EENA NG112 Technical Committee Document

Pan-European Mobile Emergency Application (PEMEA) Requirements and Functional Architecture

Title:	PEMEA Requirements and Functional Architecture		
Version:	7		
Revision Date:	02-12-2015		
Status of the document:	Draft	For comments	**Approved**

EENA asbl
info@eena.org - www.eena.org

Authors and contributors to this document

This document was written by members of EENA:

Authors	Country / Organisation
James Winterbottom	EENA Technical Committee Vice-chair

Contributors	Country / Organisation
Bertrand Casse	EENA Operations Committee Vice-chair - Deveryware
Cristina Lumberas	EENA
Peter Sanders	EENA Operations Committee Vice-chair – One2Many
Iratxe Gomez	EENA Operations Committee Co-Chair - Atos
Panayiotis Kolios	Kios Research Center – University of Cyprus

EENA asbl
info@eena.org - www.eena.org

Table of contents

EENA asbl
info@eena.org - www.eena.org

1 Executive Summary

There are presently hundreds of emergency calling applications in use across Europe but none of them permit seamless roaming and so are constrained in where they can be used. The aim of the PEMEA architecture is to allow a user to select an application that meets their needs based on functionality, cost and levels of comfort with proper security and privacy setting and provide accurate location and other information to emergency services in a time of need, anywhere across Europe. This document describes a functional architecture, requirements and data flows to support existing application providers and PSAP service providers to communicate with other providers and so enable mobile emergency calling applications to operate while roaming anywhere in Europe.

This document provides the functional architecture, requirements, and call flows for PEMEA. The companion document to this one is the protocol definitions and procedures document that defines the protocols, message details and specific nodal behaviour.

2 Introduction

This document revises the EENA 112 Smartphone Apps document [1] and provides the requirements and functional architecture for the pan-European Mobile Emergency Application (PEMEA) architecture, outlined in the EENA 112 Apps Strategy document [2]. The changes introduced by the PEMEA architecture enable much better service differentiation in terms of what functions an App can provide, while ensuring compatibility and allowing certification of interfaces that provide information to PSAP networks. The architecture also seeks to address some of the security issues raised during the review cycle of the original 112 Smartphone Apps document.

The data flows between the App running on the mobile devices and the provider of the App service (the App provider) have been moved out of scope and there is no longer any need for the app to maintain a country-based routing table. All information provided by the mobile application is sent to the App-provider, and it is the responsibility of the App provider to ensure that this data is formatted correctly prior to sending it on to the agency serving the PSAP.

This document defines this new functional architecture and assigns responsibilities to each of the identified entities. Further the document defines the information exchange formats between entities where interoperability between different organizations is required. Interfaces that are deemed out of scope are explicitly stated as such. It is envisaged that this specification will form the basis of a certification program, so any of the in-scope interfaces will need to conform to this specification and be tested against a benchmark implementation prior to certification being granted.

2.1 Why use an App with emergency calling?

The original driver behind mobile emergency applications was to use the high-accuracy location capabilities in smart-phones and make this information available to PSAPs and emergency authorities in the absence of carriers implementing accurate control-plane location solutions. It can be argued that this still the primary aim, but it quickly becomes apparent that important additional information about the caller can be provided to the PSAP over and above location information. It is this additional caller information and the use of NG112 data formats that makes the implementation and use of PEMEA a natural stepping-stone toward the implementation of NG112 and WebRTC emergency service solutions.

2.2 Why the changes from the first EENA emergency Apps architecture to PEMEA?

The primary purpose of the first EENA Apps document was to investigate options and make a recommendation. While this document did define a high-level architecture and recommended the use of PIDF-LO with additional data for the data exchange format it didn't take into account existing applications and it left the security aspects of the application for a future document.

Since the first EENA Apps document was created a plethora of new apps have appeared on the market, some of which are very feature rich. Ensuring that Application providers can provide functional differentiation between each other is key to ensuring a vibrant and diverse eco-system serving the different emergency needs of citizens with different interests and needs.

Currently, mobile emergency applications are regional and restricted in where they can provide service. Many application providers have indicated that they have a desire to extend where they can provide service. However, this will lead to multiple different applications providing service overlap because each offers different features to their user-base. To support the plurality of applications, achieving this goal requires a consistent, coordinated and cohesive approach.

The first version of the architecture required every country to have a national application server. In some cases this approach would require legislative changes and these would pose huge barriers to general acceptance and deployment.

Finally, the security model in the first version called for a pan-European certificate authority in order to authenticate legitimate applications but provided no means to authenticate a specific user. As a consequence beyond the App being legitimate there was no easy way to ascertain if the communication being initiated was real, or if the identity information being provided by the caller was real.

The changes introduced by this document attempt to address the short-comings and issues described above.

2.3 The role of PEMEA

The original role of the 112 Smartphone Apps document was to recommend an architecture to enable the delivery of emergency calls with highly-accurate location and caller information to PSAP during an emergency anywhere in Europe. Considering all the issues described above, the need arises to amend the original architecture in order to meet this goal.

PEMEA provides a functional architecture, defining roles and responsibilities as well as data exchange formats and a general security model so that PSAPs can be sure of the veracity of the information being provided and application users can be sure that information is not being misused. So PEMEA can be thought of as a framework for enabling pan-European emergency applications.

Data provided through the PEMEA framework augments, and is supplementary to, the location information provided to PSAPs by MNOs. The PEMEA network may not always be able to deliver information to the PSAP in which case the PSAP must rely solely on the information provided by the MNO. In this manner, PEMEA provides enhanced capabilities to the PSAP but does not guarantee delivery of information for every call.

3 PEMEA architecture and functional entities

As stated earlier, the extensive deployment of existing mobile emergency applications and their interconnection into a Pan-European Emergency Application ecosystem has prompted a move away from the architecture proposed in PEMEA Version 1. As a consequence new entities and responsibilities have been identified and their points of interaction identified in Figure 1.

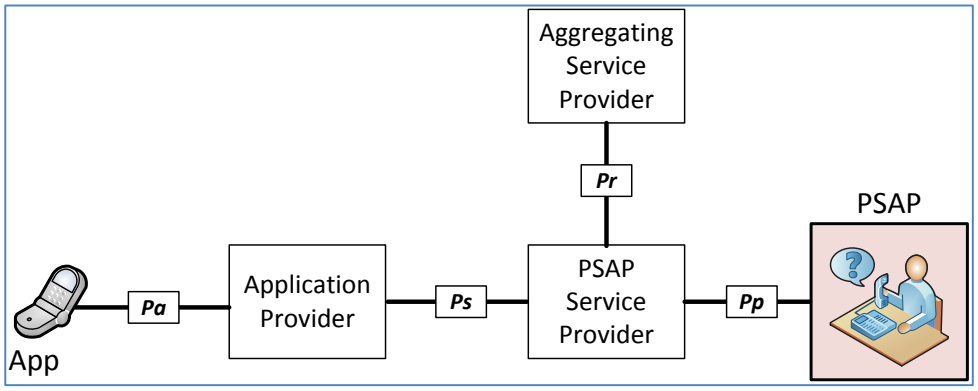


Figure 1 PEMEA Reference Architecture

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

3.1 Functional entities overview

The following descriptions apply only to the entities as shown in Figure 1 and only relate their usage in providing location and additional information related to an emergency call.

3.1.1 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.

3.1.2 Application Provider (AP)

The Application Provider (AP) is the entity that provides a mobile emergency application. It is responsible for authenticating the Application user 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 this document must 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 this document, but the AP must only send the information to one PSP to avoid multiple routing of the same messages through the network.

3.1.3 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).

Information coming from trusted sources must comply with the data formats and communication mechanisms defined in this 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 must be 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 this document.

3.1.4 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 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 this document but may be addressed by subsequent EENA operational considerations.

3.2 Interface Definitions

3.2.1 PEMEA 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 this specification 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 must provide. How these requirements are implemented is out of scope. The nominal interface is likely to be based on a data connection, however, SMS may be used as a fall back to signal the Application Provider.

3.2.2 PEMEA 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 must implement this interface in accordance with the details in this specification when they are not the same entity.

3.2.3 PEMEA 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 must implement this interface in accordance with the details in this specification to be PEMEA compliant.

3.2.4 PEMEA PSAP Interface (Pp)

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

Integrating new data sources into PSAP CPE can be costly and take time. If a PSAP wishes a cost effective and fast means of getting access to PEMEA data then the PSP may provide the PSAP with simple web access to the information. One example of how this may be provided is for the PSP to provide a web service where the PSAP call taker enters the calling number and receives a web page containing the data and a map of where the caller is.

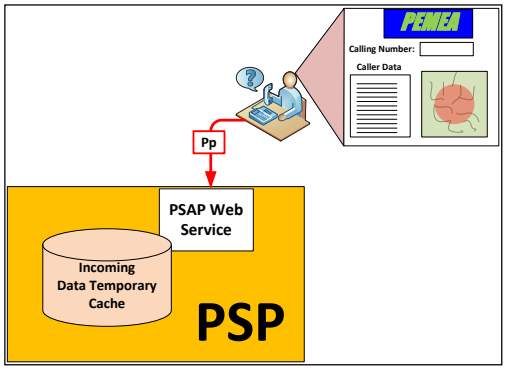


Figure 2 Basic PEMEA PSAP Integration

4 PEMEA Functional Entity Requirements

4.1 Smartphone Application Requirements

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

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

SA-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 since delay may be considerable.

SA-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.[1]

[1] It is understood that increasingly mobile operating systems are not providing Application access to this information. The Application should try to acquire it where possible as it may allow for faster data routing in some circumstances.

SA-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.[2]

SA-6. The Application should provide a mechanism for sending location and, if possible, serving cell information to the AP using SMS in the case where data service is unavailable.

4.2 Application Provider Requirements

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

AP-2. The AP shall have a trust relationship with a PSP.

AP-3. The AP shall authenticate with the PSP using a client-side X.509 certificate[3] provided to the AP by the PSP when caller data is to be sent to the PSP.

AP-4. The AP shall comply with the ***Ps*** interface specification to convey information to a PSP.

AP-5. The AP may provide a means for the destination PSAP to obtain application specific information from the AP.

4.3 PSAP Service Provider Requirements

PSP-1. The PSP shall provide a signed X.509 certificate to APs with which it agrees to accept caller information over the ***Ps*** interface.

PSP-2. A PSP shall authenticate the AP each time data is pushed.

PSP-3. A PSP shall never accept connections from an AP that fails to authenticate.

PSP-4. An oPSP shall not cache caller information if the information is pushed to a tPSP or to an ASP over the ***Pr*** interface.

PSP-5. The PSP where the call is terminated shall not cache or log caller information for longer than terminating PSAP statutes allow, and should adhere to the ***retention-expires*** value in the ***usage-rules*** element of the PIDF-LO [3].

PSP-6. If the PSP is unable to determine where the caller information should be delivered then it shall return an error to the node attempting to provide it with the information.

PSP-7. A tPSP shall provide a signed X.509 certificate to oPSPs or ASPs with which it agrees to accept caller location over the ***Pr*** interface.

PSP-8. A tPSP shall authenticate an oPSP or ASP each time is connects.

PSP-9. An oPSP shall obtain an X.509 certificate from each ASP and tPSP to which it intends to deliver caller information over the ***Pr*** interface.

PSP-10. A tPSP shall never accept connections from an oPSP or an ASP that fails to authenticate.

4.4 Aggregating Service Provider Requirements

ASP-1. The ASP shall provide a signed X.509 certificate to any oPSP with which it agrees to accept caller information from over the ***Pr*** interface.

ASP-2. The ASP shall obtain an X.509 certificate from each tPSP to which it intends to deliver caller information over the ***Pr*** interface.

ASP-3. The ASP shall never pass caller information to a tPSP that fails to authenticate.

ASP-4. The ASP shall never accept connections from an oPSP that fails to authenticate.

ASP-5. The ASP shall authenticate an oPSP each time it connects.

---

[2] It is understood that increasingly mobile operating systems are not providing access to the MSISDN or IMSI or the device. The Application should try to acquire this information where possible.

[3] http://www.itu.int/rec/T-REC-X.509-201210-I/en

ASP-6. The ASP shall authenticate itself to a tPSP each time it connects.

ASP-7. The ASP shall return an error to the oPSP if it is unable to determine where to send the caller information.

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

5 PEMEA Message Element Definitions

To keep PEMEA simple there are only three 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 subsections define the information elements required for each of these messages.

5.1 emergencyDataSend Information Elements

Table 1 emergencyDateSend Definition

Element	Inclusion	Description
Time To Live	Mandatory	Defines the number of hops allowed before message delivery must stop.
Route	Mandatory	Defines the nodes and the order through which the message has passed the nodes. This will include the node sending this message.
MSISDN	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 must appear at least once [4]. If the App cannot obtain this from the device-API at call time, then it must be configured when the application is installed or registered with the AP.
IMSI	Conditional	The device as identified by the home mobile operator. In Europe it is common for phones to support more than one SIM card, consequently this element may appear multiple times. If this value is available through the device-API then it must be provided, where the device-API does not support obtaining this value then it may be omitted.
IMEI	Conditional	The unique manufacturers serial number for the device. If this value is available through the device-API then it must be provided, where the device-API does not support obtaining this value then it may be omitted.
Location information	Mandatory	The location as determined by the device.
Serving Cell	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 must be 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.
Caller Information	Mandatory	Contains information about the caller. See Annex C for more information about what kind of caller information may be provided.
More Information	Optional	Anything additional that the AP wishes to provide to the PSAP, such as a URI for obtaining additional information. See Annex D for more information about this kind of information.

[4] SIMless devices are not considered in this specification but may be an area for 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 Definition

Element	Inclusion	Description
Sequence Number	Mandatory	A unique sequence number generated by the AP when it creates the emergencyDataSend message.
Hop	Mandatory	This a complex element that defines each node (hop) in the PEMEA signalling. There must 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 and the link number in the chain of nodes.

Table 3 Hop 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.

5.2 emergencyDataReceived Information Elements

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.

5.3 error Information Elements

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 PEMEA Message flows

The *Pa* and *Pp* message flows are out of scope of this document, so this section 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 specification.

6.1 Ps message flows

The *Ps* interface is the interface between the AP and PSP. The connection originates from the AP making it the client and PSP the server. The AP puts its data into an emergencyDataSend message (the precise format for this message is defined in a detailed protocol specification) 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 services 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 this document.

6.1.1 Ps basic flow

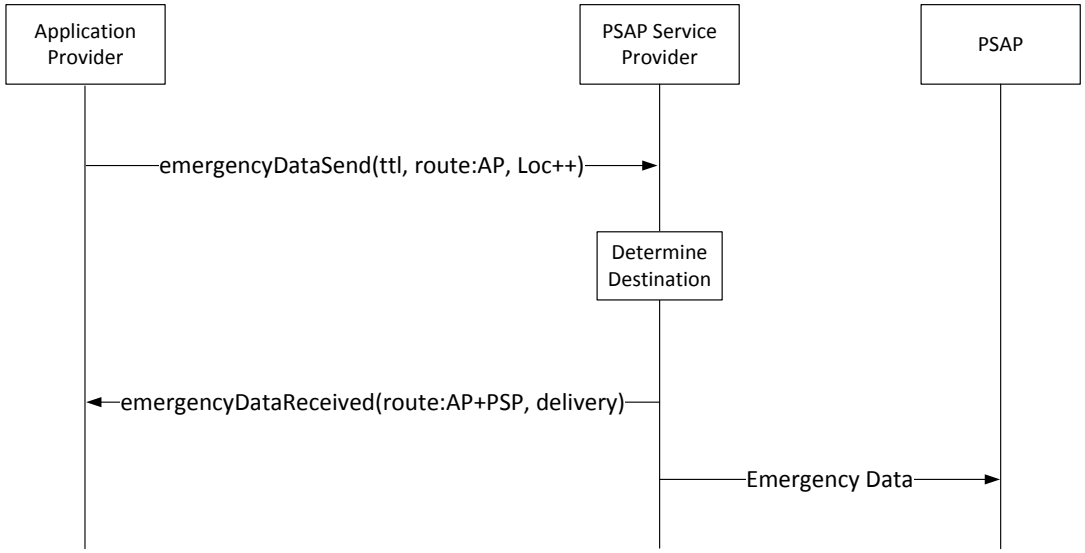


Figure 3 Ps basic message flow

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 must be dropped. The minimum value that the AP may set the ttl value to is 1.
3. adds its identity information to the route element
4. assembles and appends the location and other information
5. 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. add 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

6.1.2 Ps error flow

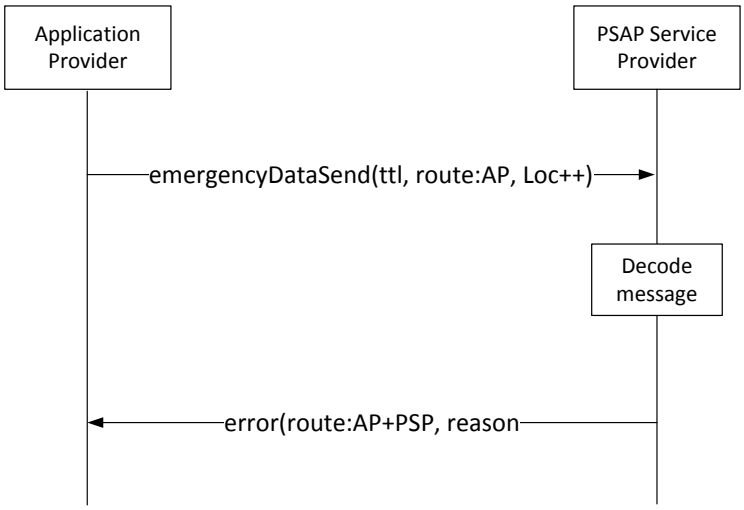


Figure 4 Ps error flow

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 must be dropped.
3. adds its identity information to the route element
4. assembles and appends the location and other information
5. 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. add 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

6.1.3 Ps routing flow

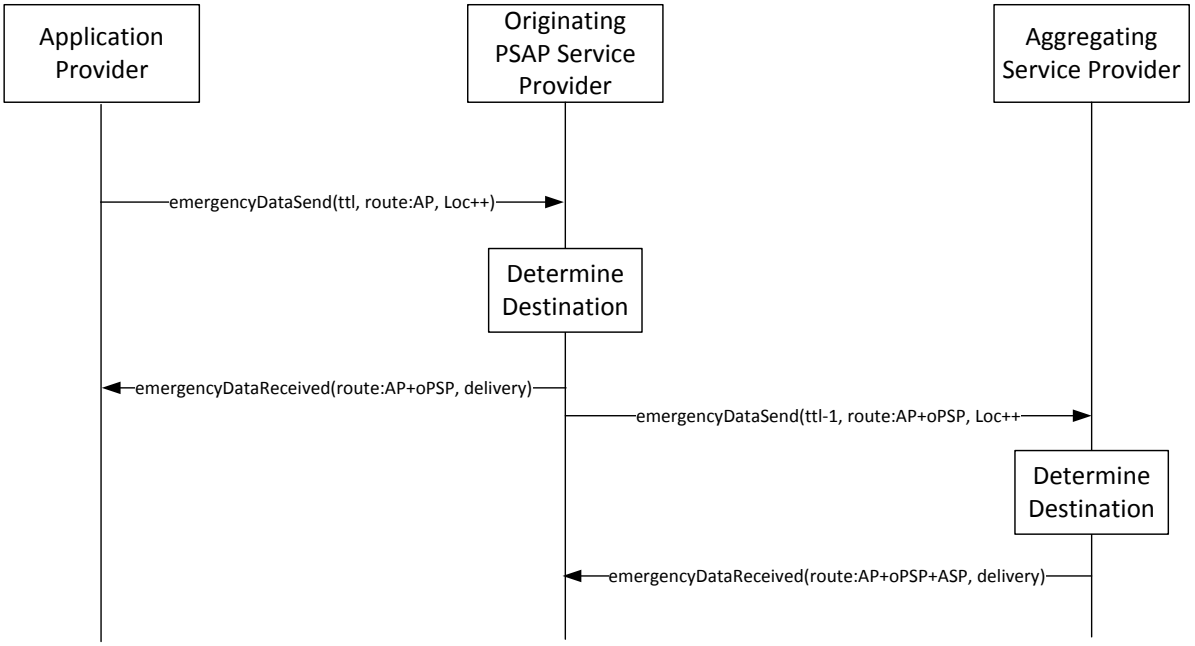


Figure 5 Ps routing flow

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 must be dropped
3. adds its identity information to the route element
4. assembles and appends the location and other information
5. 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. add 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. add 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

6.2 Pr message flows

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 section) and uses HTTP to deliver the message to the receiver (either the PSP or the ASP).

6.2.1 Pr terminating-PSP basic flow

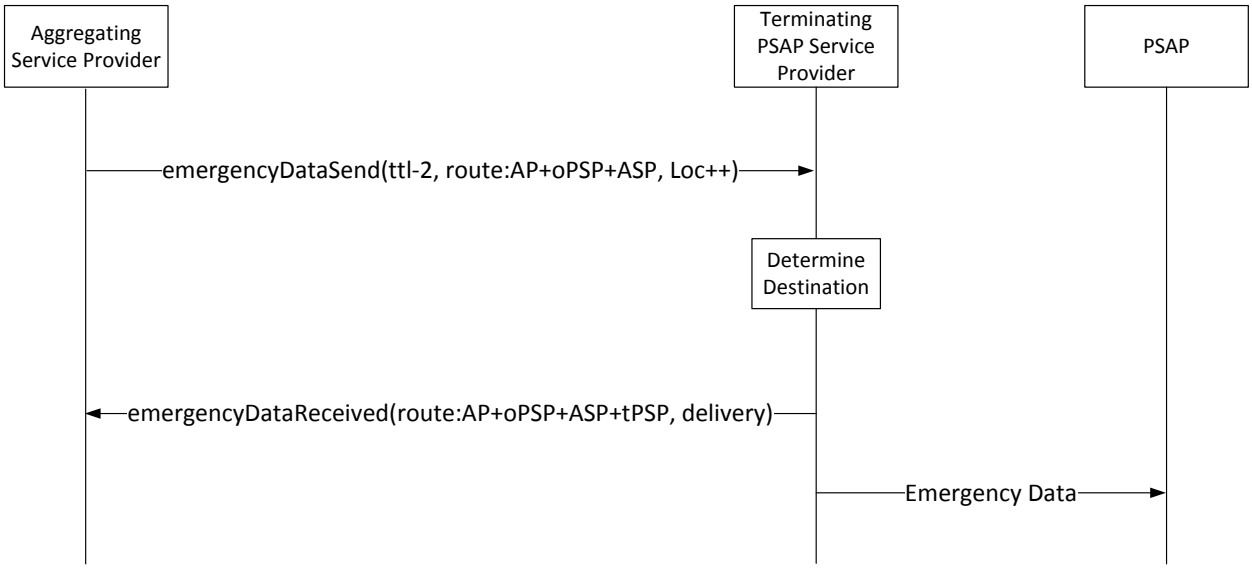


Figure 6 Pr basic message flow

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. add 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

6.2.2 Pr error flow

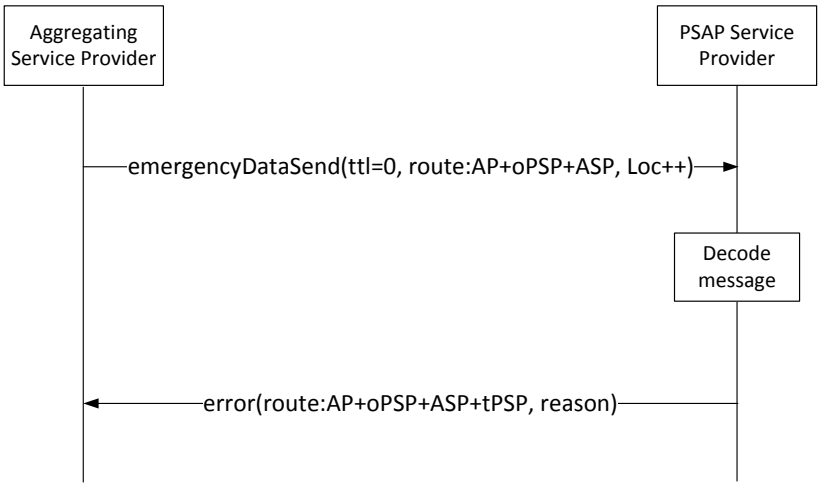


Figure 7 Pr error flow

In this flow the ASP:

1. receives the emergencyDataSend message from the oPSP
2. determines the correct tPSP and send 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

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. add 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

6.2.3 Pr end to end routing flow

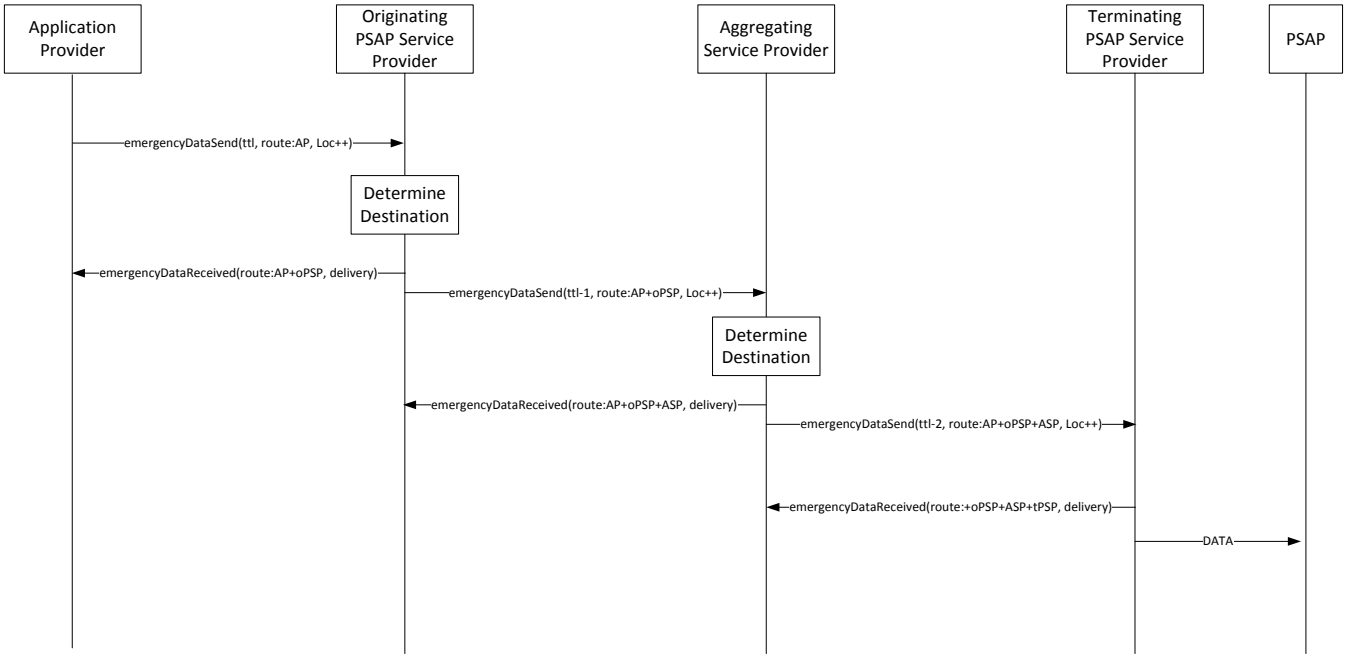


Figure 8 Ps routing flow

In this flow the AP:

1. creates and emergencyDataSend message:
2. sets the time to live (ttl) to the number of hops the AP will allow before the message must be 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

7 Abbreviations

AP	Application Provider
App	Application
ASP	Aggregating Service Provider
BSSID	Basic Service Set Identifier
CID	Cell Identifier
GNSS	Global Navigation Satellite System
CPE	Customer Premises Equipment
GPS	Global Positioning System (a type of GNSS)
HTTP	Hyper-Text Transfer Protocol
IETF	Internet Engineering Task Force
IMEI	International Mobile Equipment Identifier
IMSI	International Mobile Subscriber Identifier
MCC	Mobile Country Code
MNC	Mobile Network Code
MNO	Mobile Network Operator
MSISDN	Mobile Service International Subscriber Dial Number
NEAD	National Emergency Address Database
oPSP	Originating PSP
PEMEA	Pan European Mobile Emergency Application
PIDF-LO	Presence Information Data Format Location Object
PSAP	Public Safety Answering Point
PSP	PSAP Service Provider
tPSP	Terminating PSP
ttl	Time To Live
URI	Universal Resource Identifier
XML	eXtensible Markup Language
XSD	XML Schema Description

8 References

[1] "112 Smartphone Apps", 2.2.3, Version 1.0, 25 February 2014.

[2] "112 Apps Strategy - Pan European Mobile Emergency Apps", 2015_03_17_PEMEA, Version 1.7, 17 March 2015.

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

[4] http://www.itu.int/rec/T-REC-X.509-201210-I/en

Annex A Route Determination

The detail of this annex are informative in nature only. The PEMEA architecture defined in this 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.

Annex A-1 Concrete Routing Example

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 9 Regions of Spain

As can be seen in Figure 9, 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:

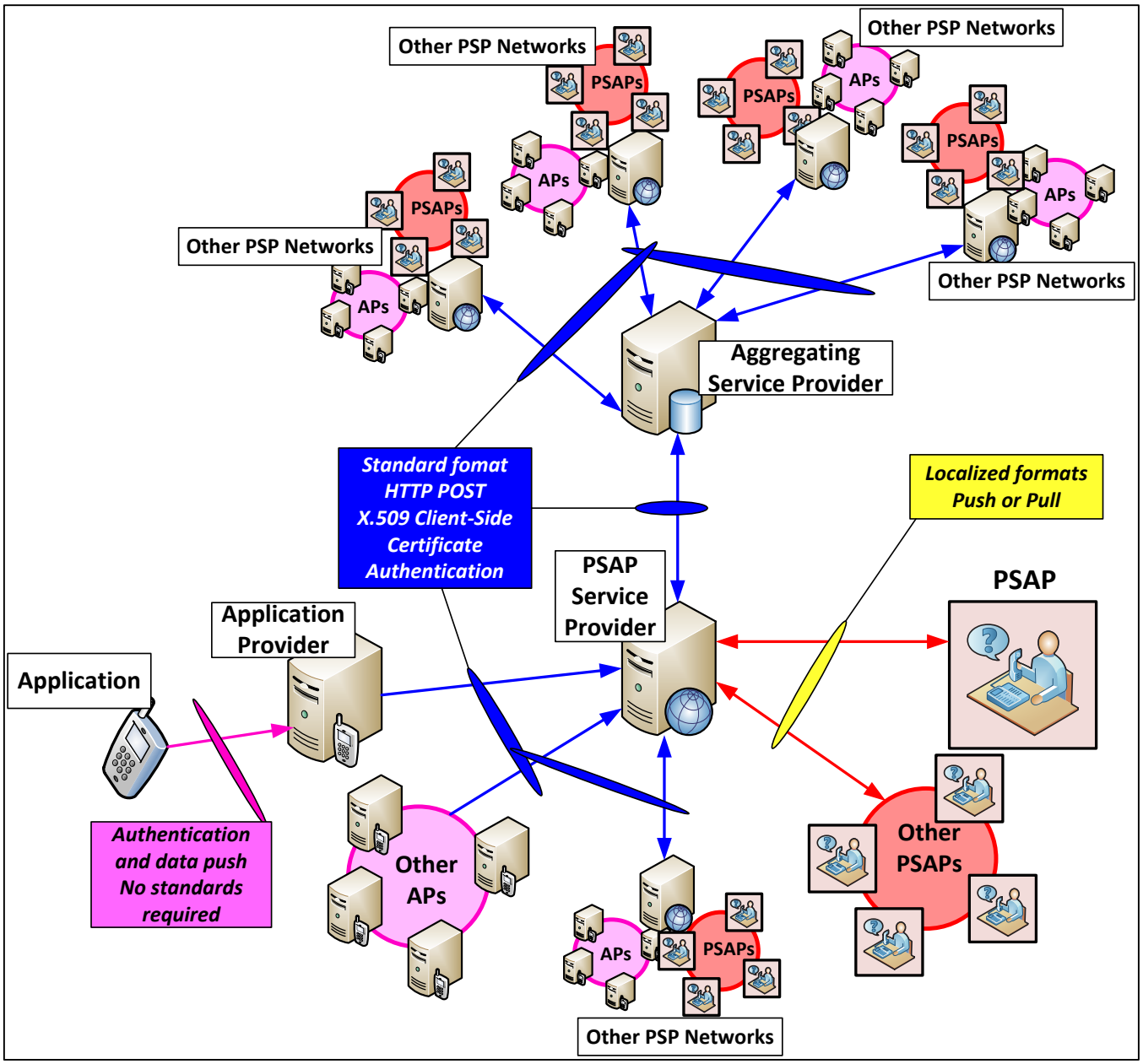
Country	Postcode	Region	Province	PSP URI
Spain	08001	Catalonia	Barcelona	https://cat.psp.sp:5980/pemea/
Spain	17001	Catalonia	Girona	https://cat.psp.sp:5980/pemea/
Spain	25001	Catalonia/	Lleida	https://cat.psp.sp:5980/pemea/
Spain	43001	Catalonia	Tarragona	https://cat.psp.sp:5980/pemea/
Spain	31001	Navarra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

EENA asbl
info@eena.org - www.eena.org

Annex B Fully-Meshed PEMEA Network

This annex is informative only and provides a view of what a fully-meshed PEMEA network may consist of.



EENA asbl
info@eena.org - www.eena.org

Annex C Caller Data

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

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 shall provide extension points so that further caller information can be added in a backwards compatible way.

Annex D 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 sort of "*callback URL*", that allows the originating AP to indicate that it can provide more information to the PSAP if required. The protocols and messaging specification will describe this URL in more detail, but in the first instance when accessed the URL a rendered HTML page is returned to the requesting entity and provides all of the additional information that AP has on the emergency caller.

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.