea.help</xc:uri>
                    </xc:url>
                  </xc:vcard>
                </pi:DataProviderContact>
              </pi:EmergencyCallData.ProviderInfo>
            </ecd:EmergencyCallDataValue>
            <ecd:EmergencyCallDataReference purpose="EmergencyCallData.SubscriberInfo"
                                            ref="https://coolAP.pemea.org:7865/duheuh38x894nxe3iu3iu"/>
          </gp:provided-by>
        </gp:geopriv>
      </pdf:status>
    </pdf:tuple>
  </pdf:presence>
</emergencyDataSend>
```

15.7 tPSP to ASP EDR

The tPSP determines the correct PSAP. It adds its address to the route, and marks the destination as being a PSAP, along with the name of the PSAP. The example illustrates the name of the PSAP in the delivery element, this could also be a URI that provides a means to contact the PSAP.

```
<emergencyDataReceived xmlns="urn:pemea:apps:xml:ns:pemea:base"
                       xmlns:xs="http://www.w3.org/2001/XMLSchema"
                       xmlns:pemea="urn:pemea:apps:xml:ns:pemea:base"
       timeStamp="2016-02-02T18:14:01.327Z">
<route msgSeq="CoolAP-7496" >
  <hops>
    <hop position="0" timeStamp="2016-02-02T18:14:00.001Z">
      <node>https://cooAP.example.com.be:2001/pemea/</node>
    </hop>
    <hop position="1" timeStamp="2016-02-02T18:14:00.521Z">
      <node>https://orig.psp.example.com:2134/pemea/</node>
    </hop>
    <hop position="2" timeStamp="2016-02-02T18:14:00.980Z">
      <node>https://asp.example.com:2195/pemea/</node>
    </hop>
    <hop position="3" timeStamp="2016-02-02T18:14:01.327Z">
      <node> https://term.psp.example.com:3297/pemea/</node>
    </hop>
  </hops>
  </route>
  <delivery destType="PSAP">PSAP Serving Caller</delivery>
</emergencyDataReceived>
```

16 PEMEA Schema

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema
     targetNamespace="urn:pemea:apps:xml:ns:pemea:base"
     xmlns:xs="http://www.w3.org/2001/XMLSchema"
     xmlns:pemea="urn:pemea:apps:xml:ns:pemea:base"
     xmlns:cell="urn:ietf:params:xml:ns:geopriv:lm:cell"
     xmlns:wifi="urn:ietf:params:xml:ns:geopriv:lm:wifi"
     xmlns:xml="http://www.w3.org/XML/1998/namespace"
     elementFormDefault="qualified" attributeFormDefault="unqualified">

  <xs:annotation>
    <xs:documentation>
        The present documentdefines PEMEA messages.
    </xs:documentation>
  </xs:annotation>

  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
             schemaLocation="http://www.w3.org/2001/xml.xsd"/>

  <!-- import the cellular and wifi namespaces from RFC  7105 -->
  <xs:import namespace="urn:ietf:params:xml:ns:geopriv:lm:cell"/>
  <xs:import namespace="urn:ietf:params:xml:ns:geopriv:lm:wifi"/>

  <!-- posIntType -->

  <xs:simpleType name="posIntType">
    <xs:restriction base="xs:nonNegativeInteger">
      <xs:minInclusive value="0"/>
    </xs:restriction>
  </xs:simpleType>

  <!-- nodeType -->

  <xs:complexType name="nodeType">
    <xs:complexContent>
      <xs:restriction base="xs:anyType">
        <xs:sequence>
          <xs:element name="node" type="xs:anyURI" minOccurs="1" maxOccurs="1"/>
          <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="position" type="pemea:posIntType" use="required"/>
        <xs:attribute name="timeStamp" type="xs:dateTime" use="required"/>
        <xs:anyAttribute namespace="##any" processContents="lax"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <!-- hopsType -->

  <xs:complexType name="hopsType">
    <xs:complexContent>
      <xs:restriction base="xs:anyType">
        <xs:sequence>
          <xs:element name="hop" type="pemea:nodeType" minOccurs="1" maxOccurs="unbounded"/>
          <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:anyAttribute namespace="##any" processContents="lax"/>
      </xs:restriction>
    </xs:complexContent>
   </xs:complexType>

  <!-- baseRouteType -->
```

```
<xs:complexType name="baseRouteType">
  <xs:complexContent>
    <xs:restriction base="xs:anyType">
      <xs:sequence/>
        <xs:attribute name="msgSeq" type="xs:token" use="required"/>
        <xs:anyAttribute namespace="##any" processContents="lax"/>
     </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<!-- routeType -->

<xs:complexType name="routeType">
  <xs:complexContent>
    <xs:extension base="pemea:baseRouteType">
      <xs:sequence>
        <xs:element name="hops" type="pemea:hopsType" minOccurs="1" maxOccurs="1"/>
        <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!-- destinationType -->

<xs:simpleType name="destinationType">
  <xs:restriction base="xs:token">
    <xs:enumeration value="PSAP"/>
    <xs:enumeration value="PSP"/>
    <xs:enumeration value="ASP"/>
  </xs:restriction>
</xs:simpleType>

<!-- destinationNodeType -->

<xs:simpleType name="destinationNodeType">
  <xs:union>
    <xs:simpleType>
      <xs:restriction base="xs:token">
        <xs:enumeration value="any"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType>
      <xs:restriction base="xs:anyURI">
        <xs:minLength value="1"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>

<!-- deliveryType -->

<xs:complexType name="deliveryType">
  <xs:simpleContent>
    <xs:extension base="pemea:destinationNodeType">
      <xs:attribute name="destType" type="pemea:destinationType" />
      <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<!-- typeOfcallerIdType -->

<xs:simpleType name="typeOfCallerIdType">
  <xs:union>
    <xs:simpleType>
      <xs:restriction base="xs:token">
        <xs:enumeration value="any"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType>
      <xs:restriction base="xs:anyURI">
        <xs:minLength value="1"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
```

```
  <!-- callerIdType -->

  <xs:complexType name="callerIdType">
    <xs:simpleContent>
      <xs:extension base="pemea:typeOfCallerIdType">
        <xs:attribute name="typeOfId" type="xs:token" />
        <xs:anyAttribute namespace="##any" processContents="lax"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <!-- callerIdListType -->

  <xs:complexType name="callerIdListType">
    <xs:complexContent>
      <xs:restriction base="xs:anyType">
        <xs:sequence>
          <xs:element name="callerId" type="pemea:callerIdType" minOccurs="1"
maxOccurs="unbounded"/>
          <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:anyAttribute namespace="##any" processContents="lax"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <!-- informationType -->

  <xs:complexType name="informationType">
    <xs:simpleContent>
      <xs:extension base="xs:anyURI">
        <xs:attribute name="typeOfInfo" type="xs:token" use="required"/>
        <xs:attribute name="protocol" type="xs:token" use="optional"/>
        <xs:anyAttribute namespace="##any" processContents="lax"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <!-- apMoreInfoType -->

  <xs:complexType name="apMoreInfoType">
    <xs:complexContent>
      <xs:restriction base="xs:anyType">
        <xs:sequence>
          <xs:element name="information" type="pemea:informationType"
                      minOccurs="1" maxOccurs="unbounded"/>
          <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:anyAttribute namespace="##any" processContents="lax"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:element name="apMoreInformation" type="pemea:apMoreInfoType"/>

  <!-- Access Data Types-->

  <xs:complexType name="accessDataBaseType">
    <xs:choice>
      <xs:element ref="cell:network"/>
      <xs:element ref="wifi:wifi"/>
    </xs:choice>
  </xs:complexType>

  <xs:element name="accessDataType" type="pemea:accessDataBaseType"/>

  <!-- Access Data -->

  <xs:complexType name="accessData">
    <xs:complexContent>
      <xs:restriction base="xs:anyType">
        <xs:sequence>
          <xs:element ref="pemea:accessDataType" minOccurs="0" maxOccurs="2"/>
          <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:anyAttribute namespace="##any" processContents="lax"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <!-- edsBaseType -->

  <xs:complexType name="edsBaseType">
    <xs:complexContent>
      <xs:restriction base="xs:anyType">
        <xs:sequence/>
        <xs:attribute name="ttl" type="pemea:posIntType" use="required"/>
        <xs:attribute name="onErrorPost" type="xs:anyURI" use="optional"/>
        <xs:attribute name="onCapSupportPost" type="xs:anyURI" use="optional"/>
        <xs:anyAttribute namespace="##any" processContents="lax"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <!-- emergencyDataSend -->

  <xs:complexType name="edsType">
    <xs:complexContent>
      <xs:extension base="pemea:edsBaseType">
        <xs:sequence>
          <xs:element name="route" type="pemea:routeType" minOccurs="1" maxOccurs="1"/>
          <xs:element name="callerIds" type="pemea:callerIdListType" minOccurs="1" maxOccurs="1"/>
          <xs:element name="apMoreInformation" type="pemea:apMoreInfoType" minOccurs="0"
maxOccurs="1"/>
          <xs:element name="accessData" type="pemea:accessData" minOccurs="0" maxOccurs="1"/>
          <xs:any namespace="##other" processContents="lax" minOccurs="1" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:element name="emergencyDataSend" type="pemea:edsType"/>

  <!-- edrBaseType -->

  <xs:complexType name="edrBaseType">
    <xs:complexContent>
      <xs:restriction base="xs:anyType">
        <xs:sequence/>
        <xs:attribute name="timeStamp" type="xs:dateTime" use="required"/>
        <xs:anyAttribute namespace="##any" processContents="lax"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <!-- emergencyDataReceived -->

  <xs:complexType name="edrType">
    <xs:complexContent>
      <xs:extension base="pemea:edrBaseType">
        <xs:sequence>
          <xs:element name="route" type="pemea:routeType" minOccurs="1" maxOccurs="1"/>
          <xs:element name="delivery" type="pemea:deliveryType"/>
          <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:element name="emergencyDataReceived" type="pemea:edrType"/>

  <!-- errorBaseType -->

  <xs:complexType name="errorBaseType">
    <xs:complexContent>
      <xs:restriction base="xs:anyType">
        <xs:sequence/>
        <xs:attribute name="timeStamp" type="xs:dateTime" use="required"/>
        <xs:anyAttribute namespace="##any" processContents="lax"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="msgInfoType">
    <xs:simpleContent>
      <xs:extension base="xs:token">
```

```
        <xs:attribute ref="xml:lang"/>
        <xs:anyAttribute namespace="##any" processContents="lax"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>


  <!-- errorType -->

  <xs:complexType name="errorType">
    <xs:complexContent>
      <xs:extension base="pemea:errorBaseType">
        <xs:sequence>
          <xs:element name="reason" type="xs:token" minOccurs="1" maxOccurs="1"/>
          <xs:element name="message" type="pemea:msgInfoType" minOccurs="0" maxOccurs="1" />
          <xs:element name="route" type="pemea:routeType" minOccurs="1" maxOccurs="1"/>
          <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:element name="error" type="pemea:errorType"/>

</xs:schema>
```

Annex A (informative): Route Determination

The details of this annex are informative in nature only. The PEMEA architecture defined in the present document is heavily dependent on the ability of PSPs and ASPs to determine where to direct the emergencyDataSend messages. Exactly how this occurs is purposefully left out of scope and for implementations to address. This is because there is no one single way to address this issue that all regions and areas can or will comply with. As a consequence, different regions will support different mechanisms for identifying the correct PSP to deliver the data to.

There are several key pieces of data provided by the AP in the emergencyDataSend message that enable a PSP to make some determinations about whether the data is destined for a PSAP that they directly serve or if they should hand off to another PSP or to an ASP. The key pieces of data are obviously the current serving node (cell or WiFi) and the actual location provided by the App.

There is nothing about a WiFi BSSID that intrinsically indicates location. To convert the BSSID into location requires the use of a third-party database, which may be operated by a private company for commercial or regulated services. Consequently, receiving a WiFi BSSID is notionally equivalent to determining the destination based on the App proffered location.

The cellular mobile serving cell is a structured identifier that provides quite a lot of information about where and which network the caller is using. The serving cell is made up of the Mobile Country Code (MCC), the Mobile Network Code (MNC), and the unique cell within that network. Since cell-id is the normal call routing identifier used across Europe for mobile calls, routing at this level of granularity should ensure that the data gets to where it needs to go. Further to this, the PSP has a direct relationship with the PSAP, and the PSAP has worked out with the mobile operators which cell-ids should route calls to it. Therefore, it may be possible for the PSP to obtain the cell lists from the PSAPs allowing them to determine which data to keep local and which data to send to the ASP.

More sophisticated PSAPs define their boundaries using geodetic coordinates often in the form of shape files. In this case, the PSP can employ a geospatial solution to determine if the location provided by the App is for a local PSAP or if it needs to be passed to an ASP.

In some cases, PSAPs boundaries not defined by polygons and geodetic coordinates by are represented by civic or municipal descriptions. While these are ultimately likely to be defined by some kind of spatial reference, the reference is not readily available. In such cases PSPs and ASPs can employ reverse geocode solutions, freely available, public services, or proprietary solutions to obtain a civic representation of any proffered or determined geodetic location. Once obtained, elements of the address can be used to determine if the data is intended for local consumption or should be sent on to an ASP.

The architecture does not constrain the number of ASPs that may exist, nor does it constrain the number of ASPs with which a PSP may have a relationship. As a consequence, an ASP may employ all of the above techniques to determine a subsequent ASP or PSP to direct the data to. It is key, however, that the PSP or ASP be able to determine and use the correct authentication credentials once a next hop is decided on.

In order to better understand how routing in this way may be possible it is useful to provide a concrete example. The example that has been chosen is Spain.

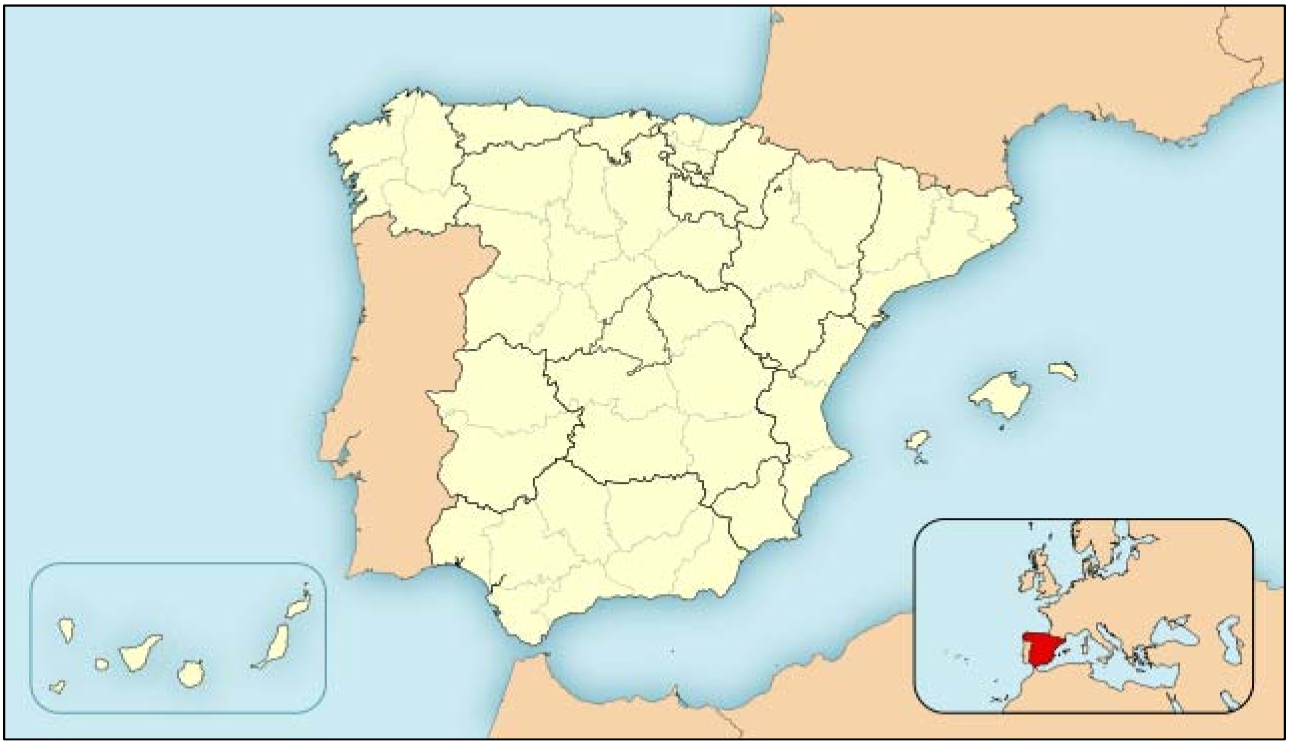


Figure A.1: Regions of Spain

As can be seen in Figure A.1, Spain has 19 regions (15 continental, 2 archipelagos, and 2 cities in northern Africa), and PSAPs are regional entities, so it only becomes necessary to identify the region in order to determine the correct PSAP, and therefore the PSP, to send the information to.

Each region in Spain consists of a number of smaller areas (provinces), as the first two number of the five-digit postal code indicates the province. A post code is assigned to an area and that area can only reside in one region, that is, a post code does not span different regions. So for PSAP routing in Spain it is sufficient to know the post code to know the region and hence know the PSAP/PSP.

In most cases the Smartphone App will provide a geodetic shape, a circle or ellipse perhaps, and the cell-id. The Cell-id allows any routing entity to know which country the caller is in based on the Mobile Country Code (MCC) component. In the case of Spain, the MCC is 214. The routing entity can then use an online reverse geocoding service to obtain an approximate civic address for where the caller is. In the case of Spain, this only needs to be sufficiently good to determine the region. Once the region is known the entity can use a database table map similar to the one below.

Table A.1

Country	Postcode	Region	Province	PSP URI
Spain	08001	Catalonia	Barcelona	https://cat.psp.sp:5980/pemea
Spain	17001	Catalonia	Girona	https://cat.psp.sp:5980/pemea
Spain	25001	Catalonia/	Lleida	https://cat.psp.sp:5980/pemea
Spain	43001	Catalonia	Tarragona	https://cat.psp.sp:5980/pemea
Spain	31001	Navarra		https://nav.psp.sp:5980/pemea
Spain	28001	Madrid		https://mad.psp.sp:5980/pemea

Not all countries will be quite as simple as Spain, and others will be far easier. Reverse geocoding is sufficient in most cases to determine the correct PSP to direct the data to. However, different countries may need different keys to allow them to identify the correct PSP.

Annex B (informative): Caller Data

This annex is informative but provides the basis for the information that should be provided in the protocols and messaging specification to follow this one.

Table B.1

Information Type	Recommended Usage	Description
Family name	Mandatory	The family or surname(s) of the caller. Some countries support multiple unhyphenated family names
Given Name	Mandatory	The given or first name(s) of the caller
Additional	Optional	Any other names that the caller may have
Prefix	Optional	Salutation such as Mr, Ms, Dr
Suffix	Optional	Generation, such Jr., III
Home Address	Conditional	Full home address of the caller if available including country
Language	Mandatory	The languages spoken by the caller. Non-oral languages such as local or national sign-language dialects should also be supported
Gender	Recommended	The Gender of the caller
Date of birth	Recommended	Allow the age of the caller to be determined
Other Contacts	Recommended	Ways other than the provided calling party number that the caller may be contactable
Emergency Family Contacts	Recommended	Next of kin or family members that may be contacted in required

All information above is relevant to the person that registered the application who will in most circumstances but not in all circumstances be the caller.

In some countries, it is understood that the PSAP does not automatically have access to the name of the caller, in places where this occurs the PSP is responsible for gatekeeping this information from the PSAP.

This data set is deliberately minimalistic, the protocol sets in the implementation specification provide extension points so that further caller information can be added in a backwards compatible way.

Annex C (informative): Additional AP Information

There are a number of emergency applications deployed today that implement specialized features that provide what their customer base sees as being useful. It is hard in an open specification such as this to cater for all of these options explicitly. It is also hard to expect all PSAPs to intuitively understand how to use, interpret and render this information without some prior knowledge of the application from which the data is sourced.

Apps may offer a whole range of communication features that augment the voice call. These may include instant messaging, chat and video-conferencing for those with speech and/or hearing disabilities.

PEMEA supports these capabilities by providing a "*Reach-back URI*", that allows the originating AP to indicate that it can provide more information to the PSAP if required.

Another proposed usage of Additional AP information element is to use it to convey to the destination PSAP and alerting URI for the AP. This allows a PSAP the ability to build up a table of all possible APs to which it may send alert messages to in the future. Having the PSAPs contact the APs directly avoids avalanche issues that may occur in centralized routing solutions such as PEMEA, however the PEMEA architecture is well suited to allowing the PSAPs to develop these direct AP messaging tables.

History

Document history		
V1.1.1	March 2018	Publication
V1.2.1	March 2020	Publication