Abstract

When an emergency call is sent to a Public Safety Answering Point (PSAP), the device that sends it, as well as any application service provider in the path of the call, or access network provider through which the call originated may have information about the call, the caller or the location which the PSAP may be able to use. This document describes data structures and a mechanism to convey such data to the PSAP. The mechanism uses a Uniform Resource Identifier (URI), which may point to either an external resource or an object in the body of the SIP message. The mechanism thus allows the data to be passed by reference (when the URI points to an external resource) or by value (when it points into the body of the message). This follows the tradition of prior emergency services standardization work where data can be conveyed by value within the call signaling (i.e., in body of the SIP message) and also by reference.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 9, 2015.

# Table of Contents

1. Introduction ................................................. 4
  1.1. Data Provider Information .............................. 8
      1.1.1. Data Provider String ............................ 8
      1.1.2. Data Provider ID ................................. 9
      1.1.3. Data Provider ID Series ........................ 9
      1.1.4. Type of Data Provider ........................... 10
      1.1.5. Data Provider Contact URI ....................... 11
      1.1.6. Data Provider Languages(s) Supported .......... 11
      1.1.7. xCard of Data Provider .......................... 12
      1.1.8. Subcontractor Principal ........................ 12
      1.1.9. Subcontractor Priority .......................... 13
      1.1.10. ProviderInfo Example ........................... 13
  1.2. Service Information .................................... 16
      1.2.1. Service Environment ............................. 16
      1.2.2. Service Type ................................... 17
      1.2.3. Service Mobility Environment .................... 19
      1.2.4. EmergencyCallData.ServiceInfo Example .......... 19
  1.3. Device Information ...................................... 20
      1.3.1. Device Classification ........................... 20
      1.3.2. Device Manufacturer ............................. 22
      1.3.3. Device Model Number .............................. 22
      1.3.4. Unique Device Identifier ........................ 22
      1.3.5. Device/Service Specific Additional Data Structure .. 23
      1.3.6. Device/Service Specific Additional Data Structure Type ........................................ 24
      1.3.7. Issues with getting new types of data into use ... 24
      1.3.8. Choosing between defining a new type of block or new type of device/service specific additional data ... 25
4.3.9. EmergencyCallData.DeviceInfo Example ............... 26
4.4. Owner/Subscriber Information .......................... 26
4.4.1. Subscriber Data Privacy Indicator .................... 26
4.4.2. xCard for Subscriber’s Data ......................... 27
4.4.3. EmergencyCallData.SubscriberInfo Example ........... 27
4.5. Comment .................................................. 30
4.5.1. Comment ................................................ 30
4.5.2. EmergencyCallData.Comment Example .................. 30
5. Data Transport Mechanisms ................................ 30
5.1. Transmitting Blocks using the Call-Info Header ........ 32
5.2. Transmitting Blocks by Reference using the provided-by
     Element ................................................. 33
5.3. Transmitting Blocks by Value using the provided-by
     Element ................................................. 34
5.4. The Content-Disposition Parameter ..................... 35
6. Examples .................................................... 36
7. XML Schemas .................................................. 48
7.1. EmergencyCallData.ProviderInfo XML Schema ............ 48
7.2. EmergencyCallData.ServiceInfo XML Schema ............. 50
7.3. EmergencyCallData.DeviceInfo XML Schema ............... 51
7.4. EmergencyCallData.SubscriberInfo XML Schema .......... 53
7.5. EmergencyCallData.Comment XML Schema ................. 54
7.6. provided-by XML Schema .................................. 55
8. Security Considerations .................................... 58
9. Privacy Considerations ..................................... 59
10. IANA Considerations ....................................... 62
10.1. Registry creation ....................................... 62
10.1.1. Provider ID Series Registry ........................ 62
10.1.2. Service Environment Registry ....................... 63
10.1.3. Service Type Registry ................................ 63
10.1.4. Service Mobility Registry ........................... 64
10.1.5. Service Provider Type Registry ...................... 64
10.1.6. Service Delivered Registry ......................... 65
10.1.7. Device Classification Registry ...................... 65
10.1.8. Device ID Type Registry ............................ 65
10.1.9. Device/Service Data Type Registry .................. 66
10.1.10. Emergency Call Data Types Registry ............... 66
10.2. ‘EmergencyCallData’ Purpose Parameter Value ......... 67
10.3. URN Sub-Namespace Registration for provided-by Registry
     Entry .................................................... 68
10.4. MIME Registrations ...................................... 68
10.4.1. MIME Content-type Registration for
        ‘application/EmergencyCallData.ProviderInfo+xml’ .. 68
10.4.2. MIME Content-type Registration for
        ‘application/EmergencyCallData.ServiceInfo+xml’ .. 69
10.4.3. MIME Content-type Registration for
        ‘application/EmergencyCallData.DeviceInfo+xml’ ... 70
10.4.4. MIME Content-type Registration for
When an IP-based emergency call is initiated, a rich set of data from multiple data sources is conveyed to the Public Safety Answering Point (PSAP). This data includes information about the calling party identity, the multimedia capabilities of the device, the request for emergency services, location information, and meta-data about the sources of the data. The device, the access network provider, and any service provider in the call path may have even more information useful for a PSAP. This document extends the basic set of data communicated with an IP-based emergency call, as described in [RFC6443] and [RFC6881], in order to carry additional data which may be useful to an entity or call taker handling the call. This data is "additional" to the basic information found in the emergency call signaling used.

In general, there are three categories of this additional data that may be transmitted with an emergency call:
Data Associated with a Location: Primary location data is conveyed in the Presence Information Data Format Location Object (PIDF-LO) data structure as defined in RFC 4119 [RFC4119] and extended by RFC 5139 [RFC5139] and RFC 6848 [RFC6848] (for civic location information), RFC 5491 [RFC5491] and RFC 5962 [RFC5962] (for geodetic location information), and [RFC7035] (for relative location). This primary location data identifies the location or estimated location of the caller. However, there may exist additional, secondary data which is specific to the location, such as floor plans, tenant and building owner contact data, heating, ventilation and air conditioning (HVAC) status, etc. Such secondary location data is not included in the location data structure but can be transmitted using the mechanisms defined in this document. Although this document does not define any structures for such data, future documents may do so following the procedures defined here.

Data Associated with a Call: While some information is carried in the call setup procedure itself (as part of the SIP headers as well as in the body of the SIP message), there is additional data known by the device making the call and/or a service provider along the path of the call. This information may include the service provider contact information, subscriber identity and contact information, the type of service the service provider and the access network provider offer, what type of device is being used, etc. Some data is broadly applicable, while other data is dependent on the type of device or service. For example, a medical monitoring device may have sensor data. The data structures defined in this document (Data Provider Information, Device Information, and Owner/Subscriber Information) all fall into the category of "Data Associated with a Call".

Data Associated with a Caller: This is personal data about a caller, such as medical information and emergency contact data. Although this document does not define any structures within this category, future documents may do so following the procedures defined here.

While this document defines data structures only within the category of Data Associated with a Call, by establishing the overall framework of Additional Data, along with general mechanisms for transport of such data, extension points and procedures for future extensions, it minimizes the work needed to carry data in the other categories. Other specifications may make use of the facilities provided here.

For interoperability, there needs to be a common way for the information conveyed to a PSAP to be encoded and identified. Identification allows emergency services authorities to know during call processing which types of data are present and to determine if
they wish to access it. A common encoding allows the data to be successfully accessed.

This document defines an extensible set of data structures, and mechanisms to transmit this data either by value or by reference, either in the Session Initiation Protocol (SIP) call signaling or in the Presence Information Data Format Location Object (PIDF-LO). The data structures are usable by other communication systems and transports as well. The data structures are defined in Section 4, and the transport mechanisms (using SIP and HTTPS) are defined in Section 5.

Each data structure described in this document is encoded as a "block" of information. Each block is an XML structure with an associated Multipurpose Internet Mail Extensions (MIME) type for identification within transport such as SIP and HTTPS. The set of blocks is extensible. Registries are defined to identify the block types that may be used and to allow blocks to be included in emergency call signaling.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

This document also uses terminology from [RFC5012]. We use the term service provider to refer to an Application Service Provider (ASP). A Voice Service Provider (VSP) is a special type of ASP. With the term "Access Network Provider" we refer to the Internet Access Provider (IAP) and the Internet Service Provider (ISP) without further distinguishing these two entities, since the difference between the two is not relevant for this document. Note that the roles of ASP and access network provider may be provided by a single company. An Emergency Services Provider is an entity directly involved in providing emergency services. This includes PSAPs, dispatch, police, fire, emergency medical, other responders, and other similar agencies.

Within each data block definition (see Section 4), the values for the "Use:" label are specified as one of the following:

'Required': means it MUST be present in the data structure.

'Conditional': means it MUST be present if the specified condition(s) is met. It MAY be present if the condition(s) is not met.
Optional’: means it MAY be present.

vCard is a data format for representing and exchanging a variety of information about individuals and other entities. For applications that use XML the format defined in vCard is not immediately applicable. For this purpose an XML-based encoding of the information elements defined in the vCard specification has been defined and the name of that specification is xCard. Since the term vCard is more familiar to most readers, we use the term xCard and vCard interchangeably.

3. Document Scope

The scope of this document is explicitly limited to emergency calls. The data structures defined here are not appropriate to be conveyed with non-emergency calls because they carry sensitive and private data.

4. Data Structures

This section defines the following five data structures, each as a data block. For each block we define the MIME type, and the XML encoding. The five data structures are:

‘DataProvider’: This block supplies name and contact information for the entity that created the data. Section 4.1 provides the details.

‘Service Information’: This block supplies information about the service. The description can be found in Section 4.2.

‘Device Information’: This block supplies information about the device placing the call. Device information can be found in Section 4.3.

‘Owner/Subscriber’: This block supplies information about the owner of the device or about the subscriber. Details can be found in Section 4.4.

‘Comment’: This block provides a way to supply free form human readable text to the PSAP or emergency responders. This simple structure is defined in Section 4.5.

Each block contains a mandatory <DataProviderReference> element. The purpose of the <DataProviderReference> element is to associate all blocks added by the same data provider as a unit. The <DataProviderReference> element associates the data provider block to each of the other blocks added as a unit. Consequently, when a data
provider adds additional data to an emergency call (such as device information) it MUST add information about itself (via the data provider block) and the blocks added contain the same value in the <DataProviderReference> element. All blocks added by a single entity at the same time MUST have the same <DataProviderReference> value. The value of the <DataProviderReference> element has the same syntax and properties (specifically, world-uniqueness) as the value of the "Message-ID" message body header field specified in RFC 5322 [RFC5322] except that the <DataProviderReference> element is not enclosed in brackets (the "<" and ">" symbols are omitted). In other words, the value of a <DataProviderReference> element is syntactically a msg-id as specified in RFC 5322 [RFC5322].

Note that the xCard format is re-used in some of the data structures to provide contact information. In an xCard there is no way to specify a "main" telephone number. These numbers are useful to emergency responders who are called to a large enterprise. This document adds a new property value to the "tel" property of the TYPE parameter called "main". It can be used in any xCard in additional data.

4.1. Data Provider Information

This block is intended to be supplied by any service provider in the path of the call or the access network provider. It includes identification and contact information. This block SHOULD be supplied by every service provider in the call path, and by the access network provider. Devices MAY use this block to provide identifying information. The MIME subtype is "application/EmergencyCallData.ProviderInfo+xml". An access network provider SHOULD provide this block either by value or by reference in the provided-by section of a PIDF-LO

4.1.1. Data Provider String

Data Element: Data Provider String

Use: Required

XML Element: <DataProviderString>

Description: This is a plain text string suitable for displaying the name of the service provider that supplied the data structure. If the device creates the structure, it SHOULD use the value of the contact header in the SIP INVITE.

Reason for Need: Inform the call taker of the identity of the entity providing the data.
How Used by Call Taker: Allows the call taker to interpret the data in this structure. The source of the information often influences how the information is used, believed or verified.

4.1.2. Data Provider ID

Data Element: Data Provider ID

Use: Required. This data MUST be provided in order to uniquely identify the service provider or access provider.

XML Element: <ProviderID>

Description: A jurisdiction-specific code for, or the fully-qualified domain name of, the access network provider or service provider shown in the <DataProvidedBy> element that created the structure. NOTE: The value SHOULD be assigned by an organization appropriate for the jurisdiction. In the U.S., the provider’s NENA Company ID MUST appear here. Additional information can be found at NENA Company Identifier Program [1] or NENA Company ID [2]. The NENA Company ID MUST be in the form of a URI in the following format: urn:nena:companyid:<NENA Company ID>. The value MAY be the fully-qualified domain name of the service provider or access provider.

Reason for Need: Inform the call taker of the identity of the entity providing the data.

How Used by Call Taker: Where jurisdictions have lists of providers the Data Provider ID provides useful information about the data source. The Data Provider ID uniquely identifies the source of the data, which might be needed especially during unusual circumstances and for routine logging.

4.1.3. Data Provider ID Series

Data Element: Data Provider ID Series

Use: Required.

XML Element: <ProviderIDSeries>

Description: Identifies the issuer of the <ProviderID>. The Provider ID Series Registry (see Section 10.1) initially contains the following valid entries:

* NENA
* EENA
* domain

Reason for Need: Identifies how to interpret the Data Provider ID. The combination of ProviderIDSeries and ProviderID MUST be globally unique.

How Used by Call Taker: Determines which provider ID registry to consult for more information

4.1.4. Type of Data Provider

Data Element: Type of Data Provider

Use: Required.

XML Element: <TypeOfProvider>

Description: Identifies the type of data provider supplying the data. A registry with an initial set of values is shown in Figure 1 (see also Section 10.1).

<table>
<thead>
<tr>
<th>Token</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Network Provider</td>
<td>Access network service provider</td>
</tr>
<tr>
<td>Telecom Provider</td>
<td>Calling or origination telecom SP</td>
</tr>
<tr>
<td>Telematics Provider</td>
<td>A sensor based service provider, especially vehicle based</td>
</tr>
<tr>
<td>Language Translation Provider</td>
<td>A spoken language translation SP</td>
</tr>
<tr>
<td>Emergency Service Provider</td>
<td>An emergency service provider conveying information to another emergency service provider.</td>
</tr>
<tr>
<td>Emergency Modality Translation</td>
<td>An emergency call specific modality translation service e.g., for sign language interpreting</td>
</tr>
<tr>
<td>Relay Provider</td>
<td>A interpretation SP, for example, video relay for sign language interpreting</td>
</tr>
<tr>
<td>Other</td>
<td>Any other type of service provider</td>
</tr>
</tbody>
</table>

Figure 1: Type of Data Provider Registry.

Reason for Need: Identifies the category of data provider.
How Used by Call Taker: This information may be helpful when deciding whom to contact when further information is needed.

4.1.5. Data Provider Contact URI

Data Element: Data Provider Contact URI

Use: Required

XML Element: <ContactURI>

Description: When provided by a service provider or an access network provider, this information MUST be a URI to a 24/7 support organization tasked to provide PSAP support for this emergency call. If the call is from a device, this SHOULD be the contact information of the owner of the device. The Data Provider Contact URI SHOULD be a TEL URI [RFC3966] in E.164 format fully specified with country code. If a TEL URI is not available, it MAY be a generic SIP URI. Note that this contact information is not used by PSAPs for callbacks (a call from a PSAP directly related to a recently terminated emergency call, placed by the PSAP using a SIP Priority header field set to "psap-callback", as described in [RFC7090]).

Reason for Need: Additional data providers may need to be contacted in error cases or other unusual circumstances.

How Used by Call Taker: To contact the supplier of the additional data for assistance in handling the call.

4.1.6. Data Provider Languages(s) Supported

Data Element: Data Provider Language(s) supported

Use: Required.

XML Element: <Language>

Description: This field encodes the language used by the entity at the Data Provider Contact URI. The content of this field consists of a single token from the language tags registry, which can be found at [LanguageTagRegistry], and is defined in [RFC5646]. Multiple instances of this element may occur but the order is significant and the preferred language should appear first. The content MUST reflect the languages supported at the contact URI.

Note that the ‘language’ media feature tag, defined in RFC 3840 [RFC3840] and the more extensive language negotiation mechanism...
proposed with [I-D.gellens-slim-negotiating-human-language] are independent of this data provider language indication.

Reason for Need: This information indicates if the emergency service authority can directly communicate with the service provider or if an interpreter will be needed.

How Used by Call Taker: If the call taker cannot speak any language supported by the service provider, a translation service will need to be added to the conversation. Alternatively, other persons at the PSAP, besides the call taker, might be consulted for help (depending on the urgency and the type of interaction).

4.1.7. xCard of Data Provider

Data Element: xCard of Data Provider

Use: Optional

XML Element: <DataProviderContact>

Description: Per [RFC6351] the xcard structure is represented within a <vcard> element. Although multiple <vcard> elements may be contained in a structure only one <vcard> element SHOULD be provided. If more than one appears, the first SHOULD be used.

There are many fields in the xCard and the creator of the data structure is encouraged to provide as much information as they have available. N, ORG, ADR, TEL, EMAIL are suggested at a minimum. N SHOULD contain the name of the support group or device owner as appropriate. If more than one TEL property is provided, a parameter from the vCard Property Value registry MUST be specified on each TEL. For encoding of the xCard this specification uses the XML-based encoding specified in [RFC6351], referred to in this document as "xCard".

Reason for Need: Information needed to determine additional contact information.

How Used by Call Taker: Assists the call taker by providing additional contact information aside from what may be included in the SIP INVITE or the PIDF-LO.

4.1.8. Subcontractor Principal

When the entity providing the data is a subcontractor, the Data Provider Type is set to that of the primary service provider and this entry is supplied to provide information regarding the subcontracting entity.
Data Element:  Subcontractor Principal

Use:  Conditional.  This data is required if the entity providing the data is a subcontractor.

XML Element:  <SubcontractorPrincipal>

Description:  Some providers outsource their obligations to handle aspects of emergency services to specialized providers.  If the data provider is a subcontractor to another provider this element contains the DataProviderString of the service provider to indicate which provider the subcontractor is working for.

Reason for Need:  Identify the entity the subcontractor works for.

How Used by Call Taker:  Allows the call taker to understand what the relationship between data providers and the service providers in the path of the call are.

4.1.9.  Subcontractor Priority

Data Element:  Subcontractor Priority

Use:  Conditional.  This element is required if the Data Provider type is set to "Subcontractor".

XML Element:  <SubcontractorPriority>

Description:  If the subcontractor has to be contacted first then this element MUST have the value "sub".  If the provider the subcontractor is working for has to be contacted first then this element MUST have the value "main".

Reason for Need:  Inform the call taker whom to contact first, if support is needed.

How Used by Call Taker:  To decide which entity to contact first if assistance is needed.

4.1.10.  ProviderInfo Example

<?xml version="1.0" encoding="UTF-8"?>
<ad:EmergencyCallData.ProviderInfo
 xmlns:ad="urn:ietf:params:xml:ns:EmergencyCallData:ProviderInfo"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <ad:DataProviderReference>string0987654321@example.org</ad:DataProviderReference>
</ad:DataProviderReference>
<ad:DataProviderString>Example VoIP Provider</ad:DataProviderString>
<ad:ProviderID>urn:nena:companyid:ID123</ad:ProviderID>
<ad:ProviderIDSeries>NENA</ad:ProviderIDSeries>
<ad:TypeOfProvider>Telecom Provider</ad:TypeOfProvider>
<ad:ContactURI>tel:+1-201-555-0123</ad:ContactURI>
<ad:Language>en</ad:Language>
<ad:DataProviderContact xmlns="urn:ietf:params:xml:ns:vcard-4.0">
  <vcard>
    <fn><text>Hannes Tschofenig</text></fn>
    <n>
      <surname>Hannes</surname>
      <given>Tschofenig</given>
      <additional/>
      <prefix/>
      <suffix>Dipl. Ing.</suffix>
    </n>
    <bday><date>--0203</date></bday>
    <anniversary>
      <date-time>20090808T1430-0500</date-time>
    </anniversary>
    <gender><sex>M</sex></gender>
    <lang>
      <parameters><pref><integer>1</integer></pref>
      </parameters>
      <language-tag>de</language-tag>
    </lang>
    <lang>
      <parameters><pref><integer>2</integer></pref>
      </parameters>
      <language-tag>en</language-tag>
    </lang>
    <org>
      <parameters><type><text>work</text></type>
      </parameters>
      <text>Example VoIP Provider</text>
    </org>
    <adr>
      <parameters>
        <type><text>work</text></type>
        <label><text>Hannes Tschofenig
          Linnoitustie 6
          Espoo, Finland
          02600</text></label>
      </parameters>
      <pobox/>
      <ext/>
    </adr>
  </vcard>
</ad:DataProviderContact>
<street>Linnoitustie 6</street>
<locality>Espoo</locality>
<region>Uusimaa</region>
<code>02600</code>
<country>Finland</country>
</adr>
	<tel>
		<parameters>
			<type>
				<text>work</text>
			<text>voice</text>
		</type>
		</parameters>
		<uri>tel:+358 50 4871445</uri>
	</tel>
	<email>
		<parameters><type><text>work</text></type>
		</parameters>
		<text>hannes.tschofenig@nsn.com</text>
	</email>
	<geo>
		<parameters><type><text>work</text></type>
		</parameters>
		<uri>geo:60.210796,24.812924</uri>
	</geo>
	?key>
		<parameters><type><text>home</text></type>
		</parameters>
		<uri>http://www.tschofenig.priv.at/key.asc</uri>
	</key>
	<tz><text>Finland/Helsinki</text></tz>
	?url>
		<parameters><type><text>home</text></type>
		</parameters>
		<uri>http://www.tschofenig.priv.at</uri>
	</url>
</vcard>
</ad:DataProviderContact>
</ad:EmergencyCallData.ProviderInfo>

Figure 2: EmergencyCallData.ProviderInfo Example.
4.2. Service Information

This block describes the service that the service provider provides to the caller. It SHOULD be included by all SPs in the path of the call. The mime subtype is "application/EmergencyCallData.ServiceInfo+xml".

4.2.1. Service Environment

Data Element: Service Environment

Use: Optional when a 'ServiceType' value is 'wireless'; required otherwise.

XML Element: <ServiceEnvironment>

Description: This element defines whether a call is from a business or residence caller. Currently, the only valid entries are 'Business', 'Residence', and 'unknown'. New values can be defined via the registry created in Figure 22.

Reason for Need: To provide context and a hint when determining equipment and manpower requirements.

How Used by Call Taker: Information may be used to provide context and a hint to assist in determining equipment and manpower requirements for emergency responders. Because there are cases where the service provider does not know (such as anonymous pre-paid service), and the type of service does not necessarily reflect the nature of the premises (for example, a business line installed in a residence, or wireless service), and the registry is not all encompassing, therefore this is at best advisory information, but since it mimics a similar capability in some current emergency calling systems (e.g., a field in the Automatic Location Information (ALI) information used with legacy North American wireline systems), it is known to be valuable. The service provider uses its best information (such as a rate plan, facilities used to deliver service or service description) to determine the information and is not responsible for determining the actual characteristics of the location from which the call originated. Because the usefulness is unknown (and less clear) for wireless, this element is OPTIONAL for wireless and REQUIRED otherwise.
4.2.2. Service Type

Data Element: Service Delivered by Provider to End User

Use: Required

XML Element: <ServiceType>

Description: This defines the type of service over which the call is placed. The implied mobility of this service cannot be relied upon. A registry with an initial set of values is defined in Figure 3.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wireless</td>
<td>Wireless Telephone Service: Includes CDMA, GSM, Wi-Fi, WiMAX, LTE (but not satellite)</td>
</tr>
<tr>
<td>coin</td>
<td>Fixed public pay/coin telephones: Any coin or credit card operated device</td>
</tr>
<tr>
<td>one-way</td>
<td>One way outbound service</td>
</tr>
<tr>
<td>prison</td>
<td>Inmate call/service</td>
</tr>
<tr>
<td>temp</td>
<td>Soft dial tone/quick service/warm disconnect/suspended</td>
</tr>
<tr>
<td>MLTS-hosted</td>
<td>Hosted multi-line telephone system such as Centrex</td>
</tr>
<tr>
<td>MLTS-local</td>
<td>Local multi-line telephone system, includes all PBX, key systems, Shared Tenant Service</td>
</tr>
<tr>
<td>sensor-</td>
<td>These are devices that generate DATA ONLY. This is a one-way information transmit without interactive media</td>
</tr>
<tr>
<td>unattended</td>
<td></td>
</tr>
<tr>
<td>sensor-</td>
<td>Devices that are supported by a monitoring service provider or that are capable of supporting interactive media</td>
</tr>
<tr>
<td>attended</td>
<td></td>
</tr>
<tr>
<td>POTS</td>
<td>Wireline: Plain Old Telephone Service</td>
</tr>
<tr>
<td>VOIP</td>
<td>An over-the-top service that provides communication over arbitrary Internet access (fixed, nomadic, mobile)</td>
</tr>
<tr>
<td>remote</td>
<td>Off premise extension</td>
</tr>
<tr>
<td>relay</td>
<td>A service where there is a human third party agent who provides additional assistance. This includes sign language relay and telematics services that provide a human on the call.</td>
</tr>
</tbody>
</table>

Figure 3: Service Delivered by Provider to End User Registry.

More than one value MAY be returned. For example, a VoIP inmate telephone service is a reasonable combination.

Reason for Need: Knowing the type of service may assist the PSAP with the handling of the call.
How Used by Call Taker: Call takers often use this information to
determine what kinds of questions to ask callers, and how much to
rely on supportive information. An emergency call from a prison
is treated differently than a call from a sensor device. As the
information is not always available, and the registry is not all
encompassing, this is at best advisory information, but since it
mimics a similar capability in some current emergency calling
systems, it is known to be valuable.

4.2.3. Service Mobility Environment

Data Element: Service Mobility Environment

Use: Required

XML Element: <ServiceMobility>

Description: This provides the service provider’s view of the
mobility of the caller’s device. As the service provider may not
know the characteristics of the actual device or access network
used, the value MUST NOT be relied upon. The registry specified
in Figure 23 reflects the following initial valid entries:

* Mobile: the device is able to move at any time

* Fixed: the device is not expected to move unless the service is
  relocated

* Nomadic: the device is not expected to change its point of
  attachment while on a call

* Unknown: no information is known about the service mobility
  environment for the device

Reason for Need: Knowing the service provider’s belief of mobility
may assist the PSAP with the handling of the call.

How Used by Call Taker: To determine whether to assume the location
of the caller might change.

4.2.4. EmergencyCallData.ServiceInfo Example
4.3. Device Information

This block provides information about the device used to place the call. It should be provided by any service provider that knows what device is being used, and by the device itself. The mime subtype is "application/EmergencyCallData.DeviceInfo+xml".

4.3.1. Device Classification

Data Element: Device Classification

Use: Optional

XML Element: <DeviceClassification>

Description: This data element defines the kind of device making the emergency call. If the device provides the data structure, the device information SHOULD be provided. If the service provider provides the structure and it knows what the device is, the service provider SHOULD provide the device information. Often the carrier does not know what the device is. It is possible to receive two Additional Data Associated with a Call data structures, one created by the device and one created by the service provider. This information describes the device, not how it is being used. This data element defines the kind of device making the emergency call. The registry with the initial set of values is shown in Figure 5.
<table>
<thead>
<tr>
<th>Token</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cordless</td>
<td>Cordless handset</td>
</tr>
<tr>
<td>fixed</td>
<td>Fixed phone</td>
</tr>
<tr>
<td>satellite</td>
<td>Satellite phone</td>
</tr>
<tr>
<td>sensor-fixed</td>
<td>Fixed (non mobile) sensor/alarm device</td>
</tr>
<tr>
<td>desktop</td>
<td>Soft client on desktop PC</td>
</tr>
<tr>
<td>laptop</td>
<td>Soft client on laptop type device</td>
</tr>
<tr>
<td>tablet</td>
<td>Soft client on tablet type device</td>
</tr>
<tr>
<td>alarm-monitored</td>
<td>Alarm system</td>
</tr>
<tr>
<td>sensor-mobile</td>
<td>Mobile sensor device</td>
</tr>
<tr>
<td>aircraft</td>
<td>Aircraft telematics device</td>
</tr>
<tr>
<td>automobile</td>
<td>Automobile/cycle/off-road telematics</td>
</tr>
<tr>
<td>truck</td>
<td>Truck/construction telematics</td>
</tr>
<tr>
<td>farm</td>
<td>Farm equipment telematics</td>
</tr>
<tr>
<td>marine</td>
<td>Marine telematics</td>
</tr>
<tr>
<td>personal</td>
<td>Personal telematics device</td>
</tr>
<tr>
<td>feature-phone</td>
<td>Feature- (not smart-) cellular phone</td>
</tr>
<tr>
<td>smart-phone</td>
<td>Smart-phone cellular phone (native)</td>
</tr>
<tr>
<td>smart-phone-app</td>
<td>Soft client app on smart-phone</td>
</tr>
<tr>
<td>unknown-device</td>
<td>Soft client on unknown device type</td>
</tr>
<tr>
<td>game</td>
<td>Gaming console</td>
</tr>
<tr>
<td>text-only</td>
<td>Other text device</td>
</tr>
<tr>
<td>NA</td>
<td>Not Available</td>
</tr>
</tbody>
</table>

Figure 5: Device Classification Registry.

Reason for Need: The device classification implies the capability of the calling device and assists in identifying the meaning of the emergency call location information that is being presented. For example, does the device require human intervention to initiate a call or is this call the result of programmed instructions? Does the calling device have the ability to update location or condition changes? Is this device interactive or a one-way reporting device?

How Used by Call Taker: May provide the call taker context regarding the caller, the capabilities of the calling device or the environment in which the device is being used, and may assist in understanding the location information and capabilities of the calling device. For example, a cordless handset may be outside or next door.
4.3.2. Device Manufacturer

Data Element: Device Manufacturer

Use: Optional

XML Element: <DeviceMfgr>

Description: The plain language name of the manufacturer of the device.

Reason for Need: Used by PSAP management for post-mortem investigation/resolution.

How Used by Call Taker: Probably not used by the calltaker, but by PSAP management.

4.3.3. Device Model Number

Data Element: Device Model Number

Use: Optional

XML Element: <DeviceModelNr>

Description: Model number of the device.

Reason for Need: Used by PSAP management for after action investigation/resolution.

How Used by Call Taker: Probably not used by the calltaker, but by PSAP management.

4.3.4. Unique Device Identifier

Data Element: Unique Device Identifier

Use: Optional

XML Element: <UniqueDeviceID>

XML Attribute: <TypeOfDeviceID>

Description: A string that identifies the specific device (or the device’s current SIM) making the call or creating an event. Note that more than one <UniqueDeviceID> may be present, to supply more than one of the identifying values.
The `<TypeOfDeviceID>` attribute identifies the type of device identifier. A registry with an initial set of values can be seen in Figure 6.

<table>
<thead>
<tr>
<th>Token</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEID</td>
<td>Mobile Equipment Identifier (CDMA)</td>
</tr>
<tr>
<td>ESN</td>
<td>Electronic Serial Number (GSM)</td>
</tr>
<tr>
<td>MAC</td>
<td>Media Access Control Address (IEEE)</td>
</tr>
<tr>
<td>WiMAX</td>
<td>Device Certificate Unique ID</td>
</tr>
<tr>
<td>IMEI</td>
<td>International Mobile Equipment ID (GSM)</td>
</tr>
<tr>
<td>IMSI</td>
<td>International Mobile Subscriber ID (GSM)</td>
</tr>
<tr>
<td>UDI</td>
<td>Unique Device Identifier</td>
</tr>
<tr>
<td>RFID</td>
<td>Radio Frequency Identification</td>
</tr>
<tr>
<td>SN</td>
<td>Manufacturer Serial Number</td>
</tr>
</tbody>
</table>

Figure 6: Registry with Device Identifier Types.

Reason for Need: Uniquely identifies the device (or, in the case of IMSI, a SIM), independent of any signaling identifiers present in the call signaling stream.

How Used by Call Taker: Probably not used by the call taker; may be used by PSAP management during an investigation.

Example: `<UniqueDeviceID TypeOfDeviceID="SN">12345</UniqueDeviceID>`

4.3.5. Device/Service Specific Additional Data Structure

Data Element: Device/service specific additional data structure

Use: Optional

XML Element: `<DeviceSpecificData>`

Description: A URI representing additional data whose schema is specific to the device or service which created it. (For example, a medical device or medical device monitoring service may have a defined set of medical data). The URI, when dereferenced, MUST yield a data structure defined by the Device/service specific additional data type value. Different data may be created by each classification; e.g., a medical device created data set.

Reason for Need: Provides device/service specific data that may be used by the call taker and/or responders.
How Used by Call Taker: Provide information to guide call takers to select appropriate responders, give appropriate pre-arrival instructions to callers, and advise responders of what to be prepared for. May be used by responders to guide assistance provided.

4.3.6. Device/Service Specific Additional Data Structure Type

Data Element: Type of device/service specific additional data structure

Use: Conditional. MUST be provided when device/service specific additional URI is provided

XML Element: <DeviceSpecificType>

Description: Value from a registry defined by this document to describe the type of data that can be retrieved from the device/service specific additional data structure. Initial values are:

* IEEE 1512

IEEE 1512 is the USDot model for traffic incidents.

Reason for Need: This data element allows identification of externally defined schemas, which may have additional data that may assist in emergency response.

How Used by Call Taker: This data element allows the end user (calltaker or first responder) to know what type of additional data may be available to aid in providing the needed emergency services.

Note: Information which is specific to a location or a caller (person) should not be placed in this section.

4.3.7. Issues with getting new types of data into use

This document describes two mechanisms which allow extension of the kind of data provided with an emergency call: define a new block or define a new service specific additional data URL for the DeviceInfo block. While defining new data types and getting a new device or application to send the new data may be easy, getting PSAPs and responders to actually retrieve the data and use it will be difficult. New mechanism providers should understand that acquiring and using new forms of data usually require software upgrades at the PSAP and/or responders, as well as training of call takers and responders in how to interpret and use the information. Legal and
operational review may also be needed. Overwhelming a call taker or responder with too much information is highly discouraged. Thus, the barrier to supporting new data is quite high.

The mechanisms this document describes are meant to encourage development of widely supported, common data formats for classes of devices. If all manufacturers of a class of device use the same format, and the data can be shown to improve outcomes, then PSAPs and responders may be encouraged to upgrade their systems and train their staff to use the data. Variations, however well intentioned, are unlikely to be supported.

Implementers should consider that data from sensor-based devices in some cases may not be useful to call takers or PSAPs (and privacy or other considerations may preclude the PSAP from touching the data), but may be of use to responders. Some standards being developed by other organizations to carry data from the PSAP to responders are designed to carry all additional data supplied in the call that conform to this document, even if the PSAP does not fetch or interpret the data. This allows responders to get the data even if the PSAP does not.

4.3.8. Choosing between defining a new type of block or new type of device/service specific additional data

For devices that have device or service specific data, there are two choices to carry it. A new block can be defined, or the device/service specific additional data URL the DeviceInfo block can be used and a new type for it defined. The data passed would likely be the same in both cases. Considerations for choosing which mechanism to register under include:

Applicability: Information which will be carried by many kinds of devices or services are more appropriately defined as separate blocks.

Privacy: Information which may contain private data may be better sent in the DeviceInfo block, rather than a new block so that implementations are not tempted to send the data by value, and thus having more exposure to the data than forcing the data to be retrieved via the URL in DeviceInfo.

Size: Information which may be very large may be better sent in the DeviceInfo block, rather than a new block so that implementations are not tempted to send the data by value. Conversely, data which is small may best be sent in a separate block so that it can be sent by value.
Availability of a server: Providing the data via the device block requires a server be made available to retrieve the data. Providing the data via new block allows it to be sent by value (CID).

4.3.9. EmergencyCallData.DeviceInfo Example

```xml
<?xml version="1.0" encoding="UTF-8"?>
<dev:EmergencyCallData.DeviceInfo
    xmlns:dev="urn:ietf:params:xml:ns:EmergencyCallData:DeviceInfo"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <dev:DataProviderReference>d4b3072df.201409182208075@example.org</dev:DataProviderReference>
  <dev:DeviceClassification>fixed</dev:DeviceClassification>
  <dev:DeviceMfgr>Nokia</dev:DeviceMfgr>
  <dev:DeviceModelNr>Lumia 800</dev:DeviceModelNr>
  <dev:UniqueDeviceID TypeOfDeviceID="IMEI">35788104</dev:UniqueDeviceID>
</dev:EmergencyCallData.DeviceInfo>
```

Figure 7: EmergencyCallData.DeviceInfo Example.

4.4. Owner/Subscriber Information

This block describes the owner of the device (if provided by the device) or the subscriber information (if provided by a service provider). The contact location is not necessarily the location of the caller or incident, but is rather the nominal contact address. The MIME type is "application/EmergencyCallData.SubscriberInfo+xml".

In some jurisdictions some or all parts of the subscriber-specific information are subject to privacy constraints. These constraints vary but dictate what information can be displayed and logged. A general privacy indicator expressing a desire for privacy is provided. The interpretation of how this is applied is left to the receiving jurisdiction as the custodians of the local regulatory requirements.

4.4.1. Subscriber Data Privacy Indicator

Attribute: privacyRequested, boolean.

Use: Conditional. This attribute MUST be provided if the owner/subscriber information block is not empty.

Description: The subscriber data privacy indicator specifically expresses the subscriber’s desire for privacy. In some
jurisdictions subscriber services can have a specific "Type of Service" which prohibits information, such as the name of the subscriber, from being displayed. This attribute should be used to explicitly indicate whether the subscriber service includes such constraints.

Reason for Need: Some jurisdictions require subscriber privacy to be observed when processing emergency calls.

How Used by Call Taker: Where privacy is indicated the call taker may not have access to some aspects of the subscriber information.

4.4.2. xCard for Subscriber’s Data

Data Element: xCARD for Subscriber’s Data

Use: Conditional. Subscriber data is provided unless it is not available. Some services, for example prepaid phones, non-initialized phones, etc., do not have information about the subscriber.

XML Element: <SubscriberData>

Description: Information known by the service provider or device about the subscriber; e.g., Name, Address, Individual Telephone Number, Main Telephone Number and any other data. N, ORG (if appropriate), ADR, TEL, EMAIL are suggested at a minimum. If more than one TEL property is provided, a parameter from the vCard Property Value registry MUST be specified on each TEL.

Reason for Need: When the caller is unable to provide information, this data may be used to obtain it

How Used by Call Taker: Obtaining critical information about the caller and possibly the location when it is not able to be obtained otherwise.

4.4.3. EmergencyCallData.SubscriberInfo Example

<?xml version="1.0" encoding="UTF-8"?>
<sub:EmergencyCallData.SubscriberInfo
 xmlns:sub="urn:ietf:params:xml:ns:EmergencyCallData:SubscriberInfo"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 privacyRequested="false">
 <sub:DataProviderReference>FEABFEC901@example.org</sub:DataProviderReference>
</sub:EmergencyCallData.SubscriberInfo>
<sub:SubscriberData xmlns="urn:ietf:params:xml:ns:vcard-4.0">
  <vcard>
    <fn><text>Simon Perreault</text></fn>
    <n>
      <surname>Perreault</surname>
      <given>Simon</given>
      <additional/>
      <prefix/>
      <suffix>ing. jr</suffix>
      <suffix>M.Sc.</suffix>
    </n>
    <bday><date>--0203</date></bday>
    <anniversary>
      <date-time>20090808T1430-0500</date-time>
    </anniversary>
    <gender><sex>M</sex></gender>
    <lang>
      <parameters><pref><integer>1</integer></pref></parameters>
      <language-tag>fr</language-tag>
    </lang>
    <lang>
      <parameters><pref><integer>2</integer></pref></parameters>
      <language-tag>en</language-tag>
    </lang>
    <org>
      <type><text>work</text></type>
      <text>Viagenie</text>
    </org>
    <adr>
      <type><text>work</text></type>
      <label><text>Simon Perreault
        2875 boul. Laurier, suite D2-630
        Quebec, QC, Canada
        G1V 2M2</text></label>
    </adr>
    <tel>
<parameters>
  <type>
    <text>work</text>
    <text>voice</text>
  </type>
</parameters>
<uri>tel:+1-418-656-9254;ext=102</uri>
</tel>
<tel>
  <parameters>
    <type>
      <text>work</text>
      <text>text</text>
      <text>voice</text>
      <text>cell</text>
      <text>video</text>
    </type>
  </parameters>
  <uri>tel:+1-418-262-6501</uri>
</tel>
<email>
  <parameters><type><text>work</text></type>
  <text>simon.perreault@viagenie.ca</text>
</parameters>
</email>
<geo>
  <parameters><type><text>work</text></type>
  <uri>geo:46.766336,-71.28955</uri>
</geo>
?key>
  <parameters><type><text>work</text></type>
  <uri>http://www.viagenie.ca/simon.perreault/simon.asc</uri>
</parameters>
</key>
<tz><text>America/Montreal</text></tz>
<url>
  <parameters><type><text>home</text></type>
  <uri>http://nomis80.org</uri>
</parameters>
</url>
</vcard>
</sub:SubscriberData>
</sub:EmergencyCallData.SubscriberInfo>

Figure 8: EmergencyCallData.SubscriberInfo Example.
4.5. Comment

This block provides a mechanism for the data provider to supply extra, human readable information to the PSAP. It is not intended for a general purpose extension mechanism nor does it aim to provide machine-readable content. The mime subtype is "application/EmergencyCallData.Comment+xml"

4.5.1. Comment

Data Element: EmergencyCallData.Comment

Use: Optional

XML Element: <Comment>

Description: Human readable text providing additional information to the PSAP staff.

Reason for Need: Explanatory information for values in the data structure.

How Used by Call Taker: To interpret the data provided.

4.5.2. EmergencyCallData.Comment Example

```xml
<?xml version="1.0" encoding="UTF-8"?>
<com:EmergencyCallData.Comment
   xmlns:com="urn:ietf:params:xml:ns:EmergencyCallData:Comment"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <com:DataProviderReference>string0987654321@example.org</com:DataProviderReference>
   <com:Comment xml:lang="en">This is an example text.</com:Comment>
</com:EmergencyCallData.Comment>
```

Figure 9: EmergencyCallData.Comment Example.

5. Data Transport Mechanisms

This section defines how to convey additional data to an emergency service provider. Two different means are specified: the first uses the call signaling; the second uses the <provided-by> element of a PIDF-LO [RFC4119].

1. First, the ability to embed a Uniform Resource Identifier (URI) in an existing SIP header field, the Call-Info header, is defined. The URI points to the additional data structure. The
Call-Info header is specified in Section 20.9 of [RFC3261]. This document adds a new compound token starting with the value ‘EmergencyCallData’ for the Call-Info "purpose" parameter. If the "purpose" parameter is set to a value starting with ‘EmergencyCallData’, then the Call-Info header contains either an HTTPS URL pointing to an external resource or a CID (content indirection) URI that allows the data structure to be placed in the body of the SIP message. The "purpose" parameter also indicates the kind of data (by its MIME type) that is available at the URI. As the data is conveyed using a URI in the SIP signaling, the data itself may reside on an external resource, or may be contained within the body of the SIP message. When the URI refers to data at an external resource, the data is said to be passed by reference. When the URI refers to data contained within the body of the SIP message, the data is said to be passed by value. A PSAP or emergency responder is able to examine the type of data provided and selectively inspect the data it is interested in, while forwarding all of it (the values or references) to downstream entities. To be conveyed in a SIP body, additional data about a call is defined as a series of MIME objects. Each block defined in this document is an XML data structure identified by its MIME type. (Blocks defined by others may be encoded in XML or not, as identified by their MIME registration.) As usual, whenever more than one MIME part is included in the body of a message, MIME-multipart (i.e., ‘multipart/mixed’) encloses them all. This document defines a set of XML schemas and MIME types used for each block defined here. When additional data is passed by value in the SIP signaling, each CID URL points to one block in the body. Multiple URIs are used within a Call-Info header field (or multiple Call-Info header fields) to point to multiple blocks. When additional data is provided by reference (in SIP signaling or provided-by), each HTTPS URL references one block; the data is retrieved with an HTTPS GET operation, which returns one of the blocks as an object (the blocks defined here are returned as XML objects).

2. Second, the ability to embed additional data structures in the <provided-by> element of a PIDF-LO [RFC4119] is defined. In addition to service providers in the call path, the access network provider may also have similar information that may be valuable to the PSAP. The access network provider MAY provide location in the form of a PIDF-LO from a location server via a location configuration protocol. The data structures described in this document are not specific to the location itself, but rather provides descriptive information having to do with the immediate circumstances about the provision of the location (who the access network is, how to contact that entity, what kind of
service the access network provides, subscriber information, etc.). This data is similar in nearly every respect to the data known by service providers in the path of the call. When the access network provider and service provider are separate entities, the access network does not participate in the application layer signaling (and hence cannot add a Call-Info header field to the SIP message), but may provide location information in a PIDF-LO object to assist in locating the caller’s device. The <provided-by> element of the PIDF-LO is a mechanism for the access network provider to supply the information about the entity or organization that supplied this location information. For this reason, this document describes a namespace per RFC 4119 for inclusion in the <provided-by> element of a PIDF-LO for adding information known to the access network provider. The access network provider SHOULD provide additional data within a <provided-by> element of a PIDF-LO it returns for emergency use (e.g., if requested with a HELD "responseTime" attribute of "emergencyRouting" or "emergencyDispatch" [RFC5985]).

One or more blocks of data registered in the Emergency Call Additional Data registry, as defined in Section 10.1, may be included or referenced in the SIP signaling (using the Call-Info header field) or in the <provided-by> element of a PIDF-LO. Every block must be one of the types in the registry. Since the data of an emergency call may come from multiple sources, the data itself needs information describing the source. Consequently, each entity adding additional data MUST supply the "Data Provider" block. All other blocks are optional, but each entity SHOULD supply any blocks where it has at least some of the information in the block.

5.1. Transmitting Blocks using the Call-Info Header

A URI to a block MAY be inserted in any SIP request or response method (most often INVITE or MESSAGE) with a Call-Info header field containing a purpose value starting with ‘EmergencyCallData’ and the type of data available at the URI. The type of data is denoted by including the root of the MIME type (not including the ‘EmergencyCallData’ prefix and any suffix such as ‘+xml’) with a ‘.’ separator. For example, when referencing a block with MIME type ‘application/EmergencyCallData.ProviderInfo+xml’, the ‘purpose’ parameter is set to ‘EmergencyCallData.ProviderInfo’. An example "Call-Info" header field for this would be:

```
Call-Info: https://www.example.com/23sedde3; purpose="EmergencyCallData.ProviderInfo"
```
A Call-info header with a purpose value starting with ‘EmergencyCallData’ MUST only be sent on an emergency call, which can be ascertained by the presence of an emergency service urn in a Route header of a SIP message.

If the data is provided by reference, an HTTPS URI MUST be included and consequently Transport Layer Security (TLS) protection is applied for protecting the retrieval of the information.

The data may also be supplied by value in any SIP request or response method that is permitted to contain a body (i.e., not a BYE request). In this case, Content Indirection (CID) [RFC2392] is used, with the CID URL referencing the MIME body part containing the data.

Transmitting data by value is especially useful in certain cases, such as when the data exists in or is generated by the originating device, but is not intended for very large data blocks. Additional security and privacy considerations apply to data transmitted by value, as discussed in Section 8 and Section 9.

More than one Call-Info header with a purpose value starting with ‘EmergencyCallData’ can be expected, but at least one MUST be provided. The device MUST provide one if it knows no service provider is in the path of the call. The device MAY insert one if it uses a service provider. Any service provider in the path of the call MUST insert its own. For example, a device, a telematics service provider in the call path, as well as the mobile carrier handling the call will each provide one. There may be circumstances where there is a service provider who is unaware that the call is an emergency call and cannot reasonably be expected to determine that it is an emergency call. In that case, that service provider is not expected to provide EmergencyCallData.

5.2. Transmitting Blocks by Reference using the provided-by Element

The <EmergencyCallDataReference> element is used to transmit an additional data block by reference within a <provided-by> element of a PIDF-LO. The <EmergencyCallDataReference> element has two attributes: ‘ref’ to specify the URL, and ‘purpose’ to indicate the type of data block referenced. The value of ‘ref’ is an HTTPS URL that resolves to a data structure with information about the call. The value of ‘purpose’ is the same as used in a ‘Call-Info’ header field (as specified in Section 5.1).

For example, to reference a block with MIME type ‘application/EmergencyCallData.ProviderInfo+xml’, the ‘purpose’ parameter is set to ‘EmergencyCallData.ProviderInfo’. An example ‘EmergencyCallDataReference’ element for this would be:
The ‘EmergencyCallDataReference’ element transmits one additional data block; multiple additional data blocks may be transmitted by using multiple ‘EmergencyCallDataReference’ elements.

For example:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<gp:provided-by
 xmlns:gp="urn:ietf:params:xml:ns:geopriv10"
 xmlns="urn:ietf:params:xml:ns:EmergencyCallData">
    <EmergencyCallDataReference
        purpose="EmergencyCallData.ServiceInfo"
        ref="https://example.com/ref2"/>
    <EmergencyCallDataReference
        purpose="EmergencyCallData.ProviderInfo"
        ref="https://example.com/ref3"/>
    <EmergencyCallDataReference
        purpose="EmergencyCallData.Comment"
        ref="https://example.com/ref4"/>
</gp:provided-by>
```

Example provided-by by Reference.

5.3. Transmitting Blocks by Value using the provided-by Element

It is RECOMMENDED that access networks supply the data specified in this document by reference, but they MAY provide the data by value.

The <EmergencyCallDataValue> element is used to transmit one or more additional data blocks by value within a <provided-by> element of a PIDF-LO. Each block being transmitted is placed (as a child element) inside the <EmergencyCallDataValue> element. (The same XML structure as would be contained in the corresponding MIME type body part is placed inside the <EmergencyCallDataValue> element.)

For example:
Example provided-by by Value.

5.4. The Content-Disposition Parameter

RFC 5621 [RFC5621] discusses the handling of message bodies in SIP. It updates and clarifies handling originally defined in RFC 3261 [RFC3261] based on implementation experience. While RFC 3261 did not mandate support for ‘multipart’ message bodies, ‘multipart/mixed’ MIME bodies are used by many extensions (including this document) today. For example, adding a PIDF-LO, SDP, and additional data in body of a SIP message requires a ‘multipart’ message body.
RFC 3204 [RFC3204] and RFC 3459 [RFC3459] define the 'handling' parameter for the Content-Disposition header field. These RFCs describe how a UAS reacts if it receives a message body whose content type or disposition type it does not understand. If the 'handling' parameter has the value "optional", the UAS ignores the message body. If the 'handling' parameter has the value "required", the UAS returns a 415 (Unsupported Media Type) response. The 'by-reference' disposition type allows a SIP message to contain a reference to the body part, and the SIP UA processes the body part according to the reference. This is the case for the Call-info header containing a Content Indirection (CID) URL.

As an example, a SIP message indicates the Content-Disposition parameter in the body of the SIP message as shown in Figure 10.

```
Content-Type: application/sdp
...Omit Content-Disposition here; defaults are ok
...SDP goes in here
--boundary1
Content-Type: application/pidf+xml
Content-ID: <target123@atlanta.example.com>
Content-Disposition: by-reference; handling=optional
...PIDF-LO goes in here
--boundary1--

Content-Type: application/EmergencyCallData.ProviderInfo+xml
Content-ID: <1234567890@atlanta.example.com>
Content-Disposition: by-reference; handling=optional
...Data provider information data goes in here
--boundary1--
```

Figure 10: Example for use of the Content-Disposition Parameter in SIP.

6. Examples

This section illustrates a longer and more complex example, as shown in Figure 11. In this example additional data is added by the end
device, included by the VoIP provider (via the PIDF-LO), and provided by the access network provider.

--- Emergency Call Setup Procedure
... Location Retrieval/Response

Figure 11: Additional Data Example Flow

The example scenario starts with the end device itself adding device information, owner/subscriber information, a location URI, and data provider information to the outgoing emergency call setup message (see step #1 in Figure 11). The SIP INVITE example is shown in Figure 12.

INVITE urn:service:sos SIP/2.0
Via: SIPS/2.0/TLS server.example.com;branch=z9hG4bK74bf9
Max-Forwards: 70
To: <urn:service:sos>
From: Hannes Tschofenig <sips:hannes@example.com>;tag=9fxced76sl
Call-ID: 3848276298220188511@example.com
Call-Info: <http://www.example.com/hannes/photo.jpg>
  ;purpose=icon,
  <http://www.example.com/hannes/> ;purpose=info,
  <cid:1234567890@atlanta.example.com>
  ;purpose=EmergencyCallData.ProviderInfo,
  <cid:0123456789@atlanta.example.com>
  ;purpose=EmergencyCallData.DeviceInfo
Geolocation: <https://ls.example.net:9768/357yc6s64ceyluy5ax3o>
Geolocation-Routing: yes
Accept: application/sdp, application/pidf+xml,
  application/EmergencyCallData.ProviderInfo+xml
CSeq: 31862 INVITE
Contact: <sip:hannes@example.com>
Content-Type: multipart/mixed; boundary=boundary1

--boundary1

Content-Type: application/sdp

...SDP goes here

--boundary1--

Content-Type: application/EmergencyCallData.DeviceInfo+xml
Content-ID: <0123456789@atlanta.example.com>
Content-Disposition: by-reference;handling=optional
<?xml version="1.0" encoding="UTF-8"?>
<dev:EmergencyCallData.DeviceInfo
  xmlns:dev="urn:ietf:params:xml:ns:EmergencyCallData:DeviceInfo"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <dev:DataProviderReference>d4b3072df09876543@[93.184.216.119]
  </dev:DataProviderReference>
  <dev:DeviceClassification>laptop</dev:DeviceClassification>
  <dev:UniqueDeviceID
    TypeOfDeviceID="MAC">00-0d-4b-30-72-df</dev:UniqueDeviceID>
</dev:EmergencyCallData.DeviceInfo>

--boundary1--

Content-Type: application/EmergencyCallData.ProviderInfo+xml
Content-ID: <1234567890@atlanta.example.com>
Content-Disposition: by-reference;handling=optional
<?xml version="1.0" encoding="UTF-8"?>
<pi:EmergencyCallData.ProviderInfo

<pi:DataProviderReference>d4b3072df09876543@[93.184.216.119]</pi:DataProviderReference>
<pi:DataProviderString>Hannes Tschofenig</pi:DataProviderString>
<pi:TypeOfProvider>Other</pi:TypeOfProvider>
<pi:ContactURI>tel:+1-555-555-0123</pi:ContactURI>
<pi:Language>en</pi:Language>
<pi:DataProviderContact
xmlns:xc="urn:ietf:params:xml:ns:vcard-4.0">
  <vcard>
    <fn><text>Hannes Tschofenig</text></fn>
    <n>
      <surname>Hannes</surname>
      <given>Tschofenig</given>
      <additional/>
      <prefix/>
      <suffix>Dipl. Ing.</suffix>
    </n>
    <bday><date>--0203</date></bday>
    <anniversary>
      <date-time>20090808T1430-0500</date-time>
    </anniversary>
    <gender><sex>M</sex></gender>
    <lang>
      <parameters><pref><integer>1</integer></pref>
      <language-tag>de</language-tag>
    </lang>
    <lang>
      <parameters><pref><integer>2</integer></pref>
      <language-tag>en</language-tag>
    </lang>
    <adr>
      <type><text>work</text></type>
      <label><text>Hannes Tschofenig
          Linnoitustie 6
          Espoo, Finland
          02600</text></label>
    </adr>
  </vcard>
</pi:DataProviderContact>
Figure 12: End Device sending SIP INVITE with Additional Data.

In this example, information available to the access network provider is included in the call setup message only indirectly via the use of the location reference. The PSAP has to retrieve it via a separate look-up step. Since the access network provider and the VoIP service provider are two independent entities in this scenario, the access
network provider is not involved in application layer exchanges; the SIP INVITE transits the access network transparently, as illustrated in steps #1 and #2. The access network does not alter the SIP INVITE.

The VoIP service provider receives the message and determines based on the Service URN that the incoming request is an emergency call. It performs typical emergency services related tasks, including location-based routing, and adds additional data, namely service and subscriber information as well as data provider information #2, to the outgoing message. For the example we assume a VoIP service provider that deploys a back-to-back user agent allowing additional data to be included in the body of the SIP message (rather than per reference in the header), which allows us to illustrate the use of multiple data provider info blocks. The resulting message is shown in Figure 13. The SIP INVITE is sent to the PSAP in step #3.

```
INVITE sips:psap@example.org SIP/2.0
Via: SIPS/2.0/TLS server.example.com;branch=z9hG4bK74bf9
Max-Forwards: 70
To: <urn:service:sos>
From: Hannes Tschofenig <sips:hannes@example.com>;tag=9fxced76sl
Call-ID: 3848276298220188511@example.com
Call-Info: <http://www.example.com/hannes/photo.jpg>;purpose=icon,
          <http://www.example.com/hannes/>;purpose=info,
          <cid:1234567890@atlanta.example.com>
          ;purpose=EmergencyCallData.ProviderInfo
          <cid:0123456789@atlanta.example.com>
          ;purpose=EmergencyCallData.DeviceInfo
Call-Info: <cid:bloorpyhex@atlanta.example.com>
          ;purpose=EmergencyCallData.ServiceInfo
Call-Info: <cid:aaabbb@atlanta.example.com>
          ;purpose=EmergencyCallData.ProviderInfo
Geolocation: <https://ls.example.net:9768/357yc6s64ceyoiluy5ax3o>
Geolocation-Routing: yes
Accept: application/sdp, application/pidf+xml,
          application/EmergencyCallData.ProviderInfo+xml
CSeq: 31862 INVITE
Contact: <sips:hannes@example.com>
Content-Type: multipart/mixed; boundary=boundary1

Content-Length: ...
--boundary1

Content-Type: application/sdp
```
Content-Type: application/EmergencyCallData.DeviceInfo+xml
Content-ID: <0123456789@atlanta.example.com>
Content-Disposition: by-reference; handling=optional
<?xml version="1.0" encoding="UTF-8"?>
<dev:EmergencyCallData.DeviceInfo
    xmlns:dev="urn:ietf:params:xml:ns:EmergencyCallData:DeviceInfo"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <dev:DataProviderReference>d4b3072df09876543@[93.184.216.119]
    <dev:DeviceClassification>laptop</dev:DeviceClassification>
    <dev:UniqueDeviceID TypeOfDeviceID="MAC">00-0d-4b-30-72-df</dev:UniqueDeviceID>
</dev:EmergencyCallData.DeviceInfo>

Content-Type: application/EmergencyCallData.ProviderInfo+xml
Content-ID: <1234567890@atlanta.example.com>
Content-Disposition: by-reference; handling=optional
<?xml version="1.0" encoding="UTF-8"?>
<pi:EmergencyCallData.ProviderInfo
    xmlns:pi="urn:ietf:params:xml:ns:EmergencyCallData:ProviderInfo"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <pi:DataProviderReference>d4b3072df09876543@[93.184.216.119]
    <pi:DataProviderString>Hannes Tschofenig</pi:DataProviderString>
    <pi:TypeOfProvider>Other</pi:TypeOfProvider>
    <pi:ContactURI>tel:+1-555-555-0123</pi:ContactURI>
    <pi:Language>en</pi:Language>
    <pi:DataProviderContact
        xmlns:xc="urn:ietf:params:xml:ns:vcard-4.0">
        <vcard>
            <fn>Hannes Tschofenig</fn>
            <n>
                <surname>Hannes</surname>
                <given>Tschofenig</given>
                <prefix/>
                <suffix>Dipl. Ing.</suffix>
            </n>
            <bday>--0203</bday>
        </vcard>
    </pi:DataProviderContact>
</pi:EmergencyCallData.ProviderInfo>
<parameters><pref><integer>1</integer></pref></parameters>

<language-tag>de</language-tag>

<parameters><pref><integer>2</integer></pref></parameters>

<language-tag>en</language-tag>

<adr>
  <parameters>
    <type><text>work</text></type>
    <label>Hannes Tschofenig
      Linnoitustie 6
      Espoo, Finland
      02600</label>
  </parameters>

  <pobox/>

  <street>Linnoitustie 6</street>
  <locality>Espoo</locality>
  <region>Uusimaa</region>
  <code>02600</code>
  <country>Finland</country>
</adr>

<tel>
  <parameters>
    <type>
      <text>work</text>
      <text>voice</text>
    </type>
  </parameters>
  <uri>tel:+358 50 4871445</uri>
</tel>

<email>
  <parameters><type><text>work</text></type>
    <text>hannes.tschofenig@nsn.com</text>
  </parameters>
</email>

<geo>
  <parameters><type><text>work</text></type>
    <uri>geo:60.210796,24.812924</uri>
  </parameters>
</geo>
<key>
  <parameters>
    <type><text>home</text></type>
  </parameters>
</key>
<uri>https://www.example.com/key.asc</uri>
</key>
<tz><text>Finland/Helsinki</text></tz>
?url>
  <parameters><type><text>home</text></type>
  </parameters>
  <uri>http://example.com/hannes.tschofenig</uri>
</url>
</vcard>
</vcard>
</pi:DataProviderContact>
</pi:EmergencyCallData.ProviderInfo>

--boundary1--

Content-Type: application/EmergencyCallData.ServiceInfo+xml
Content-ID: <bloorpyhex@atlanta.example.com>
Content-Disposition: by-reference; handling=optional
<?xml version="1.0" encoding="UTF-8"?><svc:EmergencyCallData.ServiceInfo
  xmlns:svc="urn:ietf:params:xml:ns:EmergencyCallData:ServiceInfo"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <svc:DataProviderReference>string0987654321@example.org</svc:DataProviderReference>
  <svc:ServiceEnvironment>Residence</svc:ServiceEnvironment>
  <svc:ServiceType>VOIP</svc:ServiceType>
  <svc:ServiceMobility>Unknown</svc:ServiceMobility>
</svc:EmergencyCallData.ServiceInfo>

--boundary1--

Content-Type: application/EmergencyCallData.ProviderInfo+xml
Content-ID: <aaabbb@atlanta.example.com>
Content-Disposition: by-reference; handling=optional
<?xml version="1.0" encoding="UTF-8"?><pi:EmergencyCallData.ProviderInfo
  xmlns:pi="urn:ietf:params:xml:ns:EmergencyCallData:ProviderInfo"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <pi:DataProviderReference>string0987654321@example.org</pi:DataProviderReference>
  <pi:DataProviderString>Example VoIP Provider</pi:DataProviderString>
  <pi:ProviderID>urn:nena:companyid:ID123</pi:ProviderID>
</pi:EmergencyCallData.ProviderInfo>
Finally, the PSAP requests location information from the access network provider. The response is shown in Figure 14. Along with the location information, additional data is provided in the <provided-by> element of the PIDF-LO. This request and response is step #4.

<?xml version="1.0" encoding="UTF-8"?>
<pres xmlns="urn:ietf:params:xml:ns:pidf"
xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
entity="pres:alice@atlanta.example.com">
  <dm:device id="target123-1">
    <gp:geopriv>
      <gp:location-info>
        <civicAddress
          xmlns="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr">
          <country>AU</country>
          <A1>NSW</A1>
        </civicAddress>
      </gp:location-info>
    </gp:geopriv>
  </dm:device>
</pres>
<A3>Wollongong</A3>
<A4>North Wollongong</A4>
<RD>Flinders</RD>
<STS>Street</STS>
<RDBR>Campbell Street</RDBR>
<LMK>Gilligan’s Island</LMK>
<LOC>Corner</LOC>
<NAM>Video Rental Store</NAM>
<PC>2500</PC>
<ROOM>Westerns and Classics</ROOM>
<PLC>store</PLC>
<POBOX>Private Box 15</POBOX>
</civicAddress>
</gp:location-info>
<gp:usage-rules>
<gbp:retransmission-allowed>true</gbp:retransmission-allowed>
<gbp:retention-expiry>2013-12-10T20:00:00Z</gbp:retention-expiry>
</gp:usage-rules>
<gp:method>802.11</gp:method>

<gp:provided-by
xmlns="urn:ietf:params:xml:ns:EmergencyCallData">

<EmergencyCallDataReference
purpose="EmergencyCallData.ServiceInfo"
ref="https://example.com/ref2"/>

<EmergencyCallDataValue
<EmergencyCallData.ProviderInfo
xmlns=
"urn:ietf:params:xml:ns:EmergencyCallData:ProviderInfo">
<DataProviderReference>88QV4FpfZ976T@example.com</DataProviderReference>
<DataProviderString>University of Example</DataProviderString>
<ProviderID>urn:nena:companyid:uoi</ProviderID>
<ProviderIDSeries>NENA</ProviderIDSeries>
<TypeOfProvider>Other</TypeOfProvider>
<ContactURI>tel:+1-555-824-5222</ContactURI>
<Language>en</Language>
</EmergencyCallData.ProviderInfo>

<EmergencyCallData.Comment
xmlns="urn:ietf:params:xml:ns:EmergencyCallData:Comment">
<DataProviderReference>88QV4FpfZ976T@example.com</DataProviderReference>
</EmergencyCallData.Comment>
Figure 14: Access Network Provider returning PIDF-LO with Additional Data.

7. XML Schemas

This section defines the XML schemas of the five data blocks. Additionally, the provided-by schema is specified.

7.1. EmergencyCallData.ProviderInfo XML Schema

```xml
<?xml version="1.0"?>
<xs:schema
targetNamespace="urn:ietf:params:xml:ns:EmergencyCallData:ProviderInfo"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:pi="urn:ietf:params:xml:ns:EmergencyCallData:ProviderInfo"
xmlns:xc="urn:ietf:params:xml:ns:vcard-4.0"
elementFormDefault="qualified"
attributeFormDefault="unqualified">

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

<xs:import
namespace="urn:ietf:params:xml:ns:vcard-4.0"/>

<xs:element
name="EmergencyCallData.ProviderInfo"
type="pi:ProviderInfoType"
minOccurs="0" maxOccurs="1"/>

<xs:simpleType name="SubcontractorPriorityType">
<xs:restriction base="xs:string">
<xs:enumeration value="sub"/>
<xs:enumeration value="main"/>
</xs:restriction>
</xs:simpleType>
```
</xs:restriction>
</xs:simpleType>

<xs:complexType name="ProviderInfoType">
  <xs:sequence>
    <xs:element name="DataProviderReference" type="xs:token" minOccurs="1" maxOccurs="1"/>
    <xs:element name="DataProviderString" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="ProviderID" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="ProviderIDSeries" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="TypeOfProvider" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="ContactURI" type="xs:anyURI" minOccurs="1" maxOccurs="1"/>
    <xs:element name="Language" minOccurs="1" maxOccurs="unbounded" type="xs:simpleType">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:pattern value="([a-z]{2,3}((-[a-z]{3}){0,3})?|[a-z]{4,8})
            (-[a-z]{4})?(-([a-z]{2}|d(3)))?(-([0-9a-z]{5,8}|d[0-9-a-z]{3}))*(-[0-9a-wyz]-([0-9a-z]{2,8}))*\n            (-x([-0-9a-z]{1,8}))+?x([-0-9a-z]{1,8})+[a-z]{1,3}
            (-[0-9a-z]{2,8}){1,2}"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="DataProviderContact" minOccurs="0" maxOccurs="1" type="xs:complexType">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" ref="xc:vcard"/>
      </xs:sequence>
    </xs:complexType>
  </xs:sequence>
</xs:complexType>
<xs:element name="SubcontractorPrincipal"
type="xs:string" minOccurs="0" maxOccurs="1"/>

<xs:element name="SubcontractorPriority"
type="pi:SubcontractorPriorityType"
minOccurs="0" maxOccurs="1"/>

<xs:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>

Figure 15: EmergencyCallData.ProviderInfo XML Schema.

7.2. EmergencyCallData.ServiceInfo XML Schema
<?xml version="1.0"?>
<xs:schema
targetNamespace="urn:ietf:params:xml:ns:EmergencyCallData:ServiceInfo"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:svc="urn:ietf:params:xml:ns:EmergencyCallData:ServiceInfo"

elementFormDefault="qualified"

tagNameSpace="urn:ietf:params:xml:ns:EmergencyCallData:ServiceInfo"

	schemaLocation="http://www.w3.org/2001/XMLSchema"/>

<xs:element name="EmergencyCallData.ServiceInfo"
type="svc:ServiceInfoType"/>

<xs:complexType name="ServiceInfoType">
<xs:sequence>
<xs:element name="DataProviderReference"
type="xs:token" minOccurs="1" maxOccurs="1"/>

<xs:element name="ServiceEnvironment"
type="xs:string" minOccurs="0" maxOccurs="1"/>

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

<xs:element name="ServiceMobility"
type="xs:string" minOccurs="1" maxOccurs="1"/>

<xs:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:schema>

Figure 16: EmergencyCallData.ServiceInfo XML Schema.

7.3. EmergencyCallData.DeviceInfo XML Schema

<?xml version="1.0"?>
<xs:schema
targetNamespace="urn:ietf:params:xml:ns:EmergencyCallData:DeviceInfo"
xmlns:xs="http://www.w3.org/2001/XMLSchema"

elementFormDefault="qualified"

tagNameSpace="urn:ietf:params:xml:ns:EmergencyCallData:DeviceInfo"

	schemaLocation="http://www.w3.org/2001/XMLSchema"/>

<xs:element name="EmergencyCallData.DeviceInfo"
type="svc:DeviceInfoType"/>

<xs:complexType name="DeviceInfoType">
<xs:sequence>
<xs:element name="DeviceProviderReference"
type="xs:token" minOccurs="1" maxOccurs="1"/>

<xs:element name="DeviceModel" type="xs:string" minOccurs="0" maxOccurs="1"/>

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

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

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

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

<xs:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:schema>

Figure 16: EmergencyCallData.DeviceInfo XML Schema.
schemaLocation="http://www.w3.org/2001/xml.xsd"/>
<xs:element name="EmergencyCallData.DeviceInfo"
    type="dev:DeviceInfoType"/>
<xs:complexType name="DeviceInfoType">
    <xs:sequence>
        <xs:element name="DataProviderReference" 
            type="xs:token" minOccurs="1" maxOccurs="1"/>
        <xs:element name="DeviceClassification" 
            type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="DeviceMfgr" 
            type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="DeviceModelNr" 
            type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="UniqueDeviceID" minOccurs="0" 
            maxOccurs="unbounded">
            <xs:complexType>
                <xs:simpleContent>
                    <xs:extension base="xs:string">
                        <xs:attribute name="TypeOfDeviceID" 
                            type="xs:string" use="required"/>
                    </xs:extension>
                </xs:simpleContent>
            </xs:complexType>
        </xs:element>
        <xs:element name="DeviceSpecificData" 
            type="xs:anyURI" minOccurs="0" maxOccurs="1"/>
        <xs:element name="DeviceSpecificType" 
            type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:any namespace="##other" processContents="lax" 
            minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
7.4. EmergencyCallData.SubscriberInfo XML Schema

Example XML schema:

```xml
<?xml version="1.0"?>
<xs:schema targetNamespace=
"urn:ietf:params:xml:ns:EmergencyCallData:SubscriberInfo"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:sub="urn:ietf:params:xml:ns:EmergencyCallData:SubscriberInfo"
xmlns:xc="urn:ietf:params:xml:ns:vcard-4.0"
elementFormDefault="qualified"
attributeFormDefault="unqualified">

schemaLocation="http://www.w3.org/2001/xml.xsd"/>

<xs:import namespace="urn:ietf:params:xml:ns:vcard-4.0"/>

<xs:element name="EmergencyCallData.SubscriberInfo"
    type="sub:SubscriberInfoType"/>

<xs:complexType name="SubscriberInfoType">
  <xs:complexContent>
    <xs:restriction base="xs:anyType">
      <xs:sequence>
        <xs:element name="DataProviderReference" type="xs:token" minOccurs="1" maxOccurs="1"/>
        <xs:element name="SubscriberData">
          <xs:complexType>
            <xs:sequence>
              <xs:element maxOccurs="unbounded" ref="xc:vcard"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="privacyRequested" type="xs:boolean"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
```

Figure 17: EmergencyCallData.DeviceInfo XML Schema.
Figure 18: EmergencyCallData.SubscriberInfo XML Schema.

7.5. EmergencyCallData.Comment XML Schema
<xs:schema
  targetNamespace="urn:ietf:params:xml:ns:EmergencyCallData:Comment"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:com="urn:ietf:params:xml:ns:EmergencyCallData:Comment"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
    schemaLocation="http://www.w3.org/2001/xml.xsd"/>
  <xs:element name="EmergencyCallData.Comment"
    type="com:CommentType"/>
  <xs:complexType name="CommentType">
    <xs:sequence>
      <xs:element name="DataProviderReference"
        type="xs:token" minOccurs="1" maxOccurs="1"/>
      <xs:element name="Comment"
        type="com:CommentSubType" minOccurs="0"
        maxOccurs="unbounded"/>
      <xs:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="CommentSubType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute ref="xml:lang"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:schema>

Figure 19: EmergencyCallData.Comment XML Schema.

7.6. provided-by XML Schema

This section defines the provided-by schema.
<xs:schema
    targetNamespace=
        "urn:ietf:params:xml:ns:EmergencyCallData"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:ad="urn:ietf:params:xml:ns:EmergencyCallData"
xmlns:pi="urn:ietf:params:xml:ns:EmergencyCallData:ProviderInfo"
xmlns:svc="urn:ietf:params:xml:ns:EmergencyCallData:ServiceInfo"
xmlns:dev="urn:ietf:params:xml:ns:EmergencyCallData:DeviceInfo"
xmlns:sub=
        "urn:ietf:params:xml:ns:EmergencyCallData:SubscriberInfo"
xmlns:com="urn:ietf:params:xml:ns:EmergencyCallData:Comment"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified">

    <xs:import namespace=
        "urn:ietf:params:xml:ns:EmergencyCallData:ProviderInfo"/>
    <xs:import namespace=
        "urn:ietf:params:xml:ns:EmergencyCallData:ServiceInfo"/>
    <xs:import namespace=
        "urn:ietf:params:xml:ns:EmergencyCallData:DeviceInfo"/>
    <xs:import namespace=
        "urn:ietf:params:xml:ns:EmergencyCallData:SubscriberInfo"/>
    <xs:import namespace=
        "urn:ietf:params:xml:ns:EmergencyCallData:Comment"/>

    <xs:complexType name="provided-by-Type">
<xs:sequence>
    <xs:element name="EmergencyCallDataReference" type="ad:ByRefType"
        minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="EmergencyCallDataValue" type="ad:EmergencyCallDataValueType"
        minOccurs="0" maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:schema>

<!-- Additional Data By Reference -->

<xs:complexType name="ByRefType"/>
<xs:complexContent>
  <xs:restriction base="xs:anyType">
    <xs:sequence>
      <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
    </xs:sequence>
    <xs:attribute name="purpose" type="xs:token" use="required"/>
    <xs:attribute name="ref" type="xs:anyURI" use="required"/>
  </xs:restriction>
</xs:complexType>

<!-- Additional Data By Value -->

<xs:complexType name="EmergencyCallDataValueType">
  <xs:sequence>
    <xs:element name="EmergencyCallData.ProviderInfo" type="pi:ProviderInfoType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="EmergencyCallData.ServiceInfo" type="svc:ServiceInfoType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="EmergencyCallData.DeviceInfo" type="dev:DeviceInfoType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="EmergencyCallData.SubscriberInfo" type="sub:SubscriberInfoType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="EmergencyCallData.Comment" type="com:CommentType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

</xs:schema>

Figure 20: provided-by XML Schema.
8. Security Considerations

The data structures described in this document contain information usually considered private. When information is provided by value, entities that are a party to the SIP signaling (such as proxy servers and back-to-back user agents) will have access to it and need to protect it against inappropriate disclosure. An entity that is able to eavesdrop on the SIP signaling will also have access. Some media (such as in the clear Wi-Fi) is more vulnerable than others (such as 3G or 4G cellular data traffic) to eavesdropping. Mechanisms that protect against eavesdropping (such as Transport Layer Security (TLS)) SHOULD be preferentially used whenever feasible. When information is provided by reference, HTTPS is specified for dereferencing, and the provider of the information is REQUIRED to validate the credentials of the requester. While the creation of a public key infrastructure (PKI) that has global scope may be difficult, the alternatives to creating devices and services that can provide critical information securely are more daunting. The provider of the information MAY enforce any policy it wishes to use, but PSAPs and responder agencies SHOULD deploy a PKI so that providers of additional data can check the certificate of the client and decide the appropriate policy to enforce based on that certificate.

Ideally, the PSAP and emergency responders will be given credentials signed by an authority trusted by the data provider. In most circumstances, nationally recognized credentials would be sufficient, and if the emergency services arranges a PKI, data providers could be provisioned with the root CA public key for a given nation. Some nations are developing a PKI for this, and related, purposes. Since calls could be made from devices where the device and/or the service provider(s) are not local to the emergency services authorities, globally recognized credentials are useful. This might be accomplished by extending the notion of the "forest guide" described in [RFC5582] to allow the forest guide to provide the credential of the PKI root for areas for which it has coverage information, but standards for such a mechanism are not yet available. In its absence, the data provider will need to obtain by out of band means the root CA credentials for any areas to which it is willing to provide additional data. With the credential of the root CA for a national emergency services PKI, the data provider server can validate the credentials of an entity requesting additional data by reference.

The data provider also needs a credential that can be verified by the emergency services to know that it is receiving data from an authorized server. The emergency services authorities could provide credentials, distinguishable from credentials provided to emergency
responders and PSAPs, which could be used to validate data providers. Such credentials would have to be acceptable to any PSAP or responder that could receive a call with additional data supplied by that provider. This would be extensible to global credential validation using the forest guide as mentioned above. In the absence of such credentials, the emergency services authorities could maintain a list of local data providers’ credentials as provided to them out of band. At a minimum, the emergency services authorities could obtain a credential from the DNS entry of the domain in the Additional Data URI to at least validate that the server is known to the domain providing the URI.

Data provided by devices by reference have similar credential validation issues as for service providers, and the solutions are the same.

Much of the information supplied by service providers and devices is private and confidential; service providers and devices generally go to lengths to protect this information; disclosing it in the context of an emergency call is a trade-off to protect the greater interest of the customer in an emergency.

Neither service providers nor devices will supply private information unless the call is recognized as an emergency call. In cellular telephony systems (such as those using 3GPP IMS), there are different procedures for an originating device to place an emergency versus a normal call. If a call that is really an emergency call is initiated as a normal call and the cellular service provider recognizes this, 3GPP IMS permits the service provider to either accept the call anyway or reject it with a specific code that instructs the device to retry the call as an emergency call. Service providers SHOULD choose the latter, because otherwise the device will not have included the information specified in this document (since the device didn’t recognize the call as being an emergency call).

9. Privacy Considerations

This document enables functionality for conveying additional information about the caller and the caller’s device and service to the callee. Some of this information is personal data and therefore privacy concerns arise. An explicit privacy indicator for information directly relating to the caller’s identity is defined and use is mandatory. However, observance of this request for privacy and what information it relates to is determined by the destination jurisdiction.

There are a number of privacy concerns with non-emergency real-time communication services that are also applicable to emergency calling.
Data protection regulation world-wide has, however, decided to create exceptions for emergency services since the drawbacks of disclosing personal data are outweighed by the benefit for the emergency caller. Hence, the data protection rights of individuals are commonly waived for emergency situations. There are, however, still various countries that offer some degree of anonymity for the caller towards PSAP call takers.

The functionality defined in this document, however, far exceeds the amount of information sharing found in the legacy POTS system. For this reason there are additional privacy threats to consider, which are described in more detail in [RFC6973].

Stored Data Compromise: There is an increased risk of stored data compromise since additional data is collected and stored in databases. Without adequate measures to secure stored data from unauthorized or inappropriate access at access network providers, service providers, end devices, as well as PSAPs, individuals are exposed to potential financial, reputational, or physical harm.

Misattribution: If the personal data collected and conveyed is incorrect or inaccurate then this may lead to misattribution. Misattribution occurs when data or communications related to one individual are attributed to another.

Identification: By the nature of the additional data and its capability to provide much richer information about the caller, the call, and the location, the calling party is identified in a much better way. Some users may feel uncomfortable with this degree of information sharing even in emergency services situations.

Secondary Use: There is a risk of secondary use, which is the use of collected information about an individual without the individual’s consent for a purpose different from that for which the information was collected. The stated purpose of the additional data is for emergency services purposes but theoretically the same information could be used for any other call as well. Additionally, parties involved in the emergency call may retain the obtained information and may re-use it for other, non-emergency services purposes.

Disclosure: When the data defined in this document is not properly protected (while in transit with traditional communication security techniques, and while stored using access control mechanisms) there is the risk of disclosure, which is the revelation of private information about an individual.
To mitigate these privacy risks the following countermeasures can be taken:

In regions where callers can elect to suppress certain personally identifying information, network or PSAP functionality can inspect privacy flags within the SIP headers to determine what information may be passed, stored, or displayed to comply with local policy or law. RFC 3325 [RFC3325] defines the "id" priv-value token. The presence of this privacy type in a Privacy header field indicates that the user would like the network asserted identity to be kept private with respect to SIP entities outside the trust domain with which the user authenticated, including the PSAP.

This document defines various data structures that contain privacy-sensitive data. For example, identifiers for the device (e.g., serial number, MAC address) or account/SIM (e.g., IMSI), contact information for the user, location of the caller. Local regulations may govern what data must be provided in emergency calls, but in general, the emergency call system is aided by the information described in this document. There is a tradeoff between the privacy considerations and the utility of the data. For protection, this specification requires all retrieval of data passed by reference to be protected against eavesdropping and alteration via communication security techniques (namely TLS). Furthermore, security safeguards are required to prevent unauthorized access to stored data. Various security incidents over at least the past few decades have shown that data breaches are not uncommon and are often caused by lack of proper access control frameworks, software bugs (such as buffer overflows), or missing input parsing (such as SQL injection attacks). The risks of data breaches is increased with the obligation for emergency services to retain emergency call related data for extended periods (e.g., several years are the norm).

Finally, it is also worth highlighting the nature of the SIP communication architecture, which introduces additional complications for privacy. Some forms of data can be sent by value in the SIP signaling or by reference (a URL in the SIP signaling). When data is sent by value, all intermediaries have access to the data. As such, these intermediaries may also introduce additional privacy risk. Therefore, in situations where the conveyed information is privacy-sensitive and intermediaries are involved, transmitting by reference might be appropriate, assuming the source of the data can operate a sufficient dereferencing infrastructure and that proper access control policies are available for distinguishing the different entities dereferencing the reference. Without access control policies any party in possession of the reference is able to resolve the reference and to obtain the data, including intermediaries.
10. IANA Considerations

10.1. Registry creation

This document creates a new registry called ‘Emergency Call Additional Data’. The following sub-registries are created for this registry.

10.1.1. Provider ID Series Registry

This document creates a new sub-registry called ‘Additional Call Data Provider ID Series’. As defined in [RFC5226], this registry operates under "Expert Review" rules. The expert should determine that the entity requesting a new value is a legitimate issuer of service provider IDs suitable for use in Additional Call Data.

Private entities issuing and using internally-generated IDs are encouraged to register and use a unique identifier. This guarantees that IDs issued and used by the entity are globally unique and distinguishable.

The content of this registry includes:

Name: The identifier which will be used in the ‘ProviderIDSeries’ element.

Source: The full name of the organization issuing the identifiers.

URL: A URL to the organization for further information.

The initial set of values is listed in Figure 21.

<table>
<thead>
<tr>
<th>Name</th>
<th>Source</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>NENA</td>
<td>National Emergency Number Association</td>
<td><a href="http://www.nena.org">http://www.nena.org</a></td>
</tr>
<tr>
<td>EENA</td>
<td>European Emergency Number Association</td>
<td><a href="http://www.eena.org">http://www.eena.org</a></td>
</tr>
<tr>
<td>domain</td>
<td>(The ID is a fully-qualified domain name)</td>
<td>(not applicable)</td>
</tr>
</tbody>
</table>

Figure 21: Provider ID Series Registry.
10.1.2. Service Environment Registry

This document creates a new sub-registry called ‘Additional Call Service Environment’. As defined in [RFC5226], this registry operates under "Expert Review" rules. The expert should determine that the entity requesting a new value is relevant for this service element, and that the new value is distinct from existing values, and its use is unambiguous.

The content of this registry includes:

Token: The value to be used in the <ServiceEnvironment> element.

Description: A short description of the value.

The initial set of values is listed in Figure 22.

<table>
<thead>
<tr>
<th>Token</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business</td>
<td>Business service</td>
</tr>
<tr>
<td>Residence</td>
<td>Residential service</td>
</tr>
<tr>
<td>unknown</td>
<td>Type of service unknown (e.g., anonymous pre-paid service)</td>
</tr>
</tbody>
</table>

Figure 22: Service Environment Registry.

10.1.3. Service Type Registry

This document creates a new sub-registry called ‘Additional Call Service Type’. As defined in [RFC5226], this registry operates under "Expert Review" rules. The expert should determine that the entity requesting a new value is relevant for this service element and that the requested value is clearly distinct from other values so that there is no ambiguity as to when the value is to be used or which value is to be used.

The content of this registry includes:

Name: The value to be used in the <ServiceType> element.

Description: A short description of the value.

The initial set of values is listed in Figure 3.
10.1.4. Service Mobility Registry

This document creates a new sub-registry called ‘Additional Call Service Mobility’. As defined in [RFC5226], this registry operates under "Expert Review" rules. The expert should determine that the entity requesting a new value is relevant for this service element and that the requested value is clearly distinct from other values so that there is no ambiguity as to when the value is to be used or which value is to be used.

The content of this registry includes:

Token:  The value used in the <ServiceMobility> element.
Description:  A short description of the value.

The initial set of values is listed in Figure 23.

<table>
<thead>
<tr>
<th>Token</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile</td>
<td>The device is able to move at any time</td>
</tr>
<tr>
<td>Fixed</td>
<td>The device is not expected to move unless the service is relocated</td>
</tr>
<tr>
<td>Nomadic</td>
<td>The device is not expected to change its point of attachment while on a call</td>
</tr>
<tr>
<td>Unknown</td>
<td>No information is known about the service mobility environment for the device</td>
</tr>
</tbody>
</table>

Figure 23: Service Environment Registry.

10.1.5. Service Provider Type Registry

This document creates a new sub-registry called ‘Service Provider Type’. As defined in [RFC5226], this registry operates under "Expert Review". The expert should determine that the proposed new value is distinct from existing values and appropriate for use in the TypeOfServicerProvider element.

The content of this registry includes:

Tokenproviderid:  The value used in the ’TypeOfProvider’ element.
Description: A short description of the type of service provider.

The initial set of values is defined in Figure 1.

10.1.6. Service Delivered Registry

This document creates a new sub-registry called ‘Service Delivered’. As defined in [RFC5226], this registry operates under "Expert Review" rules. The expert should consider whether the proposed service is unique from existing services and the definition of the service will be clear to implementors and PSAPs/responders.

The content of this registry includes:

Name: The value used in the ‘ServiceType’ element.

Description: Short description identifying the nature of the service.

The initial set of values are defined in Figure 3.

10.1.7. Device Classification Registry

This document creates a new sub-registry called ‘Device Classification’. As defined in [RFC5226], this registry operates under "Expert Review" rules. The expert should consider whether the proposed class is unique from existing classes and the definition of the class will be clear to implementors and PSAPs/responders.

The content of this registry includes:

Token: Value used in the ‘DeviceClassification’ element.

Description: Short description identifying the device type.

The initial set of values are defined in Figure 5.

10.1.8. Device ID Type Type Registry

This document creates a new sub-registry called ‘Additional Call Data Device ID Type’. As defined in [RFC5226], this registry operates under "Expert Review" rules. The expert should ascertain that the proposed type is well understood, and provides the information useful to PSAPs and responders to uniquely identify a device.

The content of this registry includes:

Token: The value to be placed in the ‘TypeOfDeviceID’ element.
Description: Short description identifying the type of the device ID.

The initial set of values are defined in Figure 6.

10.1.9. Device/Service Data Type Registry

This document creates a new sub-registry called 'Device/Service Data Type Registry'. As defined in [RFC5226], this registry operates under "Expert Review" and "Specification Required" rules. The expert should ascertain that the proposed type is well understood, and provides information useful to PSAPs and responders. The specification must contain a complete description of the data, and a precise format specification suitable to allow interoperable implementations.

The content of this registry includes:

Token: The value to be placed in the <DeviceSpecificType> element.

Description: Short description identifying the data.

Specification: Citation for the specification of the data.

The initial set of values are listed in Figure 24.

+----------+----------------------------------------+---------------+
| Token    | Description                            | Specification |
+----------+----------------------------------------+---------------+
| IEEE1512 | Common Incident Management Message Set | IEEE 1512-2006|
+----------+----------------------------------------+---------------+

Figure 24: Device/Service Data Type Registry.

10.1.10. Emergency Call Data Types Registry

This document creates a new sub-registry called 'Emergency Call Data Types' in the 'purpose' registry established by RFC 3261 [RFC3261]. As defined in [RFC5226], this registry operates under "Expert Review" and "Specification Required" rules. The expert is responsible for verifying that the document contains a complete and clear specification and the proposed functionality does not obviously duplicate existing functionality.

The registry contains an entry for every data block that can be sent with an emergency call using the mechanisms as specified in this document. Each data block is identified by the "root" of its MIME subtype (which is the part after 'EmergencyCallData.'). If the MIME
subtype does not start with ‘EmergencyCallData.’ then it cannot be registered here nor used in a Call-Info header as specified in this document. The subtype MAY exist under any MIME type (although most commonly these are under ‘Application/’ this is NOT REQUIRED).

The content of this registry includes:

Token: The root of the data’s MIME subtype (not including the ‘EmergencyCallData’ prefix and any suffix such as ‘+xml’)

Reference: The document that describes the data object

Note that the values from this registry are part of the ‘EmergencyCallData’ compound value; when used as a value of the ‘purpose’ parameter of the Call-Info header, the values listed in this registry are prefixed by ‘EmergencyCallData.’ per the the ‘EmergencyCallData’ registation Section 10.2.

The initial set of values are listed in Figure 25.

<table>
<thead>
<tr>
<th>Token</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProviderInfo</td>
<td>[This RFC]</td>
</tr>
<tr>
<td>ServiceInfo</td>
<td>[This RFC]</td>
</tr>
<tr>
<td>DeviceInfo</td>
<td>[This RFC]</td>
</tr>
<tr>
<td>SubscriberInfo</td>
<td>[This RFC]</td>
</tr>
<tr>
<td>Comment</td>
<td>[This RFC]</td>
</tr>
</tbody>
</table>

Figure 25: Additional Data Blocks Registry.

10.2. ‘EmergencyCallData’ Purpose Parameter Value

This document defines the ‘EmergencyCallData’ value for the “purpose” parameter of the Call-Info header field. The Call-Info header and the corresponding registry for the ‘purpose’ parameter was established with RFC 3261 [RFC3261]. Note that ‘EmergencyCallData’ is a compound value; when used as a value of the ‘purpose’ parameter of the Call-Info header, ‘EmergencyCallData’ is immediately followed by a dot (‘.’) and a value from the ‘Emergency Call Data Types’ registry Section 10.1.10.
10.3. URN Sub-Namespace Registration for provided-by Registry Entry

This section registers the namespace specified in Section 10.5.1 in the provided-by registry established by RFC 4119, for usage within the <provided-by> element of a P1DF-LO.

The schema for the <provided-by> element used by this document is specified in Section 7.6.

10.4. MIME Registrations

10.4.1. MIME Content-type Registration for ‘application/EmergencyCallData.ProviderInfo+xml’

This specification requests the registration of a new MIME type according to the procedures of RFC 6838 [RFC6838] and guidelines in RFC 7303 [RFC7303].

MIME media type name: application

MIME subtype name: EmergencyCallData.ProviderInfo+xml

Mandatory parameters: none

Optional parameters: charset (indicates the character encoding of the contents)

Encoding considerations: Uses XML, which can contain 8-bit characters, depending on the character encoding. See Section 3.2 of RFC 7303 [RFC7303].

Security considerations: This content type is designed to carry the data provider information, which is a sub-category of additional data about an emergency call. Since this data may contain personal information, appropriate precautions might be needed to limit unauthorized access, inappropriate disclosure, and eavesdropping of personal information. Please refer to Section 8 and Section 9 for more information.

Interoperability considerations: None
Published specification: [TBD: This specification]

Applications which use this media type: Emergency Services

Additional information:

    Magic Number: None
    File Extension: .xml
    Macintosh file type code: ’TEXT’

Person and email address for further information: Hannes Tschofenig, Hannes.Tschofenig@gmx.net

Intended usage: LIMITED USE

Author: This specification is a work item of the IETF ECRIT working group, with mailing list address <ecrit@ietf.org>.

Change controller: The IESG <ietf@ietf.org>

10.4.2.  MIME Content-type Registration for ‘application/EmergencyCallData.ServiceInfo+xml’

This specification requests the registration of a new MIME type according to the procedures of RFC 6838 [RFC6838] and guidelines in RFC 7303 [RFC7303].

MIME media type name: application

MIME subtype name: EmergencyCallData.ServiceInfo+xml

Mandatory parameters: none

Optional parameters: charset (indicates the character encoding of the contents)

Encoding considerations: Uses XML, which can contain 8-bit characters, depending on the character encoding. See Section 3.2 of RFC 7303 [RFC7303].

Security considerations: This content type is designed to carry the service information, which is a sub-category of additional data about an emergency call. Since this data may contain personal information, appropriate precautions may be needed to limit unauthorized access, inappropriate disclosure, and...
eavesdropping of personal information. Please refer to Section 8 and Section 9 for more information.

Interoperability considerations: None

Published specification: [TBD: This specification]

Applications which use this media type: Emergency Services

Additional information:

Magic Number: None

File Extension: .xml

Macintosh file type code: ’TEXT’

Person and email address for further information: Hannes Tschofenig, Hannes.Tschofenig@gmx.net

Intended usage: LIMITED USE

Author: This specification is a work item of the IETF ECRIT working group, with mailing list address <ecrit@ietf.org>.

Change controller: The IESG <ietf@ietf.org>

10.4.3. MIME Content-type Registration for ‘application/EmergencyCallData.DeviceInfo+xml’

This specification requests the registration of a new MIME type according to the procedures of RFC 6838 [RFC6838] and guidelines in RFC 7303 [RFC7303].

MIME media type name: application

MIME subtype name: EmergencyCallData.DeviceInfo+xml

Mandatory parameters: none

Optional parameters: charset (indicates the character encoding of the contents)

Encoding considerations: Uses XML, which can contain 8-bit characters, depending on the character encoding. See Section 3.2 of RFC 7303 [RFC7303].

Security considerations: This content type is designed to carry device information, which is a sub-category of additional data about an emergency call. Since this data contains personal information, appropriate precautions need to be taken to limit unauthorized access, inappropriate disclosure to third parties, and eavesdropping of this information. Please refer to Section 8 and Section 9 for more information.

Interoperability considerations: None

Published specification: [TBD: This specification]

Applications which use this media type: Emergency Services

Additional information:

Magic Number: None

File Extension: .xml

Macintosh file type code: ‘TEXT’

Person and email address for further information: Hannes Tschofenig, Hannes.Tschofenig@gmx.net

Intended usage: LIMITED USE

Author: This specification is a work item of the IETF ECRIT working group, with mailing list address <ecrit@ietf.org>.

Change controller: The IESG <ietf@ietf.org>

10.4.4. MIME Content-type Registration for ‘application/EmergencyCallData.SubscriberInfo+xml’

This specification requests the registration of a new MIME type according to the procedures of RFC 6838 [RFC6838] and guidelines in RFC 7303 [RFC7303].

MIME media type name: application

MIME subtype name: EmergencyCallData.SubscriberInfo+xml

Mandatory parameters: none

Optional parameters: charset (indicates the character encoding of the contents)
Encoding considerations: Uses XML, which can contain 8-bit characters, depending on the character encoding. See Section 3.2 of RFC 7303 [RFC7303].

Security considerations: This content type is designed to carry owner/subscriber information, which is a sub-category of additional data about an emergency call. Since this data contains personal information, appropriate precautions need to be taken to limit unauthorized access, inappropriate disclosure to third parties, and eavesdropping of this information. Please refer to Section 8 and Section 9 for more information.

Interoperability considerations: None

Published specification: [TBD: This specification]

Applications which use this media type: Emergency Services

Additional information:

  Magic Number: None

  File Extension: .xml

  Macintosh file type code: 'TEXT'

Person and email address for further information: Hannes Tschofenig, Hannes.Tschofenig@gmx.net

Intended usage: LIMITED USE

Author: This specification is a work item of the IETF ECRIT working group, with mailing list address <ecrit@ietf.org>.

Change controller: The IESG <ietf@ietf.org>

10.4.5. MIME Content-type Registration for 'application/EmergencyCallData.Comment+xml'

This specification requests the registration of a new MIME type according to the procedures of RFC 6838 [RFC6838] and guidelines in RFC 7303 [RFC7303].

MIME media type name: application

MIME subtype name: EmergencyCallData.Comment+xml

Mandatory parameters: none
Optional parameters: charset (indicates the character encoding of the contents)

Encoding considerations: Uses XML, which can contain 8-bit characters, depending on the character encoding. See Section 3.2 of RFC 7303 [RFC7303].

Security considerations: This content type is designed to carry a comment, which is a sub-category of additional data about an emergency call. This data may contain personal information. Appropriate precautions may be needed to limit unauthorized access, inappropriate disclosure to third parties, and eavesdropping of this information. Please refer to Section 8 and Section 9 for more information.

Interoperability considerations: None

Published specification: [TBD: This specification]

Applications which use this media type: Emergency Services

Additional information:

  Magic Number: None

  File Extension: .xml

  Macintosh file type code: 'TEXT'

Person and email address for further information: Hannes Tschofenig, Hannes.Tschofenig@gmx.net

Intended usage: LIMITED USE

Author: This specification is a work item of the IETF ECRIT working group, with mailing list address <ecrit@ietf.org>.

Change controller: The IESG <ietf@ietf.org>

10.5. URN Sub-Namespace Registration

10.5.1. Registration for urn:ietf:params:xml:ns:EmergencyCallData

  This section registers a new XML namespace, as per the guidelines in RFC 3680 [RFC3680].

  URI: urn:ietf:params:xml:ns:EmergencyCallData
Registrant Contact: IETF, ECRIT working group, <ecrit@ietf.org>, as delegated by the IESG <iesg@ietf.org>.

XML:

BEGIN
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
  "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="content-type" content="text/html;charset=iso-8859-1"/>
<title>Namespace for Additional Emergency Call Data</title>
</head>
<body>
<h1>Namespace for Additional Data related to an Emergency Call</h1>
<p>See [TBD: This document].</p>
</body>
</html>
END

10.5.2. Registration for
urn:ietf:params:xml:ns:EmergencyCallData:ProviderInfo

This section registers a new XML namespace, as per the guidelines in RFC 3688 [RFC3688].


Registrant Contact: IETF, ECRIT working group, <ecrit@ietf.org>, as delegated by the IESG <iesg@ietf.org>.

XML:
10.5.3. Registration for
  urn:ietf:params:xml:ns:EmergencyCallData:ServiceInfo

This section registers a new XML namespace, as per the guidelines in
RFC 3688 [RFC3688].


Registrant Contact: IETF, ECRIT working group, <ecrit@ietf.org>, as
delegated by the IESG <iesg@ietf.org>.

XML:
10.5.4. Registration for
urn:ietf:params:xml:ns:EmergencyCallData:DeviceInfo

This section registers a new XML namespace, as per the guidelines in
RFC 3688 [RFC3688].


Registrant Contact: IETF, ECRIT working group, <ecrit@ietf.org>, as
delegated by the IESG <iesg@ietf.org>.

XML:
10.5.5. Registration for

urn:ietf:params:xml:ns:EmergencyCallData:SubscriberInfo

This section registers a new XML namespace, as per the guidelines in
RFC 3688 [RFC3688].


Registrant Contact: IETF, ECRIT working group, <ecrit@ietf.org>, as
delegated by the IESG <iesg@ietf.org>.

XML:

10.5.6. Registration for urn:ietf:params:xml:ns:EmergencyCallData:Comment

This section registers a new XML namespace, as per the guidelines in RFC 3688 [RFC3688].

URI:   urn:ietf:params:xml:ns:EmergencyCallData:Comment

Registrant Contact:  IETF, ECRIT working group, <ecrit@ietf.org>, as delegated by the IESG <iesg@ietf.org>.

XML:
10.6. Schema Registrations

This specification registers five schemas, as per the guidelines in RFC 3688 [RFC3688].

Registrant Contact: IETF, ECRIT Working Group (ecrit@ietf.org), as delegated by the IESG (iesg@ietf.org).

XML: The XML schema can be found in Figure 15.

Registrant Contact: IETF, ECRIT Working Group (ectit@ietf.org), as delegated by the IESG (iesg@ietf.org).

XML: The XML schema can be found in Figure 16.

Registrant Contact: IETF, ECRIT Working Group (ecrit@ietf.org), as delegated by the IESG (iesg@ietf.org).

XML: The XML schema can be found in Figure 17.

10.7. VCard Parameter Value Registration

This document registers a new value in the vCARD Parameter Values registry as defined by [RFC6350] with the following template:

Value: main

Purpose: The main telephone number, typically of an enterprise, as opposed to a direct dial number of an individual employee

Conformance: This value can be used with the "TYPE" parameter applied on the "TEL" property.

Example(s): TEL;VALUE=uri;TYPE="main,voice";PREF=1;tel:+1-418-656-9000

11. Acknowledgments

This work was originally started in NENA and has benefitted from a large number of participants in NENA standardization efforts, originally in the Long Term Definition Working Group, the Data Technical Committee and most recently the Additional Data working group. The authors are grateful for the initial work and extended comments provided by many NENA participants, including Delaine Arnold, Marc Berryman, Guy Caron, Mark Fletcher, Brian Dupras, James Leyerle, Kathy McMahon, Christian Militeau, Ira Pyles, Matt Serra, and Robert (Bob) Sherry. Amursana Khiyod, Robert Sherry, Frank Rahoi, Scott Ross, Tom Klepetka provided valuable feedback regarding the vCard/xCard use in this specification.

We would also like to thank Paul Kyzivat, Gunnar Hellstrom, Martin Thomson, Keith Drage, Laura Liess, Chris Santer, Barbara Stark, Chris Santer, and Archie Cobbs for their review comments. Guy Caron deserves special mention for his detailed and extensive review comments.
12. References

12.1. Normative References


12.2. Informational References

[I-D.gellens-slim-negotiating-human-language]

[LanguageTagRegistry]


12.3. URIs


Appendix A. XML Schema for vCard/xCard

This section contains the vCard/xCard XML schema version of the Relax NG schema defined in RFC 6351 [RFC6351] for simplified use with the XML schemas defined in this document. The schema in RFC 6351 [RFC6351] is the normative source and this section is informative only.
<xs:pattern value="(\d\d(\d\d(\d\d)?)?|\-\d\d(\d\d?)|--\d\d)\d(\d\d)\d((\d\d)?)?Z([+-]\d\d(\d\d)?)?"/>
</xs:restriction>
</xs:element>
</xs:complexType>
</xs:element>
<!-- 4.3.3 -->
<xs:element name="date-time"
substitutionGroup="ns1:value-date-and-or-time">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:pattern value="(\d{8}|--\d{4}|---\d\d)T\d(\d\d\d?)(\d\d)\d(\d\d(\d\d)?)?(Z([+-]\d\d(\d\d)?)?"/>
</xs:restriction>
</xs:element>
</xs:complexType>
</xs:element>
<!-- 4.3.4 -->
<xs:element name="value-date-and-or-time" abstract="true"/>
<!-- 4.3.5 -->
<xs:complexType name="value-timestamp">
<xs:sequence>
<xs:element ref="ns1:timestamp"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="timestamp">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:pattern value="\d(\d{8})\d{6}(Z([+-]\d\d(\d\d)?)?"/>
</xs:restriction>
</xs:element>
</xs:complexType>
</xs:element>
<!-- 4.4 -->
<xs:element name="boolean" type="xs:boolean"/>
<!-- 4.5 -->
<xs:element name="integer" type="xs:integer"/>
<!-- 4.6 -->
<xs:element name="float" type="xs:float"/>
<!-- 4.7 -->
<xs:element name="utc-offset">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:pattern value="([+-]\d\d(\d\d)?)?"/>
</xs:restriction>
</xs:element>
</xs:complexType>
</xs:element>
<!-- 4.8 -->
<xs:element name="language-tag">
<xs:simpleType>
</xs:element>
<xs:restriction base="xs:string">
  <xs:pattern value="([a-z]{2,3}((-([a-z]{3}){0,3})?|[a-z]{4,8})
  (-[a-z]{4})?(-([a-z]2|d3))?(-([0-9a-z]{5,8})
  \d[0-9a-z]{3})*(-[0-9a-wyz][-0-9a-z]{2,8})*
  (-x([-0-9a-z]{1,8}))+?x([-0-9a-z]{1,8})+|[a-z]{1,3}
  (-[0-9a-z]{2,8}){1,2}="/>
</xs:restriction>
</xs:simpleType>

5.1
-->
<xs:group name="param-language">
  <xs:annotation>
    <xs:documentation>Section 5: Parameters</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element minOccurs="0" ref="ns1:language"/>
  </xs:sequence>
</xs:group>
<xs:element name="language">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ns1:language-tag"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

5.2
-->
<xs:group name="param-pref">
  <xs:sequence>
    <xs:element minOccurs="0" ref="ns1:pref"/>
  </xs:sequence>
</xs:group>
<xs:element name="pref">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="integer">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:minInclusive value="1"/>
            <xs:maxInclusive value="100"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
></xs:element>
<!-- 5.4 -->
<xs:group name="param-altid">
  <xs:sequence>
    <xs:element minOccurs="0" ref="ns1:altid"/>
  </xs:sequence>
</xs:group>
<xs:element name="altid">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ns1:text"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!-- 5.5 -->
<xs:group name="param-pid">
  <xs:sequence>
    <xs:element minOccurs="0" ref="ns1:pid"/>
  </xs:sequence>
</xs:group>
<xs:element name="pid">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" name="text">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:pattern value="\d+(\.\d+)?"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!-- 5.6 -->
<xs:group name="param-type">
  <xs:sequence>
    <xs:element minOccurs="0" ref="ns1:type"/>
  </xs:sequence>
</xs:group>
<xs:element name="type">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" name="text">
        <xs:simpleType>
          <xs:restriction base="xs:token">
            <xs:enumeration value="work"/>
            <xs:enumeration value="home"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType>
    <xs:element name="geo">
        <xs:complexType>
            <xs:sequence>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:group>

<!-- 5.10 -->
<xs:group name="param-geo">
    <xs:sequence>
        <xs:element minOccurs="0" name="geo">
            <xs:complexType>
                <xs:sequence>
                    <xs:element ref="ns1:uri"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:group>

<!-- 5.11 -->
<xs:group name="param-tz">
    <xs:sequence>
        <xs:element minOccurs="0" name="tz">
            <xs:complexType>
                <xs:choice>
                    <xs:element ref="ns1:text"/>
                    <xs:element ref="ns1:uri"/>
                </xs:choice>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:group>

6.1.3
-->
<xs:element name="source">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="parameters">
                <xs:complexType>
                    <xs:sequence>
                        <xs:group ref="ns1:param-altid"/>
                        <xs:group ref="ns1:param-pid"/>
                        <xs:group ref="ns1:param-pref"/>
                        <xs:group ref="ns1:param-mediatype"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element ref="ns1:uri"/>
</xs:complexType>
</xs:element>
<xs:element name="kind">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="text">
        <xs:complexType>
          <xs:union memberTypes="ns1:x-name ns1:iana-token">
            <xs:restriction base="xs:token">
              <xs:enumeration value="individual"/>
            </xs:restriction>
            <xs:restriction base="xs:token">
              <xs:enumeration value="group"/>
            </xs:restriction>
            <xs:restriction base="xs:token">
              <xs:enumeration value="org"/>
            </xs:restriction>
            <xs:restriction base="xs:token">
              <xs:enumeration value="location"/>
            </xs:restriction>
          </xs:union>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- 6.2.1 -->
<xs:element name="fn">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="parameters">
        <xs:complexType>
          <xs:sequence>
            <xs:group ref="ns1:param-language"/>
            <xs:group ref="ns1:param-altid"/>
            <xs:group ref="ns1:param-pid"/>
            <xs:group ref="ns1:param-pref"/>
            <xs:group ref="ns1:param-type"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="n">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="parameters">
        <xs:complexType>
          <xs:sequence>
            <xs:group ref="ns1:param-language"/>
            <xs:group ref="ns1:param-sort-as"/>
            <xs:group ref="ns1:param-altid"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element maxOccurs="unbounded" ref="ns1:surname"/>
      <xs:element maxOccurs="unbounded" ref="ns1:given"/>
      <xs:element maxOccurs="unbounded" ref="ns1:additional"/>
      <xs:element maxOccurs="unbounded" ref="ns1:prefix"/>
      <xs:element maxOccurs="unbounded" ref="ns1:suffix"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="surname" type="xs:string"/>
<xs:element name="given" type="xs:string"/>
<xs:element name="additional" type="xs:string"/>
<xs:element name="prefix" type="xs:string"/>
</xs:element>

<xs:element name="nickname">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="parameters">
        <xs:complexType>
          <xs:sequence>
            <xs:group ref="ns1:param-language"/>
            <xs:group ref="ns1:param-altid"/>
            <xs:group ref="ns1:param-pid"/>
            <xs:group ref="ns1:param-pref"/>
            <xs:group ref="ns1:param-type"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:group ref="ns1:value-text-list"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!-- 6.2.4 -->
<xs:element name="photo">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="parameters">
        <xs:complexType>
          <xs:sequence>
            <xs:group ref="ns1:param-altid"/>
            <xs:group ref="ns1:param-pid"/>
            <xs:group ref="ns1:param-pref"/>
            <xs:group ref="ns1:param-type"/>
            <xs:group ref="ns1:param-mediatype"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element ref="ns1:uri"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- 6.2.5 -->
<xs:element name="bday">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="parameters">
        <xs:complexType>
          <xs:sequence>
            <xs:group ref="ns1:param-altid"/>
            <xs:group ref="ns1:param-calscale"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:choice>
        <xs:element ref="ns1:value-date-and-or-time"/>
        <xs:element ref="ns1:text"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- 6.2.6 -->
<xs:element name="anniversary">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="parameters">
        <xs:complexType>
          <xs:sequence>
            <xs:group ref="ns1:param-altid"/>
            <xs:group ref="ns1:param-calscale"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
</!-- 6.2.7 -->
<xs:element name="gender">
<xs:complexType>
<xs:sequence>
<xs:element ref="ns1:sex"/>
<xs:element minOccurs="0" ref="ns1:identity"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="sex">
<xs:simpleType>
<xs:restriction base="xs:token">
<xs:enumeration value="/"/>
<xs:enumeration value="M"/>
<xs:enumeration value="F"/>
<xs:enumeration value="O"/>
<xs:enumeration value="N"/>
<xs:enumeration value="U"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="identity" type="xs:string"/>
</!-- 6.3.1 -->
<xs:group name="param-label">
<xs:sequence>
<xs:element minOccurs="0" ref="ns1:label"/>
</xs:sequence>
</xs:group>
<xs:element name="label">
<xs:complexType>
<xs:sequence>
<xs:element ref="ns1:text"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:element>
<xs:element name="adr">
<xs:complexType>
<xs:sequence>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
</!-- 6.2.7 -->
<xs:element name="gender">
<xs:complexType>
<xs:sequence>
<xs:element ref="ns1:sex"/>
<xs:element minOccurs="0" ref="ns1:identity"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="sex">
<xs:simpleType>
<xs:restriction base="xs:token">
<xs:enumeration value="/"/>
<xs:enumeration value="M"/>
<xs:enumeration value="F"/>
<xs:enumeration value="O"/>
<xs:enumeration value="N"/>
<xs:enumeration value="U"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="identity" type="xs:string"/>
</!-- 6.3.1 -->
<xs:group name="param-label">
<xs:sequence>
<xs:element minOccurs="0" ref="ns1:label"/>
</xs:sequence>
</xs:group>
<xs:element name="label">
<xs:complexType>
<xs:sequence>
<xs:element ref="ns1:text"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:element>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:element>
</!-- 6.2.7 -->
<xs:element name="gender">
<xs:complexType>
<xs:sequence>
<xs:element ref="ns1:sex"/>
<xs:element minOccurs="0" ref="ns1:identity"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="sex">
<xs:simpleType>
<xs:restriction base="xs:token">
<xs:enumeration value="/"/>
<xs:enumeration value="M"/>
<xs:enumeration value="F"/>
<xs:enumeration value="O"/>
<xs:enumeration value="N"/>
<xs:enumeration value="U"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="identity" type="xs:string"/>
</!-- 6.3.1 -->
<xs:group name="param-label">
<xs:sequence>
<xs:element minOccurs="0" ref="ns1:label"/>
</xs:sequence>
</xs:group>
<xs:element name="label">
<xs:complexType>
<xs:sequence>
<xs:element ref="ns1:text"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:element>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:element>
</!-- 6.2.7 -->
<xs:element name="gender">
<xs:complexType>
<xs:sequence>
<xs:element ref="ns1:sex"/>
<xs:element minOccurs="0" ref="ns1:identity"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="sex">
<xs:simpleType>
<xs:restriction base="xs:token">
<xs:enumeration value="/"/>
<xs:element minOccurs="0" name="parameters">
  <xs:complexType>
    <xs:sequence>
      <xs:group ref="ns1:param-language"/>
      <xs:group ref="ns1:param-altid"/>
      <xs:group ref="ns1:param-pid"/>
      <xs:group ref="ns1:param-pref"/>
      <xs:group ref="ns1:param-type"/>
      <xs:group ref="ns1:param-geo"/>
      <xs:group ref="ns1:param-tz"/>
      <xs:group ref="ns1:param-label"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element maxOccurs="unbounded" ref="ns1:pobox"/>
<xs:element maxOccurs="unbounded" ref="ns1:ext"/>
<xs:element maxOccurs="unbounded" ref="ns1:street"/>
<xs:element maxOccurs="unbounded" ref="ns1:locality"/>
<xs:element maxOccurs="unbounded" ref="ns1:region"/>
<xs:element maxOccurs="unbounded" ref="ns1:code"/>
<xs:element maxOccurs="unbounded" ref="ns1:country"/>

<!-- 6.4.1 -->
<xs:element name="tel">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="parameters">
        <xs:complexType>
          <xs:sequence>
            <xs:group ref="ns1:param-altid"/>
            <xs:group ref="ns1:param-pid"/>
            <xs:group ref="ns1:param-pref"/>
            <xs:element minOccurs="0" name="type">
              <xs:complexType>
                <xs:sequence>
                  <xs:element maxOccurs="unbounded" name="text">
                    <xs:simpleType>
                      <xs:restriction base="xs:token">
                        <xs:enumeration value="work"/>
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:element>
<xs:enumeration value="home"/>
<xs:enumeration value="text"/>
<xs:enumeration value="voice"/>
<xs:enumeration value="fax"/>
<xs:enumeration value="cell"/>
<xs:enumeration value="video"/>
<xs:enumeration value="pager"/>
<xs:enumeration value="textphone"/>
</xs:restriction>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:complexType>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

```xml
<xs:complexType>
  <xs:sequence>
    <xs:group ref="ns1:param-altid"/>
    <xs:group ref="ns1:param-pid"/>
    <xs:group ref="ns1:param-pref"/>
    <xs:group ref="ns1:param-type"/>
    <xs:group ref="ns1:param-mediatype"/>
  </xs:sequence>
</xs:complexType>
</xs:element>

<xs:element ref="ns1:uri"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<!-- 6.4.4 -->
<xs:element name="lang">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="parameters">
        <xs:complexType>
          <xs:sequence>
            <xs:group ref="ns1:param-altid"/>
            <xs:group ref="ns1:param-pid"/>
            <xs:group ref="ns1:param-pref"/>
            <xs:group ref="ns1:param-type"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element ref="ns1:language-tag"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- 6.5.1 -->
<xs:group name="property-tz">
  <xs:element name="tz">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="parameters">
          <xs:complexType>
            <xs:sequence>
              <xs:group ref="ns1:param-altid"/>
              <xs:group ref="ns1:param-pid"/>
              <xs:group ref="ns1:param-pref"/>
              <xs:group ref="ns1:param-type"/>
              <xs:group ref="ns1:param-mediatype"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:group>
```
<xs:element minOccurs="0" name="parameters">
  <xs:complexType>
    <xs:sequence>
      <xs:group ref="ns1:param-language"/>
      <xs:group ref="ns1:param-altid"/>
      <xs:group ref="ns1:param-pid"/>
      <xs:group ref="ns1:param-pref"/>
      <xs:group ref="ns1:param-type"/>
      <xs:group ref="ns1:param-sort-as"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="member">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="parameters">
        <xs:complexType>
          <xs:sequence>
            <xs:group ref="ns1:param-altid"/>
            <xs:group ref="ns1:param-pid"/>
            <xs:group ref="ns1:param-pref"/>
            <xs:element minOccurs="0" name="mediatype"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element ref="ns1:uri"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- 6.6.5 -->

<xs:element name="related">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="parameters">
        <xs:complexType>
          <xs:sequence>
            <xs:group ref="ns1:param-altid"/>
            <xs:group ref="ns1:param-pid"/>
            <xs:group ref="ns1:param-pref"/>
            <xs:element minOccurs="0" name="type">
              <xs:complexType>
                <xs:sequence>
                  <xs:element maxOccurs="unbounded" name="text"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- 6.6.6 -->
<xs:element name="uid">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ns1:uri"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- 6.7.6 -->
<xs:element name="clientpidmap">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ns1:sourceid"/>
      <xs:element ref="ns1:uri"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="sourceid" type="xs:positiveInteger"/>

<!-- 6.7.8 -->
<xs:element name="url">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="parameters">
        <xs:complexType>
          <xs:sequence>
            <xs:group ref="ns1:param-altid"/>
            <xs:group ref="ns1:param-pid"/>
            <xs:group ref="ns1:param-pref"/>
            <xs:group ref="ns1:param-type"/>
            <xs:group ref="ns1:param-mediatype"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element ref="ns1:uri"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:element>
<!-- 6.8.1 -->
<xs:element name="key">
  <xs:complexType>
    <xs:element minOccurs="0" name="parameters">
      <xs:complexType>
        <xs:sequence>
          <xs:group ref="ns1:param-altid"/>
          <xs:group ref="ns1:param-pid"/>
          <xs:group ref="ns1:param-pref"/>
          <xs:group ref="ns1:param-type"/>
          <xs:group ref="ns1:param-mediatype"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:choice>
      <xs:element ref="ns1:uri"/>
      <xs:element ref="ns1:fburl"/>
      <xs:element ref="ns1:caladruri"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
<!-- 6.9.1 -->
<xs:element name="fburl">
  <xs:complexType>
    <xs:element minOccurs="0" name="parameters">
      <xs:complexType>
        <xs:sequence>
          <xs:group ref="ns1:param-altid"/>
          <xs:group ref="ns1:param-pid"/>
          <xs:group ref="ns1:param-pref"/>
          <xs:group ref="ns1:param-type"/>
          <xs:group ref="ns1:param-mediatype"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element ref="ns1:uri"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
<!-- 6.9.1 -->

<xs:sequence>
  <xs:group ref="ns1:param-altid"/>
  <xs:group ref="ns1:param-pid"/>
  <xs:group ref="ns1:param-pref"/>
  <xs:group ref="ns1:param-type"/>
  <xs:group ref="ns1:param-mediatype"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="caluri">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="parameters">
        <xs:complexType>
          <xs:sequence>
            <xs:group ref="ns1:param-altid"/>
            <xs:group ref="ns1:param-pid"/>
            <xs:group ref="ns1:param-pref"/>
            <xs:group ref="ns1:param-type"/>
            <xs:group ref="ns1:param-mediatype"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element ref="ns1:uri"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:complexType>
</xs:element>
<!-- 6.9.3 -->
<xs:element name="caluri">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="parameters">
        <xs:complexType>
          <xs:sequence>
            <xs:group ref="ns1:param-altid"/>
            <xs:group ref="ns1:param-pid"/>
            <xs:group ref="ns1:param-pref"/>
            <xs:group ref="ns1:param-type"/>
            <xs:group ref="ns1:param-mediatype"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element ref="ns1:uri"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:complexType>
</xs:element>
</xs:complexType>
<!-- Top-level grammar -->
<xs:group name="property">
  <xs:choice>
    <xs:element ref="ns1:adr"/>
    <xs:element ref="ns1:anniversary"/>
    <xs:element ref="ns1:bday"/>
    <xs:element ref="ns1:caladruri"/>
    <xs:element ref="ns1:caluri"/>
    <xs:element ref="ns1:categories"/>
    <xs:element ref="ns1:clientpidmap"/>
    <xs:element ref="ns1:email"/>
    <xs:element ref="ns1:fburl"/>
    <xs:element ref="ns1:fn"/>
    <xs:group ref="ns1:property-geo"/>
    <xs:element ref="ns1:impp"/>
    <xs:element ref="ns1:key"/>
  </xs:choice>
</xs:group>
<xs:group minOccurs="0" maxOccurs="unbounded" ref="ns1:property"/>
<xs:attribute name="name" use="required"/>
</xs:complexType>
</xs:element>
</xs:schema>

Authors' Addresses

Randall Gellens
Qualcomm Technologies, Inc.
5775 Morehouse Drive
San Diego, CA  92121
US
Email: rg+ietf@qti.qualcomm.com

Brian Rosen
NeuStar
470 Conrad Dr.
Mars, PA  16046
US
Phone: +1 724 382 1051
Email: br@brianrosen.net

Hannes Tschofenig
Hall in Tirol  6060
Austria
Email: Hannes.tschofenig@gmx.net
URI:   http://www.tschofenig.priv.at

Roger Marshall
TeleCommunication Systems, Inc.
2401 Elliott Avenue
Seattle, WA  98121
US
Phone: +1 206 792 2424
Email: rmarshall@telecomsys.com
URI:   http://www.telecomsys.com
Data-Only Emergency Calls
draft-ietf-ecrit-data-only-ea-09.txt

Abstract

RFC 6443 ‘Framework for Emergency Calling Using Internet Multimedia’ describes how devices use the Internet to place emergency calls and how Public Safety Answering Points (PSAPs) can handle Internet multimedia emergency calls natively. The exchange of multimedia traffic typically involves a SIP session establishment starting with a SIP INVITE that negotiates various parameters for that session.

In some cases, however, the transmission of application data is everything that is needed. Examples of such environments include temperature sensors issuing alerts, or vehicles sending crash data. Often these alerts are conveyed as one-shot data transmissions. These type of interactions are called ‘data-only emergency calls’. This document describes a container for the data based on the Common Alerting Protocol (CAP) and its transmission using the SIP MESSAGE transaction.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as “work in progress.”

This Internet-Draft will expire on April 30, 2015.
Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction .................................................. 3
2. Terminology .................................................. 3
3. Architectural Overview ........................................ 4
4. Protocol Specification ........................................ 6
   4.1. CAP Transport ........................................... 6
   4.2. Profiling of the CAP Document Content .................... 7
   4.3. Sending a Data-Only Emergency Call ...................... 8
5. Error Handling ................................................. 8
   5.1. 425 (Bad Alert Message) Response Code .................. 9
   5.2. The AlertMsg-Error Header Field ......................... 9
6. Updates to the CAP Message .................................... 11
7. Call Backs ..................................................... 11
8. Handling Large Amounts of Data ............................... 11
9. Example .......................................................... 11
10. Security Considerations ....................................... 15
11. IANA Considerations .......................................... 17
   11.1. Registration of the ‘application/emergencyCall.cap+xml’ MIME type ........................................ 17
   11.2. IANA Registration of Additional Data Block ............ 18
   11.3. IANA Registration for 425 Response Code ............... 18
   11.4. IANA Registration of New AlertMsg-Error Header Field ... 19
   11.5. IANA Registration for the SIP AlertMsg-Error Codes ... 19
12. Acknowledgments ................................................ 20
13. References .................................................... 20
   13.1. Normative References ................................... 20
   13.2. Informative References ................................ 21
Authors’ Addresses ................................................ 22
1. Introduction

RFC 6443 [RFC6443] describes how devices use the Internet to place emergency calls and how Public Safety Answering Points (PSAPs) can handle Internet multimedia emergency calls natively. The exchange of multimedia traffic typically involves a SIP session establishment starting with a SIP INVITE that negotiates various parameters for that session.

In some cases, however, there is only application data to be conveyed from the end devices to a PSAP or some other intermediary. Examples of such environments includes sensors issuing alerts, or vehicles sending crash data. These messages may be one-shot alerts to emergency authorities and do not require establishment of a session. These type of interactions are called ‘data-only emergency calls’. In this document, we use the term "call" so that similarities between full sessions with interactive media can be exploited.

Data-only emergency calls are similar to regular emergency calls in the sense that they require the emergency indications, emergency call routing functionality and may even have the same location requirements. However, the communication interaction will not lead to the exchange of interactive media, that is, Real-Time Protocol packets, such as voice, video data or real-time text.

The Common Alerting Protocol (CAP) [cap] is a document format for exchanging emergency alerts and public warnings. CAP is mainly used for conveying alerts and warnings between authorities and from authorities to citizen/individuals. This document is concerned with citizen to authority "alerts", where the alert is sent without any interactive media.

This document describes a method of including a CAP message in a SIP transaction, either by value (CAP message is in the body of the message, using a CID) or by reference (A URI is included in the message, which when dereferenced returns the CAP message) by defining it as a block of "additional data" as defined in [I-D.ietf-ecrit-additional-data]. The additional data mechanism is also used to send alert specific data beyond that available in the CAP message. This document also describes how a SIP MESSAGE [RFC3428] transaction can be used to send a data-only call.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].
3. Architectural Overview

This section illustrates two envisioned usage modes; targeted and location-based emergency alert routing.

1. Emergency alerts containing only data are targeted to a intermediary recipient responsible for evaluating the next steps. These steps could include:

   1. Sending a call containing only data toward a Public Safety Answering Point (PSAP);

   2. Establishing a third-party initiated emergency call towards a PSAP that could include audio, video, and data.

2. Emergency alerts may be targeted to a Service URN used for IP-based emergency calls where the recipient is not known to the originator. In this scenario, the alert may contain only data (e.g., a CAP, Geolocation header and one or more Call-Info headers containing Additional Data [I-D.ietf-ecrit-additional-data] in a SIP MESSAGE).

Figure 1 shows a deployment variant where a sensor, is pre-configured (using techniques outside the scope of this document) to issue an alert to an aggregator that processes these messages and performs whatever steps are necessary to appropriately react on the alert. For example, a security firm may use different sensor inputs to dispatch their security staff to a building they protect or to initiate a third-party emergency call.
In Figure 2 a scenario is shown whereby the alert is routed using location information and the Service URN. An emergency services routing proxy (ESRP) may use LoST to determine the next hop proxy to route the alert message to. A possible receiver is a PSAP and the recipient of the alert may be call taker. In the generic case, there is very likely no prior relationship between the originator and the receiver, e.g. PSAP. A PSAP, for example, is likely to receive and accept alerts from entities it cannot authorize. This scenario corresponds more to the classical emergency services use case and the description in [RFC6881] is applicable. In this use case, the only difference between an emergency call, and an emergency data-only call is that the former uses INVITE, creates a session and negotiates one or more media streams, while the latter uses MESSAGE, does not create a session and does not have media.
Figure 2: Location-Based Emergency Alert Routing

4. Protocol Specification

4.1. CAP Transport

A CAP message may be sent on the initial message of any SIP transaction. However, this document only describes specific behavior when used with a SIP INVITE that would accompany a normal emergency call and a SIP MESSAGE transaction for a one-shot, data-only emergency call. Behavior with other transactions is not defined.
The CAP message included in a SIP message as an additional-data block [I-D.ietf-ecrit-additional-data]. Accordingly, it is introduced to the SIP message with a Call-Info header with a purpose of "emergencyCall.cap". The header may contain a URI that is used by the recipient (or in some cases, an intermediary) to obtain the CAP message. Alternative, the Call-Info header may contain a Content Indirect url [RFC2392] and the CAP message included in the body of the message. In either case, the CAP message is located in a MIME block. The MIME type is set to 'application/emergencyCall.cap+xml'.

If the server does not support the functionality required to fulfill the request then a 501 Not Implemented MUST be returned as specified in RFC 3261 [RFC3261]. This is the appropriate response when a UAS does not recognize the request method and is not capable of supporting it for any user.

The 415 Unsupported Media Type error MUST be returned as specified in RFC 3261 [RFC3261] if the server is refusing to service the request because the message body of the request is in a format not supported by the server for the requested method. The server MUST return a list of acceptable formats using the Accept, Accept-Encoding, or Accept-Language header field, depending on the specific problem with the content.

4.2. Profiling of the CAP Document Content

The usage of CAP MUST conform to the specification provided with [cap]. For the usage with SIP the following additional requirements are imposed:

sender: A few sub-categories for putting a value in the <sender> element have to be considered:

Originator is a SIP entity, Author indication irrelevant: When the alert was created by a SIP-based originator and it is not useful to be explicit about the author of the alert then the <sender> element MUST be populated with the SIP URI of the user agent.

Originator is a non-SIP entity, Author indication irrelevant: In case that the alert was created by a non-SIP based entity and the identity of this original sender wants to be preserved then this identity MUST be placed into the <sender> element. In this category the it is not useful to be explicit about the author of the alert. The specific type of identity being used will depends on the technology being used by the original originator.
Author indication relevant: In case the author is different from the actual originator of the message and this distinction should be preserved then the <sender> element MUST NOT contain the SIP URI of the user agent.

incidents: The <incidents> element MUST be present. This incident identifier MUST be chosen in such a way that it is unique for a given <sender, expires, incidents> combination. Note that the <expires> element is optional and may not be present.

scope: The value of the <scope> element MAY be set to "Private" if the alert is not meant for public consumption. The <addresses> element is, however, not used by this specification since the message routing is performed by SIP and the respective address information is already available in other SIP headers. Populating information twice into different parts of the message may lead to inconsistency.

parameter: The <parameter> element MAY contain additional information specific to the sender.

area: It is RECOMMENDED to omit this element when constructing a message. In case that the CAP message already contained an <area> element then the specified location information SHOULD be copied into the PIDF-LO structure of the 'geolocation' header.

4.3. Sending a Data-Only Emergency Call

A data-only emergency call is sent using a SIP MESSAGE transaction with a CAP URI or body as described above in a manner similar to how an emergency call with interactive media is sent, as described in [RFC6881]. The MESSAGE transaction does not create a session or send media, but otherwise, the header content of the transaction, routing, and processing of data-only calls are the same as those of other emergency calls.

5. Error Handling

This section defines a new error response code and a header field for additional information.
5.1. 425 (Bad Alert Message) Response Code

This SIP extension creates a new location-specific response code, defined as follows,

425 (Bad Alert Message)

The 425 response code is a rejection of the request due to its included alert content, indicating that it was malformed or not satisfactory for the recipient’s purpose.

A SIP intermediary can also reject an alert it receives from a UA when it understands that the provided alert is malformed.

Section 5.2 describes an AlertMsg-Error header field with more details about what was wrong with the alert message in the request. This header field MUST be included in the 425 response.

It is only appropriate to generate a 425 response when the responding entity has no other information in the request that are usable by the responder.

A 425 response code MUST NOT be sent in response to a request that lacks an alert message entirely, as the user agent in that case may not support this extension at all.

A 425 response is a final response within a transaction, and MUST NOT terminate an existing dialog.

5.2. The AlertMsg-Error Header Field

The AlertMsg-Error header provides additional information about what was wrong with the original request. In some cases the provided information will be used for debugging purposes.

The AlertMsg-Error header field has the following ABNF [RFC5234]:

```
message-header /= AlertMsg-Error
                 ; (message-header from 3261)
AlertMsg-Error  = "AlertMsg-Error" HCOLON
                   ErrorValue
                   ErrorValue
                   ErrorValue
ErrorValue      = error-code
                   *(SEMI error-params)
error-code      = 1*3DIGIT
error-params    = error-code-text
                   / generic-param ; from RFC3261
error-code-text = "code" EQUAL quoted-string ; from RFC3261
```
HCOLON, SEMI, and EQUAL are defined in RFC3261 [RFC3261]. DIGIT is defined in RFC5234 [RFC5234].

The AlertMsg-Error header field MUST contain only one ErrorValue to indicate what was wrong with the alert payload the recipient determined was bad.

The ErrorValue contains a 3-digit error code indicating what was wrong with the alert in the request. This error code has a corresponding quoted error text string that is human understandable. The text string are OPTIONAL, but RECOMMENDED for human readability, similar to the string phrase used for SIP response codes. That said, the strings are complete enough for rendering to the user, if so desired. The strings in this document are recommendations, and are not standardized - meaning an operator can change the strings - but MUST NOT change the meaning of the error code. Similar to how RFC 3261 specifies, there MUST NOT be more than one string per error code.

The AlertMsg-Error header field MAY be included in any response as an alert message was in the request part of the same transaction. For example, a UA includes an alert in an MESSAGE to a PSAP. The PSAP can accept this MESSAGE, thus creating a dialog, even though his UA determined the alert message contained in the MESSAGE was bad. The PSAP merely includes an AlertMsg-Error header value in the 200 OK to the MESSAGE informing the UA that the MESSAGE was accepted but the alert provided was bad.

If, on the other hand, the PSAP cannot accept the transaction without a suitable alert message, a 425 response is sent.

A SIP intermediary that requires the UA’s alert message in order to properly process the transaction may also sends a 425 with an AlertMsg-Error code.

This document defines an initial list of error code ranges for any SIP response, including provisional responses (other than 100 Trying) and the new 425 response. There MUST be no more than one AlertMsg-Error code in a SIP response.

AlertMsg-Error: 100 ; code="Cannot Process the Alert Payload"

AlertMsg-Error: 101 ; code="Alert Payload was not present or could not be found"

AlertMsg-Error: 102 ; code="Not enough information to determine the purpose of the alert"
AlertMsg-Error: 103 ; code="Alert Payload was corrupted"

Additionally, if an entity cannot or chooses not to process the alert message from a SIP request, a 500 (Server Internal Error) SHOULD be used with or without a configurable Retry-After header field.

6. Updates to the CAP Message

If the sender anticipates that the content of the CAP message may need to be updated during the lifecycle of the event referred to in the message, it may include an update block as defined in [I-D.rosen-ecrit-addldata-subnot].

7. Call Backs

This document does not describe any method for the recipient to call back the sender of the data-only call. Usually, these alerts are sent by automata, and do not have any mechanism to receive calls of any kind. The identifier in the From header may be useful to obtain more information, but any such mechanism is not defined in this document. The CAP message may contain related contact information for the sender.

8. Handling Large Amounts of Data

It is not atypical for sensor to have large quantities of data that they may wish to send. Including large amounts of data in a MESSAGE is not advisable, because SIP entities are usually not equipped to handle very large messages. In such cases, the sender SHOULD make use of the by-reference mechanisms defined for Additional Data which involve sending a URI in the Call-Info header and using HTTPS to retrieve the data. The CAP message itself can be sent by-reference using this mechanism as well as any or all of the Additional Data blocks that may contain sensor-specific data.

9. Example

Figure 3 shows a CAP document indicating a BURGLARY alert issued by a sensor called ‘sensor1@domain.com’. The location of the sensor can be obtained from the attached location information provided via the ‘geolocation’ header contained in the SIP MESSAGE structure. Additionally, the sensor provided some data long with the alert message using proprietary information elements only to be processed by the receiver, a SIP entity acting as an aggregator. This example reflects the description in Figure 1.

MESSAGE sip:aggregator@domain.com SIP/2.0
Via: SIP/2.0/TCP sensor1.domain.com;branch=z9hG4bK776sgdkse
Max-Forwards: 70
From: sip:sensor1@domain.com;tag=49583
To: sip:aggregator@domain.com
Call-ID: asd88asd77a@1.2.3.4
Geolocation: <cid:abcdef@domain.com>;routing-allowed=yes
Supported: geolocation
Accept: application/pidf+xml, application/emergencyCall.cap+xml
CSeq: 1 MESSAGE
Call-Info: cid:abcdef2@domain.com;purpose=emergencyCall.cap
Content-Type: multipart/mixed; boundary=boundary1
Content-Length: ...

--boundary1
Content-Type: application/emergencyCall.cap
Content-ID: <abcdef2@domain.com>
Content-Disposition: by-reference; handling=optional
<?xml version="1.0" encoding="UTF-8"?>
<alert xmlns="urn:oasis:names:tc:emergency:cap:1.1">
  <identifier>S-1</identifier>
  <sender>sip:sensor1@domain.com</sender>
  <sent>2008-11-19T14:57:00-07:00</sent>
  <status>Actual</status>
  <msgType>Alert</msgType>
  <scope>Private</scope>
  <incidents>abc1234</incidents>
  <info>
    <category>Security</category>
    <event>BURGLARY</event>
    <urgency>Expected</urgency>
    <certainty>Likely</certainty>
    <severity>Moderate</severity>
    <senderName>SENSOR 1</senderName>
    <parameter>
      <valueName>SENSOR-DATA-NAMESPACE1</valueName>
      <value>123</value>
    </parameter>
    <parameter>
      <valueName>SENSOR-DATA-NAMESPACE2</valueName>
      <value>TRUE</value>
    </parameter>
  </info>
</alert>
--boundary1
Content-Type: application/pidf+xml
Content-ID: <abcdef2@domain.com>
Content-Disposition: by-reference; handling=optional

<?xml version="1.0" encoding="UTF-8"?>
<presence
    xmlns="urn:ietf:params:xml:ns:pidf"
    xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
    xmlns:gml="http://www.opengis.net/gml"
    entity="pres:alice@atlanta.example.com">
    <dm:device id="sensor">
        <gp:geopriv>
            <gp:location-info>
                <gml:location>
                    <gml:Point srsName="urn:ogc:def:crs:EPSG::4326">
                        <gml:pos>32.86726 -97.16054</gml:pos>
                    </gml:Point>
                </gml:location>
            </gp:location-info>
            <gp:usage-rules>
                <gbp:retransmission-allowed>false</gbp:retransmission-allowed>
                <gbp:retention-expiry>2010-11-14T20:00:00Z</gbp:retention-expiry>
            </gp:usage-rules>
            <gp:method>802.11</gp:method>
        </gp:geopriv>
        <dm:timestamp>2010-11-04T20:57:29Z</dm:timestamp>
    </dm:device>
</presence>
--boundary1--

Figure 3: Example Message conveying an Alert to an Aggregator

Figure 4 shows the same CAP document sent as a data-only emergency call towards a PSAP.

MESSAGE urn:service:sos SIP/2.0
Via: SIP/2.0/TCP sip:aggreg.1.example.com;branch=z9hG4bK776abssa
Max-Forwards: 70
From: sip:aggregator@example.com;tag=32336
To: 112
Call-ID: asdf33443a@example.com
Route: sip:psap1.example.gov
Geolocation: <cid:abcdef@example.com>
; routing-allowed=yes
Supported: geolocation
Accept: application/pidf+xml, application/emergencyCall.cap+xml
Call-info: cid:abcdef2@domain.com; purpose=emergencyCall.cap
CSeq: 1 MESSAGE
Content-Type: multipart/mixed; boundary=boundary1
Content-Length: ...

--boundary1

Content-Type: application/emergencyCall.cap+xml
Content-ID: <abcdef2@domain.com>
<?xml version="1.0" encoding="UTF-8"?>
<alert xmlns="urn:oasis:names:tc:emergency:cap:1.1">
  <identifier>S-1</identifier>
  <sender>sip:sensor1@domain.com</sender>
  <sent>2008-11-19T14:57:00-07:00</sent>
  <status>Actual</status>
  <msgType>Alert</msgType>
  <scope>Private</scope>
  <incidents>abc1234</incidents>
  <info>
    <category>Security</category>
    <event>BURGLARY</event>
    <urgency>Expected</urgency>
    <certainty>Likely</certainty>
    <severity>Moderate</severity>
    <senderName>SENSOR 1</senderName>
    <parameter>
      <valueName>SENSOR-DATA-NAMESPACE1</valueName>
      <value>123</value>
    </parameter>
    <parameter>
      <valueName>SENSOR-DATA-NAMESPACE2</valueName>
      <value>TRUE</value>
    </parameter>
  </info>
</alert>

--boundary1

Content-Type: application/pidf+xml
Content-ID: <abcdef2@domain.com>
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf">
  
</presence>
Figure 4: Example Message conveying an Alert to a PSAP

10. Security Considerations

This section discusses security considerations when SIP user agents issue emergency alerts utilizing MESSAGE and CAP. Location specific threats are not unique to this document and are discussed in [I-D.ietf-ecrit-trustworthy-location] and [RFC6442].

The ECRIT emergency services architecture [RFC6443] considers classical individual-to-authority emergency calling and the identity of the emergency caller does not play a role at the time of the call establishment itself, i.e., a response to the emergency call will not depend on the identity of the caller. In case of emergency alerts generated by devices, like sensors, the processing may be different in order to reduce the number of falsely generated emergency alerts. Alerts may get triggered based on certain sensor input that may have been caused by other factors than the actual occurrence of an alert relevant event. For example, a sensor may simply be malfunctioning.
For this purpose not all alert messages are directly sent to a PSAP but rather may be pre-processed by a separate entity, potentially under supervision by a human, to filter alerts and potentially correlate received alerts with others to obtain a larger picture of the ongoing situation.

In any case, for alerts that are initiated by sensors the identity may play an important role in deciding whether to accept or ignore an incoming alert message. With the scenario shown in Figure 1 it is very likely that only authorized sensor input will be processed. For this purpose it needs to be ensured that no alert messages from an unknown origin are accepted. Two types of information elements can be used for this purpose:

1. SIP itself provides security mechanisms that allow the verification of the originator’s identity. These mechanisms can be re-used, such as P-Asserted-Identity [RFC3325] or SIP Identity [RFC4474]. The latter provides a cryptographic assurance while the former relies on a chain of trust model.

2. CAP provides additional security mechanisms and the ability to carry additional information about the sender’s identity. Section 3.3.2.1 of [cap] specifies the signing algorithms of CAP documents.

In addition to the desire to perform identity-based access control the classical communication security threats need to be considered, including integrity protection to prevent forgery and replay of alert messages in transit. To deal with replay of alerts a CAP document contains the mandatory <identifier>, <sender>, <sent> elements and an optional <expire> element. These attributes make the CAP document unique for a specific sender and provide time restrictions. An entity that has received a CAP message already within the indicated timeframe is able to detect a replayed message and, if the content of that message is unchanged, then no additional security vulnerability is created. Additionally, it is RECOMMENDED to make use of SIP security mechanisms, such as SIP Identity [RFC4474], to tie the CAP message to the SIP message. To provide protection of the entire SIP message exchange between neighboring SIP entities the usage of TLS is mandatory.

Note that none of the security mechanism in this document protect against a compromised sensor sending crafted alerts.
11. IANA Considerations

11.1. Registration of the ‘application/emergencyCall.cap+xml’ MIME type

To: ietf-types@iana.org

Subject: Registration of MIME media type application/emergencyCall.cap+xml

MIME media type name: application

MIME subtype name: cap+xml

Required parameters: (none)

Optional parameters: charset; Indicates the character encoding of enclosed XML. Default is UTF-8 [RFC3629].

Encoding considerations: Uses XML, which can employ 8-bit characters, depending on the character encoding used. See RFC 3023 [RFC3023], Section 3.2.

Security considerations: This content type is designed to carry payloads of the Common Alerting Protocol (CAP).

Interoperability considerations: This content type provides a way to convey CAP payloads.

Published specification: RFC XXX [Replace by the RFC number of this specification].

Applications which use this media type: Applications that convey alerts and warnings according to the CAP standard.
Additional information: OASIS has published the Common Alerting Protocol at http://www.oasis-open.org/committees/documents.php&wg_abbrev=emergency

Person and email address to contact for further information: Hannes Tschofenig, Hannes.Tschofenig@nsn.com

Intended usage: Limited use

Author/Change controller: IETF ECRIT working group

Other information: This media type is a specialization of application/xml RFC 3023 [RFC3023], and many of the considerations described there also apply to application/cap+xml.

11.2. IANA Registration of Additional Data Block

This document registers a new block type in the sub-registry called ‘Additional Data Blocks’ defined in [I-D.ietf-ecrit-additional-data]. The token is "cap" and the reference is this document.

11.3. IANA Registration for 425 Response Code

In the SIP Response Codes registry, the following is added

Reference: RFC-XXXX (i.e., this document)
Response code: 425 (recommended number to assign)
Default reason phrase: Bad Alert Message

<table>
<thead>
<tr>
<th>Registry</th>
<th>Response Code</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request Failure 4xx</td>
<td>425 Bad Alert Message</td>
<td>[this doc]</td>
</tr>
</tbody>
</table>

This SIP Response code is defined in Section 5.
11.4. IANA Registration of New AlertMsg-Error Header Field

The SIP AlertMsg-error header field is created by this document, with its definition and rules in Section 5, to be added to the IANA sip-parameters registry with two actions:

1. Update the Header Fields registry with

<table>
<thead>
<tr>
<th>Registry:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Name</td>
<td>compact Reference</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>AlertMsg-Error</td>
<td>[this doc]</td>
</tr>
</tbody>
</table>

2. In the portion titled "Header Field Parameters and Parameter Values", add

<table>
<thead>
<tr>
<th>Header Field</th>
<th>Parameter Name</th>
<th>Predefined Values</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlertMsg-Error</td>
<td>code</td>
<td>yes</td>
<td>[this doc]</td>
</tr>
</tbody>
</table>

11.5. IANA Registration for the SIP AlertMsg-Error Codes

This document creates a new registry for SIP, called "AlertMsg-Error Codes". AlertMsg-Error codes provide reason for the error discovered by recipients, categorized by action to be taken by error recipient. The initial values for this registry are shown below.

Registry Name: AlertMsg-Error Codes

Reference: [this doc]

Registration Procedures: Specification Required
100  "Cannot Process the Alert Payload" [this doc]
101  "Alert Payload was not present or could not be found" [this doc]
102  "Not enough information to determine the purpose of the alert" [this doc]
103  "Alert Payload was corrupted" [this doc]

Details of these error codes are in Section 5.

12. Acknowledgments

The authors would like to thank the participants of the Early Warning adhoc meeting at IETF#69 for their feedback. Additionally, we would like to thank the members of the NENA Long Term Direction Working Group for their feedback.

Additionally, we would like to thank Martin Thomson, James Winterbottom, Shida Schubert, Bernard Aboba, and Marc Linsner for their review comments.

13. References

13.1. Normative References


13.2. Informative References

[I-D.ietf-ecrit-trustworthy-location]
Tschofenig, H., Schulzrinne, H., and B. Aboba,


Authors’ Addresses

Brian Rosen
NeuStar, Inc.
470 Conrad Dr
Mars, PA 16046
US
Email: br@brianrosen.net

Henning Schulzrinne
Columbia University
Department of Computer Science
450 Computer Science Building
New York, NY 10027
US
Phone: +1 212 939 7004
Email: hgs+ecrit@cs.columbia.edu
URI: http://www.cs.columbia.edu

Hannes Tschofenig
Nokia Siemens Networks
Linnoitustie 6
Espoo 02600
Finland
Phone: +358 (50) 4871445
Email: Hannes.Tschofenig@gmx.net
URI: http://www.tschofenig.priv.at
Abstract

The trustworthiness of location information is critically important for some location-based applications, such as emergency calling or roadside assistance.

This document describes threats relating to conveyance of location in an emergency call, and describes techniques that improve the reliability and security of location information conveyed in an IP-based emergency service call. It also provides guidelines for assessing the trustworthiness of location information.
Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction ................................................. 3
   1.1. Terminology ........................................... 3
   1.2. Emergency Services Architecture ......................... 5
2. Threat Models ................................................. 8
   2.1. Existing Work ......................................... 8
   2.2. Adversary Model ....................................... 9
   2.3. Location Spoofing ..................................... 10
   2.4. Identity Spoofing ..................................... 10
3. Mitigation Techniques ......................................... 11
   3.1. Signed Location-by-Value ............................. 11
   3.2. Location-by-Reference ............................... 15
   3.3. Proxy Adding Location ................................ 18
4. Location Trust Assessment ..................................... 19
5. Security Considerations ....................................... 22
6. Privacy Considerations ....................................... 23
7. IANA Considerations ......................................... 25
8. References .................................................. 25
   8.1. Informative references ............................... 25
Acknowledgments ............................................... 28
Authors’ Addresses .............................................. 29
1. Introduction

Several public and commercial services depend upon location information in their operations. This includes emergency services (such as fire, ambulance and police) as well as commercial services such as food delivery and roadside assistance.

For circuit-switched calls from landlines, as well as for Voice over IP (VoIP) services only supporting emergency service calls from stationary devices, location provided to the Public Safety Answering Point (PSAP) is determined from a lookup using the calling telephone number. As a result, for landlines or stationary VoIP, spoofing of caller identification can result in the PSAP incorrectly determining the caller's location. Problems relating to calling party number and Caller ID assurance have been analyzed by the "Secure Telephone Identity Revisited" [STIR] Working Group as described in "Secure Telephone Identity Problem Statement and Requirements" [I-D.ietf-stir-problem-statement]. In addition to the work underway in STIR, other mechanisms exist for validating caller identification. For example, as noted in [EENA], one mechanism for validating caller identification information (as well as the existence of an emergency) is for the PSAP to call the user back, as described in [RFC7090].

Given the existing work on caller identification, this document focuses on the additional threats that are introduced by the support of IP-based emergency services in nomadic and mobile devices, in which location may be conveyed to the PSAP within the emergency call. Ideally, a call taker at a PSAP should be able to assess, in real-time, the level of trust that can be placed on the information provided within a call. This includes automated location conveyed along with the call and location information communicated by the caller, as well as identity information relating to the caller or the device initiating the call. Where real-time assessment is not possible, it is important to be able to determine the source of the call in a post-incident investigation, so as to be able to enforce accountability.

This document defines terminology (including the meaning of "trustworthy location") in Section 1.1, reviews existing work in Section 1.2, describes the threat model in Section 2, outlines potential mitigation techniques in Section 3, covers trust assessment in Section 4 and discusses security considerations in Section 5.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].
The definitions of "Internet Access Provider (IAP)", "Internet Service Provider (ISP)" and "Voice Service Provider (VSP)" are taken from "Requirements for Emergency Context Resolution with Internet Technologies" [RFC5012].

The definition of a "hoax call" is taken from "False Emergency Calls" [EENA].

The definition of "Device", "Target" and "Location Information Server" (LIS) is taken from "An Architecture for Location and Location Privacy in Internet Applications" [RFC6280], Section 7.

The term "Device" denotes the physical device, such as a mobile phone, PC, or embedded micro-controller, whose location is tracked as a proxy for the location of a Target.

The term "Target" denotes an individual or other entity whose location is sought in the Geopriv architecture. In many cases, the Target will be the human user of a Device, or it may be an object such as a vehicle or shipping container to which a Device is attached. In some instances, the Target will be the Device itself. The Target is the entity whose privacy Geopriv seeks to protect.

The term "Location Information Server" denotes an entity responsible for providing devices within an access network with information about their own locations. A Location Information Server uses knowledge of the access network and its physical topology to generate and distribute location information to devices.

The term "location determination method" refers to the mechanism used to determine the location of a Target. This may be something employed by a location information server (LIS), or by the Target itself. It specifically does not refer to the location configuration protocol (LCP) used to deliver location information either to the Target or the Recipient. This term is re-used from "GEOPRIV PIDF-LO Usage Clarification, Considerations, and Recommendations" [RFC5491].

The term "source" is used to refer to the LIS, node, or device from which a Recipient (Target or Third-Party) obtains location information.

Additionally, the terms Location-by-Value (LbyV), Location-by-Reference (LbyR), Location Configuration Protocol, Location Dereference Protocol, and Location Uniform Resource Identifier (URI) are re-used from "Requirements for a Location-by-Reference Mechanism" [RFC5808].

"Trustworthy Location" is defined as location information that can be
attributed to a trusted source, has been protected against
modification in transmit, and has been assessed as trustworthy.

"Location Trust Assessment" refers to the process by which the
reliability of location information can be assessed. This topic is
discussed in Section 4.

"Identity Spoofing" is where the attacker forges or obscures their
identity so as to prevent themselves from being identified as the
source of the attack. One class of identity spoofing attack involves
the forging of call origin identification.

The following additional terms apply to location spoofing:

"Place Shifting" is where the attacker constructs a Presence
Information Data Format Location Object (PIDF-LO) for a location
other than where they are currently located. In some cases, place
shifting can be limited in range (e.g., within the coverage area of a
particular cell tower).

"Time Shifting" is where the attacker uses or re-uses location
information that was valid in the past, but is no longer valid
because the attacker has moved.

"Location Theft" is where the attacker captures a Target’s location
information (possibly including a signature) and presents it as their
own. Location theft can occur in a single instance, or may be
continuous (e.g., where the attacker has gained control over the
victim’s device). Location theft may also be combined with time
shifting to present someone else’s location information after the
original Target has moved.

1.2. Emergency Services Architecture

This section describes how location is utilized in the Internet
Emergency Services Architecture, as well as the existing work on the
problem of hoax calls.

1.2.1. Location

The Internet architecture for emergency calling is described in
"Framework for Emergency Calling Using Internet Multimedia"
[RFC6443]. Best practices for utilizing the architecture to make
emergency calls are described in "Best Current Practice for
Communications Services in Support of Emergency Calling" [RFC6881].

As noted in "An Architecture for Location and Location Privacy in
Internet Applications" [RFC6280] Section 6.3:
"there are three critical steps in the placement of an emergency call, each involving location information:

1. Determine the location of the caller.

2. Determine the proper Public Safety Answering Point (PSAP) for the caller’s location.

3. Send a SIP INVITE message, including the caller’s location, to the PSAP."

The conveyance of location information within the Session Initiation Protocol (SIP) is described in "Location Conveyance for the Session Initiation Protocol" [RFC6442]. Conveyance of Location-by-Value (LbyV) as well as Location-by-Reference (LbyR) are supported. The Security Considerations (Section 7) discusses privacy, authentication and integrity concerns relating to conveyed location. This includes discussion of transmission layer security for confidentiality and integrity protection of SIP, as well as (undeployed) end-to-end security mechanisms for protection of location information (e.g. S/MIME). Regardless of whether transmission-layer security is utilized, location information may be available for inspection by an intermediary which, if it decides that the location value is unacceptable or insufficiently accurate, may send an error indication or replace the location, as described in [RFC6442] Section 3.4.

Although the infrastructure for location-based routing described in [RFC6443] was developed for use in emergency services, [RFC6442] supports conveyance of location within non-emergency calls as well as emergency calls. "Implications of ‘retransmission-allowed’ for SIP Location Conveyance" [RFC5606] Section 1 describes the overall architecture, as well as non-emergency usage scenarios:

The Presence Information Data Format for Location Objects (PIDF-LO [RFC4119]) carries both location information (LI) and policy information set by the Rule Maker, as is stipulated in [RFC3693]. The policy carried along with LI allows the Rule Maker to restrict, among other things, the duration for which LI will be retained by recipients and the redistribution of LI by recipients.

The Session Initiation Protocol [RFC3261] is one proposed Using Protocol for PIDF-LO. The conveyance of PIDF-LO within SIP is specified in [RFC6442]. The common motivation for providing LI in SIP is to allow location to be considered in routing the SIP message. One example use case would be emergency services, in which the location will be used by dispatchers to direct the response. Another use case might be providing location to be used by services associated with the SIP session; a location associated
with a call to a taxi service, for example, might be used to route
to a local franchisee of a national service and also to route the
taxi to pick up the caller.

1.2.2. Hoax Calls

Hoax calls have been a problem for emergency services dating back to
the time of street corner call boxes. As the European Emergency
Number Association (EENA) has noted [EENA]: "False emergency calls
divert emergency services away from people who may be in life-
threatening situations and who need urgent help. This can mean the
difference between life and death for someone in trouble."

EENA [EENA] has attempted to define terminology and describe best
current practices for dealing with false emergency calls. Reducing
the number of hoax calls represents a challenge, since emergency
services authorities in most countries are required to answer every
call (whenever possible). Where the caller cannot be identified, the
ability to prosecute is limited.

A particularly dangerous form of hoax call is "swatting" - a hoax
emergency call that draws a response from law enforcement prepared
for a violent confrontation (e.g. a fake hostage situation that
results in dispatching of a "Special Weapons And Tactics" (SWAT)
team). In 2008 the Federal Bureau of Investigation (FBI) issued a
warning [Swatting] about an increase in the frequency and
sophistication of these attacks.

As noted in [EENA], many documented cases of "swatting" involve not
only the faking of an emergency, but also falsification or
obfuscation of identity. There are a number of techniques by which
hoax callers attempt to avoid identification, and in general, the
ability to identify the caller appears to influence the incidence of
hoax calls.

Where a Voice Service Provider enables setting of the outbound caller
identification without checking it against the authenticated
identity, forging caller identification is trivial. Similarly where
an attacker can gain entry to a Private Branch Exchange (PBX), they
can then subsequently use that access to launch a denial of service
attack against the PSAP, or to make fraudulent emergency calls.
Where emergency calls have been allowed from handsets lacking a SIM
card, or where ownership of the SIM card cannot be determined, the
frequency of hoax calls has often been unacceptably high
[TASMANIA][UK][SA].

However, there are few documented cases of hoax calls that have
arisen from conveyance of untrustworthy location information within
2. Threat Models

This section reviews existing analyses of the security of emergency services, threats to geographic location privacy, threats relating to spoofing of caller identification and modification of location information in transit. In addition, the threat model applying to this work is described.

2.1. Existing Work

"An Architecture for Location and Location Privacy in Internet Applications" [RFC6280] describes an architecture for privacy-preserving location-based services in the Internet, focusing on authorization, security and privacy requirements for the data formats and protocols used by these services.

Within the Security Considerations (Section 5), mechanisms for ensuring the security of the location distribution chain are discussed; these include mechanisms for hop-by-hop confidentiality and integrity protection as well as end-to-end assurance.

"Geopriv Requirements" [RFC3693] focuses on the authorization, security and privacy requirements of location-dependent services, including emergency services. Within the Security Considerations (Section 8), this includes discussion of emergency services authentication (Section 8.3), and issues relating to identity and anonymity (Section 8.4).

"Threat Analysis of the Geopriv Protocol" [RFC3694] describes threats against geographic location privacy, including protocol threats, threats resulting from the storage of geographic location data, and threats posed by the abuse of information.

"Security Threats and Requirements for Emergency Call Marking and Mapping" [RFC5069] reviews security threats associated with the marking of signaling messages and the process of mapping locations to Universal Resource Identifiers (URIs) that point to PSAPs. RFC 5069 describes attacks on the emergency services system, such as attempting to deny system services to all users in a given area, to gain fraudulent use of services and to divert emergency calls to non-emergency sites. In addition, it describes attacks against individuals, including attempts to prevent an individual from receiving aid, or to gain information about an emergency, as well as attacks on emergency services infrastructure elements, such as mapping discovery and mapping servers.
"Secure Telephone Identity Threat Model" [I-D.ietf-stir-threats] analyzes threats relating to impersonation and obscuring of calling party numbers, reviewing the capabilities available to attackers, and the scenarios in which attacks are launched.

2.2. Adversary Model

To provide a structured analysis we distinguish between three adversary models:

External adversary model: The end host, e.g., an emergency caller whose location is going to be communicated, is honest and the adversary may be located between the end host and the location server or between the end host and the PSAP. None of the emergency service infrastructure elements act maliciously.

Malicious infrastructure adversary model: The emergency call routing elements, such as the Location Information Server (LIS), the Location-to-Service Translation (LoST) infrastructure, used for mapping locations to PSAP address, or call routing elements, may act maliciously.

Malicious end host adversary model: The end host itself acts maliciously, whether the owner is aware of this or whether it is acting under the control of a third party.

Since previous work describes attacks against infrastructure elements (e.g. location servers, call route servers, mapping servers) or the emergency services IP network, as well as threats from attackers attempting to snoop location in transit, this document focuses on the threats arising from end hosts providing false location information within emergency calls (the malicious end host adversary model).

Since the focus is on malicious hosts, we do not cover threats that may arise from attacks on infrastructure that hosts depend on to obtain location. For example, end hosts may obtain location from civilian GPS, which is vulnerable to spoofing [GPSCounter] or from third party Location Service Providers (LSPs) which may be vulnerable to attack or may not provide location accuracy suitable for emergency purposes.

Also, we do not cover threats arising from inadequate location infrastructure. For example, a stale wiremap or an inaccurate access point location database could be utilized by the Location Information Server (LIS) or the end host in its location determination, thereby leading to an inaccurate determination of location. Similarly, a Voice Service Provider (VSP) (and indirectly a LIS) could utilize the wrong identity (such as an IP address) for location lookup, thereby
providing the end host with misleading location information.

2.3. Location Spoofing

Where location is attached to the emergency call by an end host, the end host can fabricate a PIDF-LO and convey it within an emergency call. The following represent examples of location spoofing:

Place shifting: Trudy, the adversary, pretends to be at an arbitrary location.

Time shifting: Trudy pretends to be at a location she was a while ago.

Location theft: Trudy observes or obtains Alice’s location and replays it as her own.

2.4. Identity Spoofing

While this document does not focus on the problems created by determination of location based on spoofed caller identification, the ability to ascertain identity is important, since the threat of punishment reduces hoax calls. As an example, calls from pay phones are subject to greater scrutiny by the call taker.

With calls originating on an IP network, at least two forms of identity are relevant, with the distinction created by the split between the IAP and the VSP:

(a) network access identity such as might be determined via authentication (e.g., using the Extensible Authentication Protocol (EAP) [RFC3748]);

(b) caller identity, such as might be determined from authentication of the emergency caller at the VoIP application layer.

If the adversary did not authenticate itself to the VSP, then accountability may depend on verification of the network access identity. However, this also may not have been authenticated, such as in the case where an open IEEE 802.11 Access Point is used to initiate a hoax emergency call. Although endpoint information such as the IP or MAC address may have been logged, tying this back to the device owner may be challenging.

Unlike the existing telephone system, VoIP emergency calls can provide an identity that need not necessarily be coupled to a business relationship with the IAP, ISP or VSP. However, due to the time-critical nature of emergency calls, multi-layer authentication
is undesirable, so that in most cases, only the device placing the call will be able to be identified. Furthermore, deploying additional credentials for emergency service purposes (such as certificates) increases costs, introduces a significant administrative overhead and is only useful if widely deployed.

3. Mitigation Techniques

The sections that follow present three mechanisms for mitigating the threats presented in Section 2:

1. Signed location by value (Section 3.1), which provides for authentication and integrity protection of the PIDF-LO. At the time of this writing, there is only an expired straw-man proposal for this mechanism [I-D.thomson-geopriv-location-dependability], so that it is not suitable for deployment.

2. Location-by-reference (Section 3.2), which enables location to be obtained by the PSAP directly from the location server, over a confidential and integrity-protected channel, avoiding modification by the end-host or an intermediary. This mechanism is specified in [RFC6753].

3. Proxy added location (Section 3.3), which protects against location forgery by the end host. This mechanism is specified in [RFC6442].

3.1. Signed Location-by-Value

With location signing, a location server signs the location information before it is sent to the Target. The signed location information is then sent to the location recipient, who verifies it.

Figure 1 shows the communication model with the target requesting signed location in step (a), the location server returns it in step (b) and it is then conveyed to the location recipient in step (c) who verifies it. For SIP, the procedures described in "Location Conveyance for the Session Initiation Protocol" [RFC6442] are applicable for location conveyance.
A straw-man proposal for location signing is provided in "Digital Signature Methods for Location Dependability" [I-D.thomson-geopriv-location-dependability]. Note that since this document is no longer under development, location signing cannot be considered deployable at the time of this writing.

In order to limit replay attacks, this document proposes the addition of a "validity" element to the PIDF-LO, including a "from" sub-element containing the time that location information was validated by the signer, as well as an "until" sub-element containing the last time that the signature can be considered valid.

One of the consequences of including an "until" element is that even a stationary target would need to periodically obtain a fresh PIDF-LO, or incur the additional delay of querying during an emergency call.

Although privacy-preserving procedures may be disabled for emergency calls, by design, PIDF-LO objects limit the information available for real-time attribution. As noted in [RFC5985] Section 6.6:

The LIS MUST NOT include any means of identifying the Device in the PIDF-LO unless it is able to verify that the identifier is correct and inclusion of identity is expressly permitted by a Rule Maker. Therefore, PIDF parameters that contain identity are either omitted or contain unlinked pseudonyms [RFC3693]. A unique, unlinked presentity URI SHOULD be generated by the LIS for the mandatory presence "entity" attribute of the PIDF document.
Optional parameters such as the "contact" and "deviceID" elements [RFC4479] are not used.

Also, the device referred to in the PIDF-LO may not necessarily be the same entity conveying the PIDF-LO to the PSAP. As noted in [RFC6442] Section 1:

In no way does this document assume that the SIP user agent client that sends a request containing a location object is necessarily the Target. The location of a Target conveyed within SIP typically corresponds to that of a device controlled by the Target, for example, a mobile phone, but such devices can be separated from their owners, and moreover, in some cases, the user agent may not know its own location.

Without the ability to tie the target identity to the identity asserted in the SIP message, it is possible for an attacker to cut and paste a PIDF-LO obtained by a different device or user into a SIP INVITE and send this to the PSAP. This cut and paste attack could succeed even when a PIDF-LO is signed, or [RFC4474] is implemented.

To address location-spoofing attacks, [I-D.thomson-geopriv-location-dependability] proposes addition of an "identity" element which could include a SIP URI (enabling comparison against the identity asserted in the SIP headers) or an X.509v3 certificate. If the target was authenticated by the LIS, an "authenticated" attribute is added. However, inclusion of an "identity" attribute could enable location tracking, so that a "hash" element is also proposed which could contain a hash of the content of the "identity" element instead. In practice, such a hash would not be much better for real-time validation than a pseudonym.

Location signing cannot deter attacks in which valid location information is provided. For example, an attacker in control of compromised hosts could launch a denial-of-service attack on the PSAP by initiating a large number of emergency calls, each containing valid signed location information. Since the work required to verify the location signature is considerable, this could overwhelm the PSAP infrastructure.

However, while DDOS attacks are unlikely to be deterred by location signing, accurate location information would limit the subset of compromised hosts that could be used for an attack, as only hosts within the PSAP serving area would be useful in placing emergency calls.

Location signing is also difficult when the host obtains location via mechanisms such as GPS, unless trusted computing approaches, with
tamper-proof GPS modules, can be applied. Otherwise, an end host can pretend to have a GPS device, and the recipient will need to rely on its ability to assess the level of trust that should be placed in the end host location claim.

Even though location signing mechanisms have not been standardized, [NENA-12] Section 3.7 includes operational recommendations relating to location signing:

Location determination is out of scope for NENA, but we can offer guidance on what should be considered when designing mechanisms to report location:

1. The location object should be digitally signed.

2. The certificate for the signer (LIS operator) should be rooted in VESA. For this purpose, VPC and ERDB operators should issue certs to LIS operators.

3. The signature should include a timestamp.

4. Where possible, the Location Object should be refreshed periodically, with the signature (and thus the timestamp) being refreshed as a consequence.

5. Anti-spoofing mechanisms should be applied to the Location Reporting method.

[Note: The term Valid Emergency Services Authority (VESA) refers to the root certificate authority. VPC stands for VoIP Positioning Center and ERDB stands for the Emergency Service Zone Routing Database.]

As noted above, signing of location objects implies the development of a trust hierarchy that would enable a certificate chain provided by the LIS operator to be verified by the PSAP. Rooting the trust hierarchy in VESA can be accomplished either by having the VESA directly sign the LIS certificates, or by the creation of intermediate Certificate Authorities (CAs) certified by the VESA, which will then issue certificates to the LIS. In terms of the workload imposed on the VESA, the latter approach is highly preferable. However, this raises the question of who would operate the intermediate CAs and what the expectations would be.

In particular, the question arises as to the requirements for LIS certificate issuance, and how they would compare to requirements for issuance of other certificates such as an SSL/TLS web certificate.
3.2. Location-by-Reference

Location-by-Reference was developed so that end hosts can avoid having to periodically query the location server for up-to-date location information in a mobile environment. Additionally, if operators do not want to disclose location information to the end host without charging them, location-by-reference provides a reasonable alternative. Also, since location-by-reference enables the PSAP to directly contact the location server, it avoids potential attacks by intermediaries.

As noted in "A Location Dereference Protocol Using HTTP-Enabled Location Delivery (HELD)" [RFC6753], a location reference can be obtained via HTTP-Enabled Location Delivery (HELD) [RFC5985]. In addition, "Location Configuration Extensions for Policy Management" [RFC7199] extends location configuration protocols such as HELD to provide hosts with a reference to the rules that apply to a Location-by-Reference so that the host can view or set these rules.

Figure 2 shows the communication model with the target requesting a location reference in step (a), the location server returns the reference and potentially the policy in step (b), and it is then conveyed to the location recipient in step (c). The location recipient needs to resolve the reference with a request in step (d). Finally, location information is returned to the Location Recipient afterwards. For location conveyance in SIP, the procedures described in [RFC6442] are applicable.

---

Figure 2: Location by Reference
Where location by reference is provided, the recipient needs to deference the LbyR in order to obtain location. The details for the dereferencing operations vary with the type of reference, such as a HTTP, HTTPS, SIP, SIPS URI or a SIP presence URI.

For location-by-reference, the location server needs to maintain one or several URIs for each target, timing out these URIs after a certain amount of time. References need to expire to prevent the recipient of such a Uniform Resource Locator (URL) from being able to permanently track a host and to offer garbage collection functionality for the location server.

Off-path adversaries must be prevented from obtaining the target’s location. The reference contains a randomized component that prevents third parties from guessing it. When the location recipient fetches up-to-date location information from the location server, it can also be assured that the location information is fresh and not replayed. However, this does not address location theft.

With respect to the security of the de-reference operation, [RFC6753] Section 6 states:

TLS MUST be used for dereferencing location URIs unless confidentiality and integrity are provided by some other mechanism, as discussed in Section 3. Location Recipients MUST authenticate the host identity using the domain name included in the location URI, using the procedure described in Section 3.1 of [RFC2818]. Local policy determines what a Location Recipient does if authentication fails or cannot be attempted.

The authorization by possession model (Section 4.1) further relies on TLS when transmitting the location URI to protect the secrecy of the URI. Possession of such a URI implies the same privacy considerations as possession of the PIDF-LO document that the URI references.

Location URIs MUST only be disclosed to authorized Location Recipients. The GEOPRIV architecture [RFC6280] designates the Rule Maker to authorize disclosure of the URI.

Protection of the location URI is necessary, since the policy attached to such a location URI permits anyone who has the URI to view the associated location information. This aspect of security is covered in more detail in the specification of location conveyance protocols, such as [RFC6442].

For authorizing access to location-by-reference, two authorization models were developed: "Authorization by Possession" and
"Authorization via Access Control Lists". With respect to "Authorization by Possession" [RFC6753] Section 4.1 notes:

In this model, possession -- or knowledge -- of the location URI is used to control access to location information. A location URI might be constructed such that it is hard to guess (see C8 of [RFC5808]), and the set of entities that it is disclosed to can be limited. The only authentication this would require by the LS is evidence of possession of the URI. The LS could immediately authorize any request that indicates this URI.

Authorization by possession does not require direct interaction with Rule Maker; it is assumed that the Rule Maker is able to exert control over the distribution of the location URI. Therefore, the LIS can operate with limited policy input from a Rule Maker.

Limited disclosure is an important aspect of this authorization model. The location URI is a secret; therefore, ensuring that adversaries are not able to acquire this information is paramount. Encryption, such as might be offered by TLS [RFC5246] or S/MIME [RFC5751], protects the information from eavesdroppers.

Using possession as a basis for authorization means that, once granted, authorization cannot be easily revoked. Cancellation of a location URI ensures that legitimate users are also affected; application of additional policy is theoretically possible but could be technically infeasible. Expiration of location URIs limits the usable time for a location URI, requiring that an attacker continue to learn new location URIs to retain access to current location information.

In situations where "Authorization by Possession" is not suitable (such as where location hiding [RFC6444] is required), the "Authorization via Access Control Lists" model may be preferred.

Without the introduction of hierarchy, it would be necessary for the PSAP to obtain credentials, such as certificates or shared symmetric keys, for all the LISes in its coverage area, to enable it to successfully dereference LbyRs. In situations with more than a few LISes per PSAP, this would present operational challenges.

A certificate hierarchy providing PSAPs with client certificates chaining to the VESA could be used to enable the LIS to authenticate and authorize PSAPs for dereferencing. Note that unlike PIDF-LO signing (which mitigates against modification of PIDF-LOs), this merely provides the PSAP with access to a (potentially unsigned) PIDF-LO, albeit over a protected TLS channel.
Another approach would be for the local LIS to upload location information to a location aggregation point who would in turn manage the relationships with the PSAP. This would shift the management burden from the PSAPs to the location aggregation points.

3.3. Proxy Adding Location

Instead of relying upon the end host to provide location, it is possible for a proxy that has the ability to determine the location of the end point (e.g., based on the end host IP or MAC address) to retrieve and add or override location information. This requires deployment of application layer entities by ISPs, unlike the two other techniques. The proxies could be used for emergency or non-emergency communications, or both.

The use of proxy-added location is primarily applicable in scenarios where the end host does not provide location. As noted in [RFC6442] Section 4.1:

A SIP intermediary SHOULD NOT add location to a SIP request that already contains location. This will quite often lead to confusion within LRIs. However, if a SIP intermediary adds location, even if location was not previously present in a SIP request, that SIP intermediary is fully responsible for addressing the concerns of any 424 (Bad Location Information) SIP response it receives about this location addition and MUST NOT pass on (upstream) the 424 response. A SIP intermediary that adds a locationValue MUST position the new locationValue as the last locationValue within the Geolocation header field of the SIP request.

A SIP intermediary MAY add a Geolocation header field if one is not present -- for example, when a user agent does not support the Geolocation mechanism but their outbound proxy does and knows the Target’s location, or any of a number of other use cases (see Section 3).

As noted in [RFC6442] Section 3.3:

This document takes a "you break it, you bought it" approach to dealing with second locations placed into a SIP request by an intermediary entity. That entity becomes completely responsible for all location within that SIP request (more on this in Section 4).

While it is possible for the proxy to override location included by the end host, [RFC6442] Section 3.4 notes the operational limitations:

Tschofenig, et. al  Informational  [Page 18]
Overriding location information provided by the user requires a deployment where an intermediary necessarily knows better than an end user -- after all, it could be that Alice has an on-board GPS, and the SIP intermediary only knows her nearest cell tower. Which is more accurate location information? Currently, there is no way to tell which entity is more accurate or which is wrong, for that matter. This document will not specify how to indicate which location is more accurate than another.

The disadvantage of this approach is the need to deploy application layer entities, such as SIP proxies, at IAPs or associated with IAPs. This requires a standardized VoIP profile to be deployed at every end device and at every IAP. This might impose interoperability challenges.

Additionally, the IAP needs to take responsibility for emergency calls, even for customers they have no direct or indirect relationship with. To provide identity information about the emergency caller from the VSP it would be necessary to let the IAP and the VSP to interact for authentication (see, for example, "Diameter Session Initiation Protocol (SIP) Application" [RFC4740]). This interaction along the Authentication, Authorization and Accounting infrastructure is often based on business relationships between the involved entities. An arbitrary IAP and VSP are unlikely to have a business relationship. In case the interaction between the IAP and the VSP fails due to the lack of a business relationship then typically a fall-back would be provided where no emergency caller identity information is made available to the PSAP and the emergency call still has to be completed.

4. Location Trust Assessment

The ability to assess the level of trustworthiness of conveyed location information is important, since this makes it possible to understand how much value should be placed on location information, as part of the decision making process. As an example, if automated location information is understood to be highly suspect or is absent, a call taker can put more effort into verifying the authenticity of the call and to obtaining location information from the caller.

Location trust assessment has value regardless of whether the location itself is authenticated (e.g. signed location) or is obtained directly from the location server (e.g. location-by-reference) over security transport, since these mechanisms do not provide assurance of the validity or provenance of location data.

To prevent location-theft attacks, the "entity" element of the PIDF-LO is of limited value if an unlinked pseudonym is provided in this...
field. However, if the LIS authenticates the target, then the linkage between the pseudonym and the target identity can be recovered in a post-incident investigation.

As noted in [I.D.thomson-geopriv-location-dependability], if the location object was signed, the location recipient has additional information on which to base their trust assessment, such as the validity of the signature, the identity of the target, the identity of the LIS, whether the LIS authenticated the target, and the identifier included in the "entity" field.

Caller accountability is also an important aspect of trust assessment. Can the individual purchasing the device or activating service be identified or did the call originate from a non-service initialized (NSI) device whose owner cannot be determined? Prior to the call, was the caller authenticated at the network or application layer? In the event of a hoax call, can audit logs be made available to an investigator, or can information relating to the owner of an unlinked pseudonym be provided, enabling investigators to unravel the chain of events that lead to the attack?

In practice, the source of the location data is important for location trust assessment. For example, location provided by a Location Information Server (LIS) whose administrator has an established history of meeting emergency location accuracy requirements (e.g. Phase II) may be considered more reliable than location information provided by a third party Location Service Provider (LSP) that disclaims use of location information for emergency purposes.

However, even where an LSP does not attempt to meet the accuracy requirements for emergency location, it still may be able to provide information useful in assessing about how reliable location information is likely to be. For example, was location determined based on the nearest cell tower or 802.11 Access Point (AP), or was a triangulation method used? If based on cell tower or AP location data, was the information obtained from an authoritative source (e.g. the tower or AP owner) and when was the last time that the location of the tower or access point was verified?

For real-time validation, information in the signaling and media packets can be cross checked against location information. For example, it may be possible to determine the city, state, country or continent associated with the IP address included within SIP Via: or Contact: headers, or the media source address, and compare this against the location information reported by the caller or conveyed in the PIDF-LO. However, in some situations only entities close to the caller may be able to verify the correctness of location information.
Real-time validation of the timestamp contained within PIDF-LO objects (reflecting the time at which the location was determined) is also challenging. To address time-shifting attacks, the "timestamp" element of the PIDF-LO, defined in [RFC3863], can be examined and compared against timestamps included within the enclosing SIP message, to determine whether the location data is sufficiently fresh. However, the timestamp only represents an assertion by the LIS, which may or may not be trustworthy. For example, the recipient of the signed PIDF-LO may not know whether the LIS supports time synchronization, or whether it is possible to reset the LIS clock manually without detection. Even if the timestamp was valid at the time location was determined, a time period may elapse between when the PIDF-LO was provided and when it is conveyed to the recipient. Periodically refreshing location information to renew the timestamp even though the location information itself is unchanged puts additional load on LISes. As a result, recipients need to validate the timestamp in order to determine whether it is credible.

While this document focuses on the discussion of real-time determination of suspicious emergency calls, the use of audit logs may help in enforcing accountability among emergency callers. For example, in the event of a hoax call, information relating to the owner of the unlinked pseudonym could be provided to investigators, enabling them to unravel the chain of events that lead to the attack. However, while auditability is an important deterrent, it is likely to be of most benefit in situations where attacks on the emergency services system are likely to be relatively infrequent, since the resources required to pursue an investigation are likely to be considerable. However, although real-time validation based on PIDF-LO elements is challenging, where LIS audit logs are available (such as where a law enforcement agency can present a subpoena), linking of a pseudonym to the device obtaining location can be accomplished during an investigation.

Where attacks are frequent and continuous, automated mechanisms are required. For example, it might be valuable to develop mechanisms to exchange audit trails information in a standardized format between ISPs and PSAPs / VSPs or heuristics to distinguish potentially fraudulent emergency calls from real emergencies. While a Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) may be applied to suspicious calls to lower the risk from bot-nets, this is quite controversial for emergency services, due to the risk of delaying or rejecting valid calls.
5. Security Considerations

Although it is important to ensure that location information cannot be faked, the mitigation techniques presented in this document are not universally applicable. For example, there will be many GPS-enabled devices that will find it difficult to utilize any of the solutions described in Section 3. It is also unlikely that users will be willing to upload their location information for "verification" to a nearby location server located in the access network.

This document focuses on threats that arise from conveyance of misleading location information, rather than caller identification or authentication and integrity protection of the messages in which location is conveyed. Nevertheless, these aspects are important. In some countries, regulators may not require the authenticated identity of the emergency caller (e.g., emergency calls placed from PSTN pay phones or SIM-less cell phones). Furthermore, if identities can easily be crafted (as it is the case with many VoIP offerings today), then the value of emergency caller authentication itself might be limited. As a result, attackers can forge emergency calls with a lower risk of being held accountable, which may encourage hoax calls.

In order to provide authentication and integrity protection for the Session Initiation Protocol (SIP) messages conveying location, several security approaches are available. It is possible to ensure that modification of the identity and location in transit can be detected by the location recipient (e.g., the PSAP), using cryptographic mechanisms, as described in "Enhancements for Authenticated Identity Management in the Session Initiation Protocol" [RFC4474]. However, compatibility with Session Border Controllers (SBCs) that modify integrity-protected headers has proven to be an issue in practice, and as a result, a revision is in progress [I.D.ietf-stir-rfc4474bis]. In the absence of an end-to-end solution, SIP over Transport Layer Security (TLS) can be used to provide message authentication and integrity protection hop-by-hop.

PSAPs remain vulnerable to distributed denial of service attacks, even where the mitigation techniques described in this document are utilized. Placing a large number of emergency calls that appear to come from different locations is an example of an attack that is difficult to carry out within the legacy system, but is easier to imagine within IP-based emergency services. Also, in the current system, it would be very difficult for an attacker from country 'Foo' to attack the emergency services infrastructure located in country 'Bar', but this attack is possible within IP-based emergency services.
While manually mounting the attacks described in Section 2 is non-trivial, the attacks described in this document can be automated. While manually carrying out a location theft would require the attacker to be in proximity to the location being spoofed, or to collude with another end host, an attacker able to run code on an end host can obtain its location, and cause an emergency call to be made. While manually carrying out a time shifting attack would require that the attacker visit the location and submit it before the location information is considered stale, while traveling rapidly away from that location to avoid apprehension, these limitations would not apply to an attacker able to run code on the end host. While obtaining a PIDF-LO from a spoofed IP address requires that the attacker be on the path between the HELD requester and the LIS, if the attacker is able to run code requesting the PIDF-LO, retrieve it from the LIS, and then make an emergency call using it, this attack becomes much easier. To mitigate the risk of automated attacks, service providers can limit the ability of untrusted code (such as WebRTC applications written in Javascript) to make emergency calls.

Emergency services have three finite resources subject to denial of service attacks: the network and server infrastructure, call takers and dispatchers, and the first responders, such as fire fighters and police officers. Protecting the network infrastructure is similar to protecting other high-value service providers, except that location information may be used to filter call setup requests, to weed out requests that are out of area. Even for large cities PSAPs may only have a handful of call takers on duty. So even if automated techniques are utilized to evaluate the trustworthiness of conveyed location and call takers can, by questioning the caller, eliminate many hoax calls, PSAPs can be overwhelmed even by a small-scale attack. Finally, first responder resources are scarce, particularly during mass-casualty events.

6. Privacy Considerations

The emergency calling architecture described in [RFC6443] utilizes the PIDF-LO format defined in [RFC4119]. As described in the location privacy architecture [RFC6280], privacy rules that may include policy instructions are conveyed along with the location object.

The intent of the location privacy architecture was to provide strong privacy protections, as noted in [RFC6280] Section 1.1:

A central feature of the Geopriv architecture is that location information is always bound to privacy rules to ensure that entities that receive location information are informed of how they may use it. These rules can convey simple directives ("do
not share my location with others"), or more robust preferences
("allow my spouse to know my exact location all of the time, but
only allow my boss to know it during work hours")... The binding
of privacy rules to location information can convey users’ desire
for and expectations of privacy, which in turn helps to bolster
social and legal systems’ protection of those expectations.

However, in practice this architecture has limitations which apply
within emergency and non-emergency situations. As noted in Section
1.2.2, concerns about hoax calls have lead to restrictions on
anonymous emergency calls. Caller identification (potentially
asserted in SIP via P-Asserted-Identity and via SIP Identity) may be
used during emergency calls. As a result, in many cases location
information transmitted within SIP messages can be linked to caller
identity. For example, in case of signed LbyV, there are privacy
concerns arising from linking the location object to identifiers to
prevent replay attacks, as described in Section 3.1.

The ability to observe location information during emergency calls
may also represent a privacy risk. As a result, [RFC6443] requires
transmission layer security for SIP messages, as well as interactions
with the location server. However, even where transmission layer
security is used, privacy rules associated with location information
may not apply.

In many jurisdictions, an individual requesting emergency assistance
is assumed to be granting permission to the PSAP, call taker and
first responders to obtain their location in order to accelerate
dispatch. As a result, privacy policies associated with location are
implicitly waived when an emergency call is initiated. In addition,
when location information is included within SIP messages either in
emergency or non-emergency uses, SIP entities receiving the SIP
message are implicitly assumed to be authorized location recipients,
as noted in [RFC5606] Section 3.2:

Consensus has emerged that any SIP entity that receives a SIP
message containing LI through the operation of SIP’s normal
routing procedures or as a result of location-based routing should
be considered an authorized recipient of that LI. Because of this
presumption, one SIP element may pass the LI to another even if
the LO it contains has <retransmission-allowed> set to "no"; this
sees the passing of the SIP message as part of the delivery to
authorized recipients, rather than as retransmission. SIP
entities are still enjoined from passing these messages outside
the normal routing to external entities if <retransmission-
allowed> is set to "no", as it is the passing to third parties
that <retransmission-allowed> is meant to control.
Where LbyR is utilized rather than LbyV, it is possible to apply more restrictive authorization policies, limiting access to intermediaries and snoopers. However, this is not possible if the "authorization by possession" model is used.

7. IANA Considerations

This document does not require actions by IANA.

8. References

8.1. Informative References

[I-D.ietf-stir-problem-statement]

[I-D.ietf-stir-threats]

[I-D.ietf-stir-rfc4474bis]

[I-D.thomson-geopriv-location-dependability]


[GPSCounter]

INTERNET-DRAFT            Trustworthy Location              28 July 2014


Tschofenig, et. al               Informational          [Page 26]


Acknowledgments

We would like to thank the members of the IETF ECRIT working group, including Marc Linsner and Brian Rosen, for their input at IETF 85 that helped get this documented pointed in the right direction. We would also like to thank members of the IETF GEOPRIV WG, including Andrew Newton, Murugaraj Shanmugam, Martin Thomson, Richard Barnes and Matt Lepinski for their feedback to previous versions of this document. Thanks also to Pete Resnick, Adrian Farrel, Alissa Cooper, Bert Wijnen and Meral Shirazipour who provided review comments in IETF last call.
Authors' Addresses

Hannes Tschofenig
Austria

Email: Hannes.tschofenig@gmx.net
URI: http://www.tschofenig.priv.at

Henning Schulzrinne
Columbia University
Department of Computer Science
450 Computer Science Building, New York, NY 10027
US

Phone: +1 212 939 7004
Email: hgs@cs.columbia.edu
URI: http://www.cs.columbia.edu

Bernard Aboba
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
US

Email: bernard_aboba@hotmail.com
Uniform Resource Name (URN) extension for automatic and manual Emergency Services
draft-jesske-ecrit-ecall-urn-extension-01

Abstract

This document describes and discusses a mechanism for extending sos URN’s to fulfill the emergency call requirements of carriers.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 16, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
Uniform Resource Name (URN) for emergency and other Well-Known Services are well defined within RFC5031 [2]. There are also existing proposals for automatic and manual ecall IN-Vehicles which defines their own URN’s and additional procedures for vehicle specific eCalls. Due to the existing telecommunication world and their implementations a combination of both is needed. Due to the fact that only one URN can be used within the request line of a SIP Message a new approach is needed. Therefore this document proposes to extend the existing URN’s with an extension to reflect this.

This draft should be seen as proposal which could be incorporated into draft-rosen-ecrit-ecall-10 [3] after discussing this approach.

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [1].

3. Overview


Within the existing implementation in GSM networks for emergency call as specified in 3GPP TS 22.101 [4] the following requirements and code points are described:

---
It shall be possible to initiate emergency calls to different emergency call centers, depending on the type of emergency. The following types of emergency calls shall be possible:

- Police
- Ambulance
- Fire Brigade
- Marine Guard
- Mountain Rescue
- Manually Initiated eCall
- Automatically Initiated eCall

....

It shall be possible to tie any emergency call number to any single emergency call type or to any combination of emergency call types.

This means in practice that a emergency Call Center from police must differentiate between automatic and manual eCalls. Same belongs to the other types like fire brigade Marine Guard ect.

Requirement:

- Req-1: It shall be possible to tie a manually initiated eCall with different type of emergency type i.e. Police, Ambulance, Fire Brigade, Marine Guard and Mountain Rescue.
- Req-2: It shall be possible to tie a automatically initiated eCall with different type of emergency type i.e. Police, Ambulance, Fire Brigade, Marine Guard and Mountain Rescue.

These requirements will not allow a second different URN Format as described in draft-rosen-ecrit-ecall-10 [3]. This draft adds two additional URN urn:service:sos.ecall.manual and urn:service:sos.ecall.automatic Which allows to address the ecall itself. But the requirement in 3GPP TS 22.101 [4] described will not be satisfied. Therefore this document proposes an other format of the URN.
4. IANA Consideration

IANA is requested to register the URNs described below under the sub-services ‘sos’ registry defined in Section 4.2 of RFC5031[2]

4.1. sublabel for sos service and the related subservices

An additional sublabel for sos and the 5 base values of police, ambulance, fire, mountain are defined.

.ecall

This sublevel indicated that an ecall has been triggered.

.automatic

This sublevel indicates that an eCall had been triggered automatically. This could be a sensor-controlled measuring of heat or pressure in case of car accidents or fire situations.

.manual

This sublevel indicates that an eCall had been triggered based on the manual interaction of a person.

<table>
<thead>
<tr>
<th>Service</th>
<th>Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sos.ecall</td>
<td>RFC xxxx</td>
<td>Emergency services, automatic eCall</td>
</tr>
<tr>
<td>sos.ecall.automatic</td>
<td>RFC xxxx</td>
<td>Emergency services, automatic eCall</td>
</tr>
<tr>
<td>sos.ecall.manual</td>
<td>RFC xxxx</td>
<td>Emergency services, manual eCall</td>
</tr>
<tr>
<td>sos.police.ecall.automatic</td>
<td>RFC xxxx</td>
<td>Police, law enforcement, automatic eCall</td>
</tr>
<tr>
<td>sos.police.ecall.manual</td>
<td>RFC xxxx</td>
<td>Police, law enforcement, manual eCall</td>
</tr>
<tr>
<td>sos.ambulance.ecall.automatic</td>
<td>RFC xxxx</td>
<td>Ambulance service, automatic eCall</td>
</tr>
<tr>
<td>sos.ambulance.ecall.manual</td>
<td>RFC xxxx</td>
<td>Ambulance service, manual eCall</td>
</tr>
<tr>
<td>sos.fire.ecall.automatic</td>
<td>RFC xxxx</td>
<td>Fire service, automatic eCall</td>
</tr>
<tr>
<td>sos.fire.ecall.manual</td>
<td>RFC xxxx</td>
<td>Fire service, manual eCall</td>
</tr>
<tr>
<td>sos.marine.ecall.automatic</td>
<td>RFC xxxx</td>
<td>Maritime search and rescue, automatic eCall</td>
</tr>
<tr>
<td>sos.marine.ecall.manual</td>
<td>RFC xxxx</td>
<td>Maritime search and rescue, manual eCall</td>
</tr>
<tr>
<td>sos.mountain.ecall.automatic</td>
<td>RFC xxxx</td>
<td>Mountain rescue, automatic eCall</td>
</tr>
<tr>
<td>sos.mountain.ecall.manual</td>
<td>RFC xxxx</td>
<td>Mountain rescue, manual eCall</td>
</tr>
</tbody>
</table>
5. Security Considerations

This document does not raise security considerations beyond those described in RFC5031 [2].

6. Contributors and Acknowledgements

The author would like to thank Dieter Jacobsohn, Maik Kirsch and Thomas Dennert for clarifying the requirements on automatic and manual eCall in the existing GSM world.

7. Normative References


Author’s Address

Roland Jesske
Deutsche Telekom
Heinrich-Hertz-Strasse 3-7
Darmstadt 64307
Germany

Phone: +4961515812766
Email: r.jesske@telekom.de
Abstract

This document introduces a new way to provide returned location information in LoST responses that is either of a completed or similar form to the original input civic location, based on whether valid or invalid civic address elements are returned within the findServiceResponse message. This document defines a new extension to the findServiceResponse message within the LoST protocol [RFC5222] that enables the LoST protocol to return a completed civic address element set for a valid location response, and one or more suggested sets of similar location information for invalid LoST responses. These two types of civic addresses are referred to as either "complete location" or "similar location", and are included as compilation of ca type xml elements within the existing LoST findServiceResponse message structure.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 24, 2015.
1. Introduction

The LoST protocol [RFC5222] supports the validation of civic location information as input, by providing a set of validation result status indicators. The current usefulness of the supported xml elements, "valid", "invalid", and "unchecked", is limited, because while they each provide an indication of validity for any one location element as a part of the whole civic address, the mechanism is insufficient in providing either the complete set of civic address elements that the LoST server contains, or of providing alternate suggestions (hints) as to which civic address is intended for use.
Whether the input civic location is valid and missing information, or invalid due to missing or wrong information during input, this document provides a mechanism to return a complete set of civic address elements for those valid or invalid cases.

This enhancement to the validation feature within LoST is required by systems that rely on accurate location for processing in order to increase the likelihood that the correct and/or complete form of a civic location becomes known in those cases where it is incomplete or just plain wrong. One such use case is that of location based emergency calling. The use of this protocol extension will reduce user and system input errors, and will result in a higher level of civic address matching, reducing the number of mismatch errors, where a civic address that appears to be valid gets wrongly associated with the physical location of the caller.

The structure of this document includes terminology, Section 2, followed by a discussion of the basic elements involved in location validation. The use of these elements, by way of example, is discussed in an overview section, Section 3, with accompanying rationale, and a brief discussion of the impacts to LoST, and its current schema.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119], with the important qualification that, unless otherwise stated, these terms apply to the design of the Location Configuration Protocol and the Location Dereferencing Protocol, not its implementation or application.

The following terms are defined in this document:

Location: The term Location can be used to refer to either a civic location or a geodetic location.

Geodetic Location: a geographic coordinate set of values that describes a point within a defined geographic datum. For example, a WGS84 referenced latitude, longitude coordinate pair (2D), or latitude, longitude, and altitude (3D). Note: geodetic location is defined here for context, but is not used elsewhere within this document.

Civic Location: The term civic location applies to a set of one or more civic address elements that are used in conjunction with each other, and in accordance with a known ruleset to designates a
place within a defined grid or basemap. The example used within this document is a street address as defined in [RFC5139]

Civic Address: The term Civic Address is used interchangeably with the term Civic Location within this document.

Street Address: The term Street Address is used to represent a place, or location on a defined grid or map. While generally equated to both terms, Civic Location and Civic Address, it is not used within this document.

Civic Address Element: The term Civic Address Element is used within this document to apply to an individual CAtype data descriptor, for example, as is defined in [RFC4776]

Invalid: The status result of the unsuccessful attempt to match an individual input data as part of a larger set of data that has already been successfully matched and as shown by the [RFC5222] defined xml named element

Valid: The status result of the successful attempt to match an individual input data as part of a larger set of data that has already also been successfully matched and shown by the [RFC5222] defined xml named element

Invalid Location: A Civic Location that was included in a LoST request and subsequently returned with one or more civic address elements marked as invalid.

Valid Location: A Civic Location that was included in a LoST request and subsequently returned with all civic address elements marked as valid.

Complete Location: An expanded civic location that includes other civic address elements in addition to the existing validated civic address elements provided as input to a LoST server.

Similar Location: A suggested civic location that is comparatively close to the civic location which was input, but which had one or more invalid civic address elements returned by the LoST server.

Returned Location Information: A set of standard civic address elements returned in a LoST response.
3. Overview of Returned Location Information

This document describes an extension to LoST [RFC5222] to allow additional location information to be returned in a findServiceResponse for two different use cases.

When a LoST server is asked to validate a civic location, its goal is to take the set of civic address elements provided as the location information in the LoST request, and find a unique location in its database that matches the information in the request. Uniqueness might not require values for all possible elements in the civic address that the database might hold. Further, the input location information might not represent the form of location the users of the LoST service prefer to have. As an example, there are LoST civic address elements that could be used to define a postal location, suitable for delivery mail as well as a municipal location suitable for responding to an emergency call. While the LoST server might be able to determine the location from the postal elements provided, the emergency services would prefer that the municipal location be used for any subsequent emergency call. Since validation is often performed well in advance of an end-user placing an emergency call, if the LoST server could return the preferred form of location (or more properly, the municipal elements in addition to the postal elements), those elements could be stored in a LIS and used in a later emergency call.

Since a LoST server often contains more data than what is included within a findService request, it is expected that this additional location information, if present, SHOULD be returned within response messages that contain valid civic address elements. For valid location responses, where a LoST server contains additional location information relating to that civic address, the findServiceResponse message MAY return additional location information along with the original validated civic address elements in order to form a complete location based on local implementation policy.

In addition, this document describes the reuse of the same mechanism, but for a different purpose: to supply similar location information in the case where a LoST server response includes one or more civic address elements marked as invalid, constituting an invalid location response, offering one or more suggested alternative address that would consist of one or more valid locations.

LoST servers that implement this extension have no way to alert clients that may not be aware of the extension’s capabilities, other than supplying the extended data set. It is expected that a LoST client implementation that is not aware of this extension for complete and/or similar location SHOULD be able to still receive the
findServiceResponse data, while throwing away any extra complete or similar location data.

In a valid location response, a LoST server returns a response to a findService request that contains a set of civic address elements marked valid, the location information in the findServiceResponse message MAY be extended to include additional location information specific for that location. As an example, the query might contain a HNO (house number), RD (road name) and A3 (city) and a few more caType elements, but might not contain A1 (state), PC (Postal Code) CAtypes. The HNO, RD, STS, POD, and A3 civic address elements might be sufficient enough to the LoST server to uniquely locate the address specified in the request and thus be considered valid. Yet, downstream entities might find it helpful to have the additional country, A1 (state), and PC, (Postal Code), civic address elements that are present within the LoST server, be included as part of a complete location response. Since [RFC5222] currently does not have a way for this additional location information to be returned in the findServiceResponse, this document extends the LoST protocol so that it can include a completeLocation element within the findServiceResponse message, allowing for the representation of complete location information.

An example showing complete location information supplied:

input address: 6000 15th Ave NW Seattle

complete location: 6000 15th Ave NW Seattle, WA 98105 US

By constrast, when invalid location is received from the LoST server, with this extension, the same mechanism works as follows: if a LoST server returns a response to a findService request that contains a set of civic address elements with one or more labeled as invalid, the location information in the findServiceResponse is extended to include additional location information that it knows is specific for that location. Differing results based on somewhat close input data as used above, where the HNO, RD, STS, A1, and A3 civic address elements are not sufficient to locate a unique address leads to an invalid location result. This is the case, despite the fact that the LoST server typically contains additional civic address elements which could have resulted in a uniquely identifiable location if additional data had been supplied with the query. Since [RFC5222] currently does not have a way for this additional location information to be returned in the findServiceResponse, this document extends [RFC5222] so that the LoST findServiceResponse message can include one or more similarLocation elements within the findServiceResponse message representing similar civic locations.
To show this, suppose that a slightly modified address as above is inserted within a Lost findService request:

input address: 6000 15th Ave N Seattle, WA.

Different from the previous use case, this time we make the assumption that the address is deemed "invalid" by the LoST server because there is no such thing as "15th Ave N" within the LoST server's data for the city of Seattle. However, we also happen to know for this example that there are two addresses within the address dataset that are "similar", when all parts of the address are taken as a whole. These similar addresses that could be suggested to the user are as follows:

similar address #1: 6000 15th Ave NW Seattle, WA 98107

similar address #2: 6000 15th Ave NE Seattle, WA 98105

This document proposes to include the above similar addresses as civicAddress elements in the response to locationValidation. The next section shows examples of the LoST request and response xml message fragments for the above valid and invalid scenarios, returning the complete or similar addresses, respectively:

4. Returned Location Information

The LoST server knows the data that is available internally, and can determine which additional civic address elements can be provided either as part of a complete location or a similar location. The inclusion of either complete location or similar location is not triggered by any message parameter, but is triggered based on whether the returned location information is valid or invalid. It is not turned on or off, but is implementation specific.

5. Complete Location returned for Valid Location response

Based on the example input request, returned location information is provided in a findServiceResponse message when the original input address is considered valid, but is missing some additional data that the LoST server has.

<!-- =====Request=================================== -->

<findService xmlns="urn:ietf:params:xml:ns:lost1"
  validateLocation="true">

<location id="587cd3880" profile="civic">
  <civicAddress xmlns="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr">
    <A1>WA</A1>
    <A3>Seattle</A3>
    <RD>15th</RD>
    <STS>Ave</STS>
    <POD>NW</POD>
    <HNO>6000</HNO>
  </civicAddress>
</location>

<service>urn:service:sos</service>

<!-- =====Response================================== -->

<findServiceResponse >
  xmlns="urn:ietf:params:xml:ns:lost1"
  xmlns:rli="urn:ietf:params:xml:ns:lost-rli1"

  <mapping expires="NO-CACHE"
    lastUpdated="2006-11-01T01:00:00Z"
    source="authoritative.example"
    sourceId="8799e346000098aa3e">
    <displayName xml:lang="en">Seattle 911</displayName>
    <service>urn:service:sos</service>
    <uri>sip:seattle-911@example.com</uri>
    <serviceNumber>911</serviceNumber>
  </mapping>

<locationValidation>
  <valid>ca:A3 ca:RD ca:STS ca:POD ca:HNO</valid>
  <invalid></invalid>
  <unchecked></unchecked>
  <rli:completeLocation>  <!-- completed address -->
    <ca:civicAddress>
<findServiceResponse>

<!-- =============================================== -->

6. Similar Location returned for Invalid Location response

The following example shows returned location information provided in a findServiceResponse message when the original input address is considered invalid, because of the unmatchable POD data (in this example) that the LoST server needs to provide a unique mapping.

<!-- =====Request=================================== -->

<findService xmlns="urn:ietf:params:xml:ns:lost1" validateLocation="true">

<location id="587cd3880" profile="civic">
<civicAddress xmlns="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr">

<country>US</country>
</civicAddress>

</location>

</findService>
<civicAddress>
  <A1>WA</A1>
  <A3>Seattle</A3>
  <RD>15th</RD>
  <STS>Ave</STS>
  <POD>N</POD>
  <HNO>6000</HNO>
</civicAddress>
</location>

<service>urn:service:sos</service>

</findService>

<!-- =====Response=================================== -->

<findServiceResponse>
  <mapping
    expires="NO-CACHE"
    lastUpdated="2006-11-01T01:00:00Z"
    source="authoritative.example"
    sourceId="8799e346000098aa3e">
    <displayName xml:lang="en">Seattle 911</displayName>
    <service>urn:service:sos</service>
    <uri>sip:seattle-911@example.com</uri>
    <serviceNumber>911</serviceNumber>
  </mapping>

  <locationValidation
    <invalid>ca:POD</invalid>
    <unchecked>ca:HNO</unchecked>
    <rli:similarLocation>  <!-- similar location info -->
      <ca:civicAddress>  <!-- similar address #1 -->
        <ca:country>US</ca:country>
        <ca:A1>WA</ca:A1>
        <ca:A3>SEATTLE</ca:A3>
        <ca:RD>15TH</ca:RD>
        <ca:STS>AVE</ca:STS>
    </ca:civicAddress>
    <ca:civicAddress>  <!-- similar address #2 -->
      <ca:country>US</ca:country>
      <ca:A1>WA</ca:A1>
      <ca:A3>SEATTLE</ca:A3>
      <ca:RD>15TH</ca:RD>
      <ca:STS>AVE</ca:STS>
    </ca:civicAddress>
  </rli:similarLocation>
</findServiceResponse>

<ca:civicAddress> <!-- similar address #2 -->
  <ca:country>US</ca:country>
  <ca:A1>WA</ca:A1>
  <ca:A3>SEATTLE</ca:A3>
  <ca:RD>15TH</ca:RD>
  <ca:STS>AVE</ca:STS>
  <ca:POD>NE</ca:POD>
  <ca:HNO>6000</ca:HNO>
  <ca:PC>98105</ca:PC>
</ca:civicAddress>
</rli:similarLocation>
</locationValidation>

<path>
  <via source="authoritative.example"/>
</path>

<locationUsed id="587cd3880"/>
</findServiceResponse>

<!-- =========================================================================

7. Relax NG schema

This section provides the Relax NG schema of LoST extensions in the compact form. The verbose form is included in a later section [to be supplied in a later version of this draft].

namespace a = "http://relaxng.org/ns/compatibility/annotations/1.0"
default namespace ns1 = "urn:ietf:params:xml:ns:lost-rli1"

##
## Extension to LoST to support returned location information
##
start =

returnedLocation

div {
    returnedLocationResponse =
        element returnedLocationResponse {
            completeLocation, similarLocation, extensionPoint
        }
    }

## completeLocation

div {
    completeLocation =
        element location {
            attribute id { xsd:token },
            locationInformation
        }+
    }

## similarLocation

div {
    similarLocation =
        element location {
            attribute id { xsd:token },
            locationInformation
        }+
    }

## Location Information

div {
    locationInformation =
        extensionPoint+,
        attribute profile { xsd:NMTOKEN }?
    }

Patterns for inclusion of elements from schemas in other namespaces.

Any element not in the LoST namespace.
notLost = element * - (ns1:* | ns1:*) { anyElement }

##
## A wildcard pattern for including any element from any other namespace.
##
anyElement =
    {element * { anyElement }
     | attribute * { text }
     | text}*

##
## A point where future extensions (elements from other namespaces) can be added.
##
extensionPoint = notRLI* 

8. Security Considerations

Whether the input to the LoST server is valid or invalid, the LoST server ultimately determines what it considers to be valid. Even in the case where the input location is valid, the requester still might not actually understand where that location is. For this kind of valid location use case, this described extension would typically return more location information than the requester started with, which might reveal more about the location. While this might be very desirable in some scenarios including, for example, supporting an emergency call, it might not be as desirable for other services. Individual LoST server implementations SHOULD consider the risk of releasing more detail verses the value in doing so. Generally, it is not expected that this would be a significant problem as the requester must have enough location information to be considered valid, which in most cases is enough to uniquely locate the address. Providing more CAtypes generally doesn’t actually reveal anything more. For invalid locations that are submitted, this extension would allow the LoST response to include location information which is similar to what was input, again resulting in more information provided in the response than was known during input. LoST server implementations SHOULD evaluate the particular use cases where this extension is supported, and weigh the risks around its use. Many similar database services available today via the Internet offer similar features, such as "did you mean", and address completion, so this capability is not introducing any fundamentally new threat.
9. IANA Considerations

9.1. Relax NG Schema Registration

 URI: urn:ietf:params:xml:schema:lost-rli1

Registrant Contact: IETF ECRIT Working Group, Brian Rosen
(br@brianrosen.net).

Relax NG Schema: The Relax NG schema to be registered is contained
in Section 7. Its first line is

default namespace = "urn:ietf:params:xml:ns:lost-rli1

and its last line is

}

9.2. LoST Namespace Registration

 URI: urn:ietf:params:xml:ns:lost-rli1

Registrant Contact: IETF ECRIT Working Group, Brian Rosen
(br@brianrosen.net).

XML:

BEGIN
<?xml version="2.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
"http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="content-type"
 content="text/html; charset=iso-8859-1"/>
<title>LoST Planned Change Namespace</title>
</head>
<body>
<h1>Namespace for LoST Returned Location Information extension</h1>
<h2>urn:ietf:params:xml:ns:lost-rli1</h2>
<p>See <a href="http://www.rfc-editor.org/rfc/rfc????.txt">
RFC????</a>.</p>
</body>
</html>
END
10. Acknowledgements

11. Changes from Previous Versions

11.1. Changes from draft-marshall-03 to -04

- Revised the text in Section 1 to better describe how this extension can be useful (Bradner)
- Utilized RFC2119 language in the draft rather than removing the reference to it (Bradner)
- Added some text to explain how notification of this extension is expected for those clients that are not aware of this extension could be notified (Bradner)
- Modified security section text to include security considerations for both valid and invalid addresses used as input. (Stark)
- Acknowledged: need for extension point RNG/detailed xml and examples (Stark)
- Reworked terminology section and aligned with text based on comments (Stark)
- General editorial cleanup

12. References

12.1. Normative References


12.2. Informative References

Authors’ Addresses

Roger Marshall
TeleCommunication Systems, Inc.
2401 Elliott Avenue
2nd Floor
Seattle, WA  98121
US
Phone: +1 206 792 2424
Email: rmarshall@telecomsys.com
URI:   http://www.telecomsys.com

Jeff Martin
TeleCommunication Systems, Inc.
2401 Elliott Avenue
2nd Floor
Seattle, WA  98121
US
Phone: +1 206 792 2584
Email: jmartin@telecomsys.com
URI:   http://www.telecomsys.com

Brian Rosen
Neustar
470 Conrad Dr
Mars, PA  16046
US
Email: br@brianrosen.net
Service Coverage Scope for Service URN
draft-mongrain-ecrit-service-coverage-scope-urn-00.txt

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This Internet-Draft will expire on September 25, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
1. Introduction

For some given location, there may be multiple providers that supply the same given service. These providers offer their services for different coverage areas. For example, in the United States of America, both the city police and the state police handle emergency calls. If one needs to reach the state police to request emergency assistance, a different Service URN is needed in order to obtain the appropriate contact URI when querying the LoST service [RFC5222]. This is accomplished by appending a Service Coverage Scope at the end of the Service URN.

The presence of a Service Coverage Scope in a Service URN indicates that the requester wants the service provider that provides the given service over the given coverage area for the given location. For example "urn:service:sos.police.country" specifies the national/federal police, "urn:service:sos.police.A1" the state/provincial/region/prefecture police, "urn:service:sos.police.A1" the county/parish/gun/district sheriff/police and "urn:service:sos.police.A3" the city/township/shi police.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].
3. Service Coverage Scope

A Service Coverage Scope SHALL be one of the following:

<table>
<thead>
<tr>
<th>Service Coverage Scope</th>
<th>Coverage Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>country</td>
<td>Entire country</td>
</tr>
<tr>
<td>A1</td>
<td>national subdivision (state, region, province, prefecture)</td>
</tr>
<tr>
<td>A2</td>
<td>county, parish, gun (JP), district (IN)</td>
</tr>
<tr>
<td>A3</td>
<td>city, township, shi (JP)</td>
</tr>
<tr>
<td>A4</td>
<td>city division, borough, city district, ward, chou (JP)</td>
</tr>
<tr>
<td>A5</td>
<td>neighborhood, block</td>
</tr>
</tbody>
</table>

The astute reader will immediately notice that the Service Coverage Scope names are identical to elements found in the civic location format [RFC4119]. This is to avoid utilizing political division nomenclature that is not universally applicable. For example "urn:service:sos.police.A1" specifies provincial police in Canada, prefecture police in Japan and state police in the United States of America.

A Service Coverage Scope MAY be used with any valid Service URN when a coverage area needs to be specified.

A Service Coverage Scope SHALL be the last element of a Service URN.
There SHALL NOT be more than one Service Coverage Scope specified in a Service URN.

When a Service Coverage Scope is specified, the service requested SHALL be provided over the entire coverage area. For example, the fictitious Whoville Fire Department cannot be contacted using "urn:service:sos.fire.A2" as it does not provide fire prevention services over the entire county/parish/gun/district the city of Whoville belongs to.

Nothing in this document prevents a service provider from offering their service over multiple coverage areas. For example, three neighbouring counties may decide to utilize the same sheriff department for all three counties. The Service URN "urn:service:sos.police.A2" can be used in all three counties to reach the sheriff department. The sheriff department cannot be reached using "urn:service:sos.police.A1" since it does not cover the entire state/province/region/prefecture.

The use of a Service Coverage Scope SHALL NOT change the intent of the Service URN it is appended to. For example, if a police department exists at the country level but it does not accept emergency calls, "urn:service:sos.police.country" cannot not be used to reach it as "urn:service:sos" and its subservices are intended to be used only when requesting immediate assistance [RFC5031].

4. Impact of Service Coverage Scopes on LoST Service Implementations

An implementation of a LoST service does not need to be modified in order to utilize Service Coverage Scopes. Service URNs containing a Service Coverage Scope are provisioned exactly the same way as any other Service URNs. Care must be given to ensure that service providers provisioned using a Service URN that includes a Service Coverage Scope do indeed provide the service over the entire designated coverage area.

This document introduces a new element that is appended to a Service URN. In the future, other subservices will be introduced that will also be added to Service URNs. It can be likely that a LoST Service is queried with a Service URN that is not provisioned, especially if the requester is a device which originates from a location where such a service exists. It may be desirable that the LoST Service returns a "next best thing" answer instead of serviceNotImplemented error. As specified in RFC-5222 section 5.4 [RFC5222], a LoST Service implementation can substitute another service in the case...
the given service is not defined. A method to accomplish this is to unroll the given Service URN. When a request to the LoST Service evaluates to what would be a serviceNotImplemented error, the LoST Service SHOULD remove the last element of the provided Service URN and re-evaluate the request. If the re-evaluation still results in what would be serviceNotImplemented, it SHOULD repeat the process again until all elements of the Service URN have been removed, in which case the LoST Service SHALL return a serviceNotImplemented error back to the requester.

5. Security Considerations

<Blatantly copied from RFC-5031> As an identifier, the Service URN does not appear to raise any particular security issues. The services described by the URN are meant to be well-known, even if the particular service instance is access-controlled, so privacy considerations do not apply to the URN.

There are likely no specific privacy issues when including a Service URN on a web page, for example. On the other hand, ferrying the URN in a signaling protocol can give attackers information on the kind of service desired by the caller. For example, this makes it easier for the attacker to automatically find all calls for emergency services or directory assistance. Appropriate, protocol-specific security mechanisms need to be implemented for protocols carrying Service URNs. The mapping protocol needs to address a number of threats, as detailed in [RFC5069]. That document also discusses the security considerations related to the use of the Service URN for emergency services.

6. IANA Considerations

<TBD>

7. References

7.1. Normative References

8. Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.

Authors’ Addresses

Daniel Mongrain
Frequentis
Suite 1002
150 Metcalfe Street
Ottawa, Ontario Canada
K2P 1P1

Email: daniel.mongrain@frequentis.com
Abstract

Additional Call Data is sent in a SIP Call-Info header or in a provided-by element of a PIDF-LO. Sometimes, the information needs to be updated while an emergency call is in progress. It is best for the Public Safety Answering Point (PSAP) to control the timing and frequency of updates. This document describes a SIP Subscribe/Notify Package to supply updates of Additional Call Data.
the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction ........................................... 2
2. Terminology ............................................ 2
3. SUBSCRIBE/NOTIFY Package for Additional Call Data .......... 2
   3.1. Event Package Name .................................. 3
   3.2. Event Package Parameters ............................. 3
   3.3. SUBSCRIBE Bodies ..................................... 3
   3.4. Subscription Duration ................................ 3
   3.5. NOTIFY Bodies ......................................... 3
   3.6. Notifier Processing of SUBSCRIBE Requests ............ 3
   3.7. Notifier Generation of NOTIFY Requests ................ 3
   3.8. Subscriber Processing of NOTIFY Requests ............. 4
   3.9. Handling of Forked Requests ........................... 4
   3.10. Rate of Notification .................................. 4
4. SUBSCRIBE Additional Data Block ............................ 4
   4.1. Update SUBSCRIBE URI .................................. 4
5. Security Considerations .................................... 4
6. Privacy Considerations .................................... 5
7. IANA Considerations ...................................... 5
   7.1. Event Package Registration ............................. 5
   7.2. MIME Content-type Registration for
        'application/emergencyCall.SvcInfo+xml' ............. 5
   7.3. Block Registration ..................................... 6
8. Normative References ..................................... 6
Author’s Address ........................................... 7

1. Introduction

This document provides a mechanism to update Additional Call Data sent with an emergency call as described in
[I-D.ietf-ecrit-additional-data] using the SIP SUBSCRIBE/NOTIFY method. It also defines a new block that provides the URL to which a
SUBSCRIBE can be sent by the PSAP to the provider of Additional Call Data.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
document are to be interpreted as described in RFC 2119 [RFC2119].

3. SUBSCRIBE/NOTIFY Package for Additional Call Data
This document defines an Event Package as defined in RFC 6655 [RFC6655]

3.1. Event Package Name

The name of this event package is "additional-call-data".

3.2. Event Package Parameters

This event package does not define any package parameters

3.3. SUBSCRIBE Bodies

This event package defines no message bodies to be used in the SUBSCRIBE message.

3.4. Subscription Duration

A subscription would not last longer than an emergency call, but the length of a call varies widely. A few minutes is a reasonable first subscription time. PSAPs should not expect a data source to accept subscriptions longer than 10 minutes.

3.5. NOTIFY Bodies

The content of a NOTIFY body will be a set of blocks as defined in [I-D.ietf-ecrit-additional-data]. No delta or difference mechanism is provided for, but a block that did not change from the prior transmission MAY be omitted. To get the subscription address, the PSAP would have to have gotten the entire block set, by value or by reference, and subsequent NOTIFY messages (including the initial one) need only contain blocks which have changed. Blocks that have not changed MAY be sent in any NOTIFY, at the option of the data provider.

3.6. Notifier Processing of SUBSCRIBE Requests

Upon receipt of a SUBSCRIBE request, the notifier applies authorization according to local policy. Typically, PSAPS will have credentials that may be useful to data providers in making such authorization decisions.

3.7. Notifier Generation of NOTIFY Requests

NOTIFY messages are generated whenever the data in one or more blocks change. Small changes in values that are not significant to handling emergency calls SHOULD NOT generate new NOTIFY requests.
3.8. Subscriber Processing of NOTIFY Requests

Upon receipt of a NOTIFY message, the subscriber applies any information in the message to update its view of the underlying data.

3.9. Handling of Forked Requests

Forking of Additional Call Data requests is not expected to occur. In the aberrant circumstance that a SUBSCRIBE request is forked, the subscriber SHOULD terminate all but one subscription.

3.10. Rate of Notification

While some data (e.g. sensor data) may change rapidly, PSAPs and responders cannot usually make use of a high rate of NOTIFY requests. Notifiers MUST implement event rate control RFC 6446 [RFC6446]. In the absence of an event rate filter, Notifiers MUST NOT send notifications more frequently than once every twenty seconds.

4. SUBSCRIBE Additional Data Block

This document defines a new Additional Data block type to contain the URI to send a SUBSCRIBE to.

4.1. Update SUBSCRIBE URI

Data Element: Update SUBSCRIBE URI

Use: Optional

XML Element: <UpdateSubscribeURI>

Description: If the data provider anticipates some block data may change during the processing of an emergency call, it MAY provide this URI to send a SUBSCRIBE to. This MUST be a SIP URI.

Reason for Need: Provide a PSAP controlled update mechanism for blocks that may change during an emergency call.

How Used by Call Taker: To obtain updates for Additional Call Data.

5. Security Considerations
Security considerations for the SUBSCRIBE/NOTIFY update mechanism are identical to those in [I-D.ietf-ecrit-additional-data]. The same credentials described in that document would be used to identify the PSAP and the data provider. The SUBSCRIBE URI should be protected against casual observation, and thus SIPS or HTTPS, as appropriate SHOULD be used on the original transmission of blocks which contains the SUBSCRIBE URI block.

Rapid updates could overwhelm PSAPs. The event rate controls defined in Section 3.10 are essential to allow PSAPs to control the update rate.

6. Privacy Considerations

The privacy considerations detailed in [I-D.ietf-ecrit-additional-data] apply to updates of the blocks as well as the original transmission.

7. IANA Considerations

7.1. Event Package Registration

This document defines a new Event Package as described in [RFC6655] and registers it in the Event packages and Event template-packages registry. The Package Name is "additional-call-data", The Type is "package". The contact is "Brian Rosen, br@brianrosen.net" and the Reference is this document.

7.2. MIME Content-type Registration for 'application/emergencyCall.SvcInfo+xml'

This specification requests the registration of a new MIME type according to the procedures of RFC 4288 [RFC4288] and guidelines in RFC 3023 [RFC3023].

MIME media type name: application

MIME subtype name: emergencyCall.UpdateSubscribeURI+xml

Mandatory parameters: none

Optional parameters: charset

Indicates the character encoding of enclosed XML.

Encoding considerations:
Uses XML, which can employ 8-bit characters, depending on the character encoding used. See Section 3.2 of RFC 3023 [RFC3023].

Security considerations:

This content type is designed to carry an event package subscription URI, which is a sub-category of additional data about an emergency call.

Please refer to Section 5 for more information about the sensitivity of the SUBSCRIBE URI.

Interoperability considerations: None

Published specification: [TBD: This document]

Applications which use this media type: Emergency Services

Additional information:

Magic Number: None

File Extension: .xml

Macintosh file type code: 'TEXT'

Person and email address for further information: Brian Rosen, br@brianrosen.net

Intended usage: LIMITED USE

Author: This specification is a work item of the IETF ECRIT working group, with mailing list address <ecrit@ietf.org>.

Change controller: The IESG <ietf@ietf.org>

7.3.  Block Registration

This document registers a new Additional Data block as defined in [I-D.ietf-ecrit-additional-data] and registers it in the Additional Call Data Blocks Registry. The Token is "UpdateSubscribeURI", the Reference is this document.

8.  Normative References

[I-D.ietf-ecrit-additional-data]


Author’s Address

Brian Rosen
NeuStar
470 Conrad Dr.
Mars, PA 16046
US

Phone: +1 724 382 1051
Email: br@brianrosen.net
Abstract

This document describes how to re-use the emergency services mechanisms specified for the Session Initiation Protocol (SIP) to accomplishing emergency calling support in vehicles. Profiling and simplifications are possible due to the nature of the functionality that is going to be provided in vehicles with the usage of Global Positioning System (GPS).

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 13, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
1. Introduction

Emergency calls made from vehicles can assist with the objective of significantly reducing road deaths and injuries. Unfortunately, drivers often have a poor location-awareness, especially on urban roads (also during night) and abroad. In the most crucial cases, the victim(s) may not be able to call because they have been injured or trapped.

In Europe the European Commission has launched the 'eCall' initiative that may best be described as a user-initiated or automatically triggered system to provide notifications to Public Safety Answering Points (PSAPs), by means of cellular communications, that a vehicle has crashed, and to provide geodetic location information and where possible a voice channel to the PSAP.

The general term for such systems is Automatic Crash Notification (ACN). ACN systems transmit some amount of data specific to the incident, referred to generally as "crash data." While different systems transmit different amounts of crash data, standardized formats, structures, and mechanisms are needed to provide interoperability among systems and PSAPs.

This document describes how existing IETF mechanisms are used to provide the realization of next-generation ACN in general, including European eCall.

This document registers the 'application/emergencyCall.VEDS+xml' MIME content-type, and registers the 'VEDS' entry in the Emergency Call Additional Data registry.
The Vehicle Emergency Data Set (VEDS) is an XML structure defined by the Association of Public-Safety Communications Officials (APCO) and the National Emergency Number Association (NENA). The 'application/emergencyCall.VEDS+xml' MIME content-type is used to identify it. The 'VEDS' entry in the Emergency Call Additional Data registry is used to construct a 'purpose' parameter value for conveying VEDS data in a Call-Info header.

Circuit-switched eCall systems transmit crash data as a defined set, the Minimum Set of Data (MSD) [eCall-MSD]. The MSD for circuit-switched eCall is a binary format defined by CEN, the European Committee for Standardization. It is expected that CEN will choose to define the XML schema for the eCall MSD for use in next-generation systems. Once this done, a MIME content-type (e.g., 'application/emergencyCall.eCall.MSD+xml') and Emergency Call Additional Data entry (e.g., 'eCall.MSD') need to be registered for the MSD. Note that Appendix Appendix A explains how the functionality available in IETF specifications maps to the functionality required for the MSD of the mobile circuit switched voice solution.

CEN and/or other entities may define additional sets of data in the same manner: a standardized format, such as XML, is defined, and a MIME content-type and Emergency Call Additional Data entry registered.

An In-Vehicle System (IVS) transmits crash data by encoding it in one of the standardized and registered formats (such as VEDS or eCall.MSD) and attaching it to an INVITE as a data block. The block is identified by its MIME content-type, and pointed to by a CID URL in a Call-Info header with a 'purpose' parameter value corresponding to the block.

The mechanisms described here can be used to deploy ACN systems in general including eCall by providing for emergency calls that are identifiable as ACN calls or specifically eCall calls and that carry one or more defined crash data objects.

1.1. Overview of Current Deployment Models

Current (circuit-switched or legacy) systems for placing emergency calls from vehicles, including automatic crash notification system, generally use one of three architectural models: Telematics Service Provider (TSP), direct, and paired handset. These three models are illustrated below.

In the TSP model the IVS transmits crash data to the TSP using proprietary means. The TSP operator bridges in the PSAP and communicates location, crash, and other data to the call taker verbally (there is a three-way voice call between the vehicle, the TSP, and the PSAP).
In the paired model the IVS uses a Bluetooth link to a previously-paired handset to establish an emergency call with the PSAP and then communicates location data to the PSAP via text-to-speech; crash data is not conveyed.

In the direct model the IVS communicates crash data to the PSAP via the eCall in-band modem (in the voice call).

1.2. Migration to IP-based Models

The migration to next-generation (all-IP) would then look like as follows.

In the TSP model the IVS transmits crash data to the TSP using either proprietary or standard means. The TSP bridges in the PSAP and transmits crash and other data to the PSAP using IETF specifications. There is a three-way call between the vehicle, the TSP, and the PSAP.

In the paired model, the IVS uses a Bluetooth link to a previously-paired handset to establish an emergency call with the PSAP; it is not clear what facilities are or will be available for transmitting crash data.

In the direct model the IVS communicates crash data to PSAP using Internet protocols.
This document is focused on the interface to the PSAP, that is, how an emergency call (including location and crash data) is setup and data is transmitted to the PSAP using existing IETF specifications. The goal is to re-use existing specifications rather than to invent new. For the direct model (such as the European eCall), this is the end-to-end description. For the TSP model, this describes the right-hand side, leaving the left-hand side up to the entities involved (e.g., IVS and TSP vendors) who are then free to use the same mechanism as for the right-hand side or not.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

This document re-uses terminology defined in Section 3 of [RFC5012]. Additionally, we use the following abbreviations:

IVS: In-Vehicle System
TSP: Telematics Service Provider
MSD: Minimum Set of Data
VEDS: Vehicle Emergency Data Set
NENA: National Emergency Number Association
APCO: Association of Public-Safety Communications Officials
CEN: European Committee for Standardization
ESInet: Emergency Services IP network

3. Profile
In the context of emergency calls placed from a vehicle it is assumed that the car is equipped with a built-in GPS receiver. For this reason only geodetic location information will be sent within an emergency call. The following location shapes MUST be implemented: 2d and 3d Point (see Section 5.2.1 of [RFC5491]), Circle (see Section 5.2.3 of [RFC5491]), and Ellipsoid (see Section 5.2.7 of [RFC5491]). The coordinate reference systems (CRS) specified in [RFC5491] are also mandatory for this document. The <direction> element, as defined in [RFC5962] which indicates the direction of travel of the vehicle, is important for dispatch and hence it MUST be included in the PIDF-LO. The <heading> element specified in [RFC5962] MUST be implemented and MAY be included.

This specification also inherits the ability to utilize test call functionality from Section 15 of [RFC6881].

4. Example

Figure 7 shows an emergency call placed from a vehicle whereby location information information is directly attached to the SIP INVITE message itself. The call uses the request URI ’urn:service:sos.ecall.automatic’ service URN and is recognized as an emergency call because the request URI starts with ’urn:service:sos’. The VoIP provider routes the call to an Emergency services IP Network (ESInet), as for any emergency call. The ESInet routes the call to an appropriate PSAP using location information and the fact that that it is an eCall carrying crash data. (In deployments where there is no ESInet, the VoIP provider may route directly to an appropriate PSAP.) The emergency call continues towards the PSAP and in this example it hits the ESRP, as the entry point to the ESInet. Finally, the emergency call will be received by a call taker and first responders will be dispatched.
The example, shown in Figure 8, illustrates a SIP emergency call eCall INVITE that is being conveyed with location information encoded in a PIDF-LO and VEDS data.
INVITE urn:service:sos.ecall.automatic SIP/2.0
To: urn:service:sos.ecall.automatic
From: <sip:+13145551111@example.com>;tag=9fxced76sl
Call-ID: 3848276298220188511@atlanta.example.com
Geolocation: <cid:target123@example.com>
Geolocation-Routing: no
Call-Info: cid:1234567890@atlanta.example.com;
purpose=emergencyCallData.VEDS
Accept: application/sdp, application/pidf+xml
CSeq: 31862 INVITE
Content-Type: multipart/mixed; boundary=boundary1
Content-Length: ...
--boundary1

Content-Type: application/sdp

...Session Description Protocol (SDP) goes here
--boundary1

Content-Type: application/pidf+xml
Content-ID: <target123@atlanta.example.com>
<?xml version="1.0" encoding="UTF-8">
<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"
entity="sip:+13145551111@example.com">
<dm:device id="123">
  <gp:geopriv>
    <gp:location-info>
      <gml:Point srsName="urn:ogc:def:crs:EPSG::4326">
        <gml:pos>-34.407 150.883</gml:pos>
      </gml:Point>
      <dy:Dynamic>
        <dy:heading>278</dy:heading>
        <dy:direction><dy:direction>
        </dy:Dynamic>
      </dy:Dynamic>
    </gp:location-info>
    <gp:usage-rules/>
    <method>gps</method>
  </gp:geopriv>
  <timestamp>2012-04-5T10:18:29Z</timestamp>
  <dm:deviceID>1M8GDM9A_KP042788</dm:deviceID>
</dm:device>
</presence>

--boundary1
5. Security Considerations

This document does not raise security considerations beyond those described in [RFC5069]. As with emergency service systems with end host provided location information there is the possibility that that location is incorrect, either intentionally (in case of a denial of service attack against the emergency services infrastructure) or due to a malfunctioning devices. The reader is referred to [I-D.ietf-ecrit-trustworthy-location] for a discussion of some of these vulnerabilities.

6. IANA Considerations

6.1. Service URN Registration

IANA is requested to register the URN 'urn:service:sos.ecall' under the sub-services 'sos' registry defined in Section 4.2 of [RFC5031].

This service identifier reaches a public safety answering point (PSAP), which in turn dispatches aid appropriate to the emergency related to accidents of vehicles. Two sub-services are registered as well, namely

urn:service:sos.ecall.manual

This service URN indicates that an eCall had been triggered based on the manual interaction of the driver or a passenger.

urn:service:sos.ecall.automatic

This service URN indicates that an eCall had been triggered automatically, for example, due to a crash. No human involvement was detected.

6.2. MIME Content-type Registration for 'application/emergencyCall.VEDS+xml'

This specification requests the registration of a new MIME type according to the procedures of RFC 4288 [RFC4288] and guidelines in RFC 3023 [RFC3023].

MIME media type name: application

MIME subtype name: emergencyCall.VEDS+xml

Mandatory parameters: none
Optional parameters: charset

Indicates the character encoding of enclosed XML.

Encoding considerations: Uses XML, which can employ 8-bit characters, depending on the character encoding used. See Section 3.2 of RFC 3023 [RFC3023].

Security considerations: This content type is designed to carry vehicle crash data during an emergency call. This data may contain personal information including vehicle VIN, location, direction, etc. Appropriate precautions need to be taken to limit unauthorized access, inappropriate disclosure to third parties, and eavesdropping of this information. Please refer to Section 7 and Section 8 of [I-D.ietf-ecrit-additional-data] for more information.

Interoperability considerations: None

Published specification: [TBD: This specification]

Applications which use this media type: Emergency Services

Additional information: None

Magic Number: None

File Extension: .xml

Macintosh file type code: 'TEXT'

Person and email address for further information: Hannes Tschofenig, Hannes.Tschofenig@gmx.net

Intended usage: LIMITED USE

Author: This specification is a work item of the IETF ECRIT working group, with mailing list address <ecrit@ietf.org>.

Change controller: The IESG <ietf@ietf.org>

6.3. Registration of the 'VEDS' entry in the Emergency Call Additional Data registry

This specification requests IANA to add the 'VEDS' entry to the Emergency Call Additional Data registry, with a reference to this document. The Emergency Call Additional Data registry has been established by [I-D.ietf-ecrit-additional-data].

7. Contributors

We would like to thank Ulrich Dietz for his help with earlier versions of the document.
8. Acknowledgements

We would like to thank Michael Montag, Arnoud van Wijk, Ban Al-Bakri, and Gunnar Hellstroem for their feedback.

9. References

9.1. Normative References


9.2. Informative references

Appendix A. Matching Functionality with eCall Minimum Set of Data (MSD)

[eCall-MSD] outlines a number of data elements that are transmitted in an emergency call triggered by a vehicle. Note that the work on eCall for mobile circuit switched voice is constrained in a number of ways since legacy eCall uses an inband voice modem for backwards compatibility with the already deployed cellular infrastructure to transmit data from a vehicle to a PSAP. Since the functionality in this document is based on the Session Initiation Protocol (SIP) these limitations do not exist. As such, it is not useful to transmit the MSD inband in the voice channel but to rather use the SIP mechanisms standardized for emergency call handling. Any voice, video, or real-text communication will be negotiated using the Session Description Protocol (SDP), as shown in Figure 8, and the actual media stream will then take place in RTP packets. For transmitting location information an XML-based data structure had been defined, the so-called Presence Information Data Format Location Object (PIDF-LO).

The following list compares the eCall minimum set of data with the functionality provided in this document.

Version of the MSD Format: Conveying information in a SIP-based emergency call is accomplished by using XML payloads and XML provides namespace declarations that allow a recipient of that information to distinguish different versions and additional extensions. For example, if additional data about a vehicle is defined and can be transmitted by vehicle then a respective extension can be defined for use inside a previously-defined XML structure. One or more top-level structures can be transmitted using the mechanism defined in [I-D.ietf-ecrit-additional-data]. Selecting the appropriate extension point depends on the type of extension envisioned.

Message Identifier: Every SIP INVITE message contains a Call-ID, which is a globally unique identifier for this call.
Test Call Indication: A service URN starting with "test." indicates a request for an automated test. For example, "urn:service:test.sos.ecall.automatic" indicates such a test feature. This functionality is defined in [RFC6881].

Automatic Activation Indication: This document registers new service URNs, which allow the differentiation between manually and automatically triggered emergency calls. The two service URNs are: urn:service:sos.ecall.automatic and urn:service:sos.ecall.manual

Vehicle Identification: The PIDF data structure contains a deviceID field that holds the Vehicle Identification Number (VIN).

Timestamp of Incident Event: The PIDF-LO element contains the timestamp when the PIDF-LO was created, which is at the time of the incident.

Vehicle Location: The location of the vehicle is conveyed using the PIDF location object, as described in Section 3.

Vehicle Direction: The direction of the vehicle is part of location information, as described in Section 3.

Recent Vehicle Location: With this optional functionality multiple location objects may be required to be transported simultaneously. This can be achieved using <timed-presence>, defined in RFC 4481 [RFC4481].

Additional Data: [I-D.ietf-ecrit-additional-data] provides the ability to carry additional data for an emergency call.

While most fields have an equivalent already in the corresponding SIP emergency signaling payloads there are currently no fields defined in [I-D.ietf-ecrit-additional-data] that allow information about the "Vehicle Type Encoding", "Number of Passengers", and "Vehicle Propulsion Storage type" to be conveyed. Extensions for those fields will have to be defined.

Authors’ Addresses

Brian Rosen
NeuStar, Inc.
470 Conrad Dr
Mars, PA 16046
US

Email: br@brianrosen.net