

ECRIT
Internet-Draft
Intended status: Standards Track
Expires: April 19, 2014

B. Rosen
NeuStar
H. Tschofenig
Nokia Solutions and Networks
R. Marshall
TeleCommunication Systems, Inc.
R. Gellens
Qualcomm Technologies, Inc.
J. Winterbottom

October 16, 2013

Additional Data related to an Emergency Call
draft-ietf-ecrit-additional-data-14.txt

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 April 19, 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.

Table of Contents

1. Introduction	4
2. Terminology	6
3. Data Structures	6
3.1. Data Provider Information	7
3.1.1. Data Provider String	7
3.1.2. Data Provider ID	8
3.1.3. Data Provider ID Series	8
3.1.4. Type of Data Provider	9
3.1.5. Data Provider Contact URI	9
3.1.6. Data Provider Languages(s) Supported	10
3.1.7. xCard of Data Provider	10
3.1.8. Subcontractor Principal	11
3.1.9. Subcontractor Priority	11
3.1.10. emergencyCall.ProviderInfo Example	12
3.2. Service Information	14
3.2.1. Service Environment	14
3.2.2. Service Delivered by Provider to End User	15
3.2.3. Service Mobility Environment	16
3.2.4. emergencyCall.SvcInfo Example	17
3.3. Device Information	17
3.3.1. Device Classification	17
3.3.2. Device Manufacturer	19
3.3.3. Device Model Number	19
3.3.4. Unique Device Identifier	19
3.3.5. Type of Device Identifier	20
3.3.6. Device/Service Specific Additional Data Structure	20
3.3.7. Device/Service Specific Additional Data Structure Type	21
3.3.8. Choosing between defining a new type of block or new	

type of device/service specific additional data . . .	22
3.3.9. emergencyCall.DevInfo Example	22
3.4. Owner/Subscriber Information	23
3.4.1. Subscriber Data Privacy Indicator	23
3.4.2. xCard for Subscriber's Data	24
3.4.3. emergencyCall.SubInfo Example	24
3.5. Comment	26
3.5.1. Comment	27
3.5.2. emergencyCall.Comment Example	27
4. Transport	27
4.1. Transmitting Blocks using the Call-Info Header	29
4.2. Transmitting Blocks by Reference using the Provided-By Element	30
4.3. Transmitting Blocks by Value using the Provided-By Element	30
4.4. The Content-Disposition Parameter	31
5. Examples	32
6. XML Schemas	36
6.1. emergencyCall.ProviderInfo XML Schema	36
6.2. emergencyCall.SvcInfo XML Schema	37
6.3. emergencyCall.DevInfo XML Schema	38
6.4. emergencyCall.SubInfo XML Schema	39
6.5. emergencyCall.Comment XML Schema	39
6.6. Provided-By XML Schema	40
7. Security Considerations	42
8. Privacy Considerations	43
9. IANA Considerations	45
9.1. Registry creation	46
9.1.1. Provider ID Series Registry	46
9.1.2. Service Provider Type Registry	46
9.1.3. Service Delivered Registry	47
9.1.4. Device Classification Registry	47
9.1.5. Device ID Type Type Registry	47
9.1.6. Device/Service Data Type Registry	48
9.1.7. Additional Data Blocks Registry	48
9.2. 'emergencyCallData' Purpose Parameter Value	49
9.3. URN Sub-Namespace Registration for provided-by Registry Entry	49
9.4. MIME Registrations	49
9.4.1. MIME Content-type Registration for 'application/emergencyCall.ProviderInfo+xml'	49
9.4.2. MIME Content-type Registration for 'application/emergencyCall.SvcInfo+xml'	50
9.4.3. MIME Content-type Registration for 'application/emergencyCall.DevInfo+xml'	51
9.4.4. MIME Content-type Registration for 'application/emergencyCall.SubInfo+xml'	52
9.4.5. MIME Content-type Registration for	

'application/emergencyCall.Comment+xml'	53
9.5. URN Sub-Namespace Registration	54
9.5.1. Registration for urn:ietf:params:xml:ns:emergencyCallAddlData	54
9.5.2. Registration for urn:ietf:params:xml:ns:emergencyCallProviderInfo . .	55
9.5.3. Registration for urn:ietf:params:xml:ns:emergencyCall.SvcInfo	56
9.5.4. Registration for urn:ietf:params:xml:ns:emergencyCall.DevInfo	57
9.5.5. Registration for urn:ietf:params:xml:ns:emergencyCall.SubInfo	57
9.5.6. Registration for urn:ietf:params:xml:ns:emergencyCall.Comment	58
9.6. Schema Registrations	59
9.7. VCard Parameter Value Registration	60
10. Acknowledgments	60
11. References	60
11.1. Normative References	60
11.2. Informational References	61
Appendix A. XML Schema for vCard/xCard	62
Authors' Addresses	85

1. Introduction

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 emergency service number, 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 data communicated in an emergency call:

Data Associated with a Location: Location data is conveyed in the Presence Information Data Format Location Object (PIDF-LO) data structure originally 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

[I-D.ietf-geopriv-relative-location] (for relative location). There may be data that is specific to the location not available in the location data structure itself, such as floor plans, tenant and building owner contact data, heating, ventilation and air conditioning (HVAC) status, etc.

Data Associated with a Call: While 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 the 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 kind of device is being used, etc. Some data is device or service dependent data. For example, a car telematics system may have crash information. A medical monitoring device may have sensor data.

Data Associated with a Caller: This is personal data about a caller, such as medical information and emergency contact data.

This document only defines data structures relevant to data associated with the call but defines extension points for other data to be added via other specifications.

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 accessed.

This document defines the data structures and a way to communicate the information in several ways. Although current standardization efforts around IP-based emergency services are focused on the Session Initiation Protocol (SIP) and HTTP the data structures in XML format described in this document are usable for other communication systems as well. In Section 3 the data structures are defined and the SIP/HTTP transport components are defined in Section 4 to offer a clear separation between the two.

More technically, the data structure described in this document is represented as one or more "blocks" of information. Each of the blocks is an XML structure with an associated Multipurpose Internet Mail Extensions (MIME) type for encapsulation, and an extensible set of these types constitute the data set. A registry is defined to list the block types that may be included.

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.

In the data block definitions, see Section 3, the values for the "Use:" label are specified as one of:

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

'Conditional': means they MUST be present if the specified condition(s) is met. They MAY be present if the condition(s) is not met.

'Optional': means they 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 interchangeable but it would be accurate to use the term vCard only.

3. 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 data structure. The five data structures are:

'Data Provider': This block supplies name and contact information for the entity that created the data. Section 3.1 provides the details.

'Service Information': This block supplies information about the service. The description can be found in Section 3.2.

'Device Information': This block supplies information about the device placing the call. Device information can be found in Section 3.3.

'Owner/Subscriber': This block supplies information about the owner of the device or about the subscriber. Details can be found in Section 3.4.

'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 3.5.

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.

3.1. Data Provider Information

This block is intended to be provided 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 provided 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/emergencyCall.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

3.1.1. Data Provider String

Data Element: Data Provider String

Use: Required

XML Element: <DataProviderString>

Description: This is a plain language string suitable for displaying the name of the service provider that created the additional data structure. If the device created the structure the value is identical to the contact header in the SIP INVITE.

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

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.

3.1.2. Data Provider ID

Data Element: Data Provider ID

Use: Conditional. This data SHOULD be provided if the service provider or access provider is located in a jurisdiction that maintains such ids. For example, in North America, this would be a "NENA Company ID".

XML Element: <ProviderID>

Description: A jurisdiction specific code for the access provider or service provider shown in the <DataProvidedBy> element that created the structure of the call. NOTE: In the US, the NENA Company ID must appear here. Additional information can be found at <http://www.nena.org/nena-company-id>. The NENA Company ID MUST be in the form of a URI in the following format:
urn:nena:companyid:<NENA Company ID>

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

How Used by Call Taker: Where jurisdictions have lists of providers the Data Provider ID provides useful information about the data source.

3.1.3. Data Provider ID Series

Data Element: Data Provider ID Series

Use: Conditional. If Data Provider ID is provided, Data Provider ID Series is required.

XML Element: <ProviderIDSeries>

Description: Identifies the issuer of the ProviderId. A registry will reflect the following valid entries:

- * NENA
- * EENA

Reason for Need: Identifies how to interpret the Data Provider ID.

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

3.1.4. Type of Data Provider

Data Element: Type of Data Provider ID

Use: Conditional. If Data Provider ID is provided, Type of Data Provider ID is required.

XML Element: <TypeOfProviderID>

Description: Identifies the type of data provider id being supplied in the ProviderId data element. A registry with an initial set of values is shown in Figure 1.

Token	Description
Access Network Provider	Access network service provider
Service Provider	Calling or Origination telecom SP
Service Provider Subcontractor	A contractor to another kind of SP
Telematics Provider	A sensor based SP, especially vehicle based
Language Translation Provider	A spoken language translation SP
Emergency Service Provider	An emergency service provider conveying information to another emergency service provider.
Emergency Modality Translation	An emergency call specific modality translation service e.g., for sign language
Relay Provider	A interpretation SP, for example, video relay for sign language interpreting
Other	Any other type of service provider

Figure 1: Type of Data Provider ID Registry.

Reason for Need: Identifies what kind of data provider this is.

How Used by Call Taker: To decide who to contact when further information is needed

3.1.5. 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 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 would reflect the contact information of the owner of the device. If a telephone number is the contact address then it MUST be tel URI. If it is provided as a SIP URI then it MUST be in the form of sip:telephonenumber@serviceprovider:user=phone.

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

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

3.1.6. Data Provider Languages(s) Supported

Data Element: Data Provider Language(s) supported

Use: Required.

XML Element: <Language>

Description: The language used by the entity at the Data Provider Contact URI as an alpha 2-character code as defined in ISO 639-1:2002 Codes for the representation of names of languages -- Part 1: Alpha-2 code Multiple instances of this element may occur. Order is significant; preferred language should appear first. The content MUST reflect the languages supported at the contact URI.

Reason for Need: Information needed to determine if emergency service authority can communicate with the service provider or if an interpreter will be needed.

How Used by Call Taker: If call taker cannot speak language(s) 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).

3.1.7. xCard of Data Provider

Data Element: xCard of Data Provider

Use: Optional

XML Element: <DataProviderContact>

Description: 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 name of 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]. and is hereinafter referred to as "xCard"

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

How Used by Call Taker: Assists call taker by providing additional contact information that may not be included in the SIP invite or the PIDF-LO.

3.1.1.8. Subcontractor Principal

Data Element: Subcontractor Principal

Use: Conditional. This data is required if the Data Provider type is subcontractor.

XML Element: <SubcontratorPrincipal>

Description: If the data provider is a subcontractor to another provider, such as an access infrastructure provider or telematics 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.

3.1.1.9. Subcontractor Priority

Data Element: Subcontractor Priority

Use: Conditional. This data is required if the Data Provider type is "subcontractor".

XML Element: <SubcontractorPriority>

Description: If the subcontractor has to be contacted first then this element MUST have the value "sub". If the access or service provider 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.

3.1.10. emergencyCall.ProviderInfo Example

```
<?xml version="1.0" encoding="UTF-8"?>
<ad:emergencyCall.ProviderInfo
  xmlns:ad="urn:ietf:params:xml:ns:emergencyCall.ProviderInfo"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ad:DataProviderString>Example VoIP Provider
  </ad:DataProviderString>
  <ad:ProviderID>urn:nena:companyid:ID123</ad:ProviderID>
  <ad:ProviderIDSeries>NENA</ad:ProviderIDSeries>
  <ad:TypeOfProvider>Service Provider</ad:TypeOfProvider>
  <ad:ContactURI>sip:voip-provider@example.com</ad:ContactURI>
  <ad:Language>EN</ad:Language>
  <xc: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>
```

```
</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/>
  <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>
</xc:DataProviderContact>
</ad:emergencyCall.ProviderInfo>
```

Figure 2: emergencyCall.ProviderInfo Example.

3.2. 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/emergencyCall.SvcInfo+xml".

3.2.1. Service Environment

Data Element: Service Environment

Use: Required

XML Element: <SvcEnvironment>

Description: This element defines whether a call is from a business or residence caller. Currently, the only valid entries are 'Business' or 'Residence'.

Reason for Need: To assist in determining equipment and manpower requirements.

How Used by Call Taker: Information may be used to assist in determining equipment and manpower requirements for emergency responders. 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. 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 where the call originates from.

3.2.2. Service Delivered by Provider to End User

Data Element: Service Delivered by Provider to End User

Use: Required

XML Element: <SvcDelByProvider>

Description: This defines the type of service the end user has subscribed to. The implied mobility of this service cannot be relied upon. A registry with an initial set of values is defined in Figure 3.

Name	Description
Wrlless	Wireless Telephone Service: Includes Satellite, CDMA, GSM, Wi-Fi, WiMAX, LTE (Long Term Evolution)
Coin	Fixed Public Pay/Coin telephones: Any coin or credit card operated device
1way	One way outbound service
Prison	Inmate call/service
Temp	Soft dialtone/quick service/warm disconnect/suspended
MLTS	Multi-line telephone system: Includes all PBX, Centrex, key systems, Shared Tenant Service
SenseU	Sensor, unattended: Includes devices that generate DATA ONLY. This is one-way information exchange and there will be no other form of communication
SenseA	Sensor, attended: Includes devices that are supported by a monitoring service provider or automatically open a two-way communication path
POTS	Wireline: Plain Old Telephone Service
VOIP	VoIP Telephone Service: A type of service that offers communication over internet protocol, such as Fixed Nomadic, Mobile, ...
Remote	Off premise extension
Relay	Relay Service: a type of service where

	there is a human 3rd party agent who provides some kind of additional assistance to the caller. Includes sign language relay and telematics services which provide a service assistant on the call.	
--	---	--

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.

3.2.3. Service Mobility Environment

Data Element: Service Mobility Environment

Use: Required

XML Element: <SvcMobility>

Description: This provides the service providers view of the mobility of the caller. As the service provider may not know the characteristics of the actual access network used, the value not be relied upon. A registry will reflect the following initial valid entries:

- * Mobile: the device should be 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.

3.2.4. emergencyCall.SvcInfo Example

```
<?xml version="1.0" encoding="UTF-8"?>
<svc:emergencyCall.SvcInfo
  xmlns:svc="urn:ietf:params:xml:ns:emergencyCall.SvcInfo"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <svc:SvcEnvironment>Business</svc:SvcEnvironment>
  <svc:SvcDelByProvider>MLTS</svc:SvcDelByProvider>
  <svc:SvcMobility>Fixed</svc:SvcMobility>
</svc:emergencyCall.SvcInfo>
```

Figure 4: emergencyCall.SvcInfo Example.

3.3. 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/emergencyCall.DevInfo+xml".

3.3.1. 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.

Token	Description
Cordless	Cordless handset
Fixed	Fixed phone
Mobile	Mobile handset
ATA	analog terminal adapter
Satphone	Satellite phone
FSense	Stationary computing device (alarm system, data sensor)
Guard	Guardian devices
Desktop	Desktop PC
Laptop	Laptop computing device
Tablet	Tablet computing device
Alarm	Alarm system
MSense	Mobile Data sensor
Beacon	Personal beacons (spot)
Auto	Auto telematics
Truck	Truck telematics
Farm	Farm equipment telematics
Marine	Marine telematics
PDA	Personal digital assistant
PND	Personal navigation device)
SmrtPhn	Smart phone
Itab	Internet tablet
Game	Gaming console
Video	Video phone
Text	Other text device
SoftPhn	Soft phone or soft client software
NA	Not Available

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 assist with location of caller. For example, a cordless handset may be outside or next door. May

provide the calltaker some context about the caller, the capabilities of the device used for the call or the environment the device is being used in.

3.3.2. 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.

3.3.3. 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.

3.3.4. Unique Device Identifier

Data Element: Unique Device Identifier

Use: Optional

XML Element: <UniqueDeviceID>

Description: String that identifies the specific device making the call or creating an event.

Reason for Need: Uniquely identifies the device as opposed to any signaling identifiers encountered in the call signaling stream.

How Used by Call Taker: Probably not used by the calltaker; they would need to refer to management for investigation.

3.3.5. Type of Device Identifier

Data Element: Type of Device Identifier

Use: Conditional: must be provided if the DeviceID is provided

XML Element: <TypeOfDeviceID>

Description: Identifies the type of device identifier being generated in the unique device identifier data element. A registry with an initial set of values can be seen in Figure 6.

Token	Description
MEID	Mobile Equipment Identifier (CDMA)
ESN	Electronic Serial Number(GSM)
MAC	Media Access Control Address (IEEE)
WiMAX	Device Certificate Unique ID
IMEI	International Mobile Equipment ID (GSM)
UDI	Unique Device Identifier
RFID	Radio Frequency Identification
SN	Manufacturer Serial Number

Figure 6: Registry with Device Identifier Types.

Reason for Need: Identifies how to interpret the Unique Device Identifier.

How Used by Call Taker: Additional information that may be used to assist with call handling.

3.3.6. 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. An example is the Vehicular Emergency Data Set (VEDS) structure for a vehicle telematics device. 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 telematics created VEDS 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.

3.3.7. 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

- * VEDS

IEEE 1512 is the USDOT model for traffic incidents and VEDS provides data elements needed for an efficient emergency response to vehicular emergency 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.

3.3.8. 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 DevInfo 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 DevInfo 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 DevInfo.

Size: Information which may be very may be better sent in the DevInfo 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).

3.3.9. emergencyCall.DevInfo Example

```
<?xml version="1.0" encoding="UTF-8"?>
<svc:emergencyCall.DevInfo
  xmlns:svc="urn:ietf:params:xml:ns:emergencyCall.DevInfo"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <svc:DeviceClassification>Fixed phone</svc:DeviceClassification>
  <svc:DeviceMfgr>Nokia</svc:DeviceMfgr>
  <svc:DeviceModelNr>Lumia 800</svc:DeviceModelNr>
  <svc:UniqueDeviceID>35788104</svc:UniqueDeviceID>
  <svc:TypeOfDeviceID>IMEI</svc:TypeOfDeviceID>
</svc:emergencyCall.DevInfo>
```

Figure 7: emergencyCallDevInfo Example.

3.4. 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 subtype is "application/emergencyCall.Subscriber+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.

3.4.1. 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.

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

3.4.2. 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.

3.4.3. emergencyCall.SubInfo Example

```
<?xml version="1.0" encoding="UTF-8"?>
<ad:emergencyCall.SubInfo
  xmlns:ad="urn:ietf:params:xml:ns:emergencyCall.SubInfo"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  privacyRequested="false">
  <xc:SubscriberData xmlns:xc="urn:ietf:params:xml:ns:vcard-4.0">
    <vcards 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>
      </vcards>
    </xc:SubscriberData>
  </ad:emergencyCall.SubInfo>
```



```
<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>
  <parameters><type><text>work</text></type>
  </parameters>
  <text>Viagenie</text>
</org>
<adr>
  <parameters>
    <type><text>work</text></type>
    <label><text>Simon Perreault
      2875 boul. Laurier, suite D2-630
      Quebec, QC, Canada
      G1V 2M2</text></label>
  </parameters>
  <pobox/>
  <ext/>
  <street>2875 boul. Laurier, suite D2-630</street>
  <locality>Quebec</locality>
  <region>QC</region>
  <code>G1V 2M2</code>
  <country>Canada</country>
</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>
    </parameters>
    <text>simon.perreault@viagenie.ca</text>
</email>
<geo>
    <parameters><type><text>work</text></type>
    </parameters>
    <uri>geo:46.766336,-71.28955</uri>
</geo>
<key>
    <parameters><type><text>work</text></type>
    </parameters>
    <uri>
        http://www.viagenie.ca/simon.perreault/simon.asc
    </uri>
</key>
<tz><text>America/Montreal</text></tz>
<url>
    <parameters><type><text>home</text></type>
    </parameters>
    <uri>http://nomis80.org</uri>
</url>
</vcard>
</vcards>
</xc:SubscriberData>
</ad:emergencyCall.SubInfo>

```

Figure 8: emergencyCall.SubInfo Example.

3.5. 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/emergencyCall.Comment+xml"

3.5.1. Comment

Data Element: EmergencyCall.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

3.5.2. emergencyCall.Comment Example

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

Figure 9: EmergencyCall.Comment Example.

4. Transport

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. Besides a service provider 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 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.

One or more blocks of data registered in the Emergency Call Additional Data registry, as defined in Section 9.1, 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.

4.1. Transmitting Blocks using the Call-Info Header

A URI to a block MAY be inserted in a 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 'emergencyCall' prefix and any suffix such as '+xml') with a '.' separator. For example, when referencing a block with MIME type 'application/emergencyCall.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 a SIP message. In this case, Content Indirection (CID) [RFC2392] is used, with the CID URL referencing the MIME body part.

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.

4.2. 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 4.1).

For example, to reference a block with MIME type 'application/emergencyCall.ProviderInfo+xml', the 'purpose' parameter is set to 'emergencyCallData.ProviderInfo'. An example 'emergencyCallDataReference' element for this would be:

```
<emergencyCallDataReference ref="https://www.example.com/23sedde3"
  purpose="emergencyCallData.ProviderInfo"/>
```

4.3. 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 an additional data block by value within a 'Provided-By' element of a PIDF-LO. The 'emergencyCallDataValue' element has one attribute: 'purpose' to indicate the type of data block contained. The value of 'purpose' is the same as used in a 'Call-Info' header field (as specified in Section 4.1, and in Section 4.1). The same XML structure as would be contained in the corresponding MIME type body part is placed inside the 'emergencyCallDataValue' element.

For example:

```
<provided-by
  xmlns="urn:ietf:params:xml:ns:emergencyCallAddlData">
```

```
<emergencyCallData>
  <byRef purpose="emergencyCallData.SvcInfo"
    ref="https://example.com/ref2"/>
  <sub:emergencyCall.Comment
    xmlns:sub="urn:ietf:params:xml:ns:emergencyCall.Comment">
    <sub:Comment xml:lang="en">This is an example text.
  </sub:Comment>
</sub:emergencyCall.Comment>
</emergencyCallData>
<emergencyCallDataValue
  purpose="emergencyCallData.ProviderInfo">
  <ProviderID>Test</ProviderID>
  <ProviderIDSeries>NENA</ProviderIDSeries>
  <TypeOfProviderID>Access Infrastructure Provider
</TypeOfProviderID>
  <ContactURI>sip:15555550987@burf.example.com/user=phone
  </ContactURI>
</emergencyCallDataValue>
</provided-by>
```

Example Provided-By by Value.

4.4. 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, however, used by many extensions (including additional data) 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 here

--boundary1

Content-Type: application/pidf+xml
Content-ID: <target123@atlanta.example.com>
Content-Disposition: by-reference;handling=optional

...PIDF-LO goes here

--boundary1--

Content-Type: application/emergencyCall.ProviderInfo+xml
Content-ID: <1234567890@atlanta.example.com>
Content-Disposition: by-reference;handling=optional

...Additional data goes here

--boundary1--

```

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

5. Examples

This section provides three examples of communicating additional data. In Figure 11 additional data is communicated in a SIP INVITE per value. In Figure 12 we illustrate how additional data is added by a SIP proxy per reference. Finally, an example for including additional data in the <Provided-By> element of a PIDF-LO is illustrated.

```

INVITE sips:bob@biloxi.example.com SIP/2.0
Via: SIP/2.0/TLS pc33.atlanta.example.com;branch=z9hG4bK74bf9
Max-Forwards: 70
To: Bob <sips:bob@biloxi.example.com>
From: Alice <sips:alice@atlanta.example.com>;tag=9fxced76sl
Call-ID: 3848276298220188511@atlanta.example.com
Call-Info: <http://www.example.com/alice/photo.jpg> ;purpose=icon,
           <http://www.example.com/alice/> ;purpose=info,
           <cid:1234567890@atlanta.example.com>
           ;purpose=emergencyCallData.ProviderInfo
Geolocation: <cid:target123@atlanta.example.com>

```



```

Geolocation-Routing: no
Accept: application/sdp, application/pidf+xml,
       application/emergencyCallProviderinfo+xml
CSeq: 31862 INVITE
Contact: <sips:alice@atlanta.example.com>
Content-Type: multipart/mixed; boundary=boundary1

Content-Length: ...

--boundary1

Content-Type: application/sdp

...SDP goes here

--boundary1

Content-Type: application/pidf+xml
Content-ID: <target123@atlanta.example.com>
Content-Disposition: by-reference;handling=optional

...PIDF-LO goes here

--boundary1--

Content-Type: application/emergencyCall.ProviderInfo+xml
Content-ID: <1234567890@atlanta.example.com>
Content-Disposition: by-reference;handling=optional

...Additional data goes here

--boundary1--

```

Figure 11: Example: Attaching Additional Data via CID to a SIP INVITE.

```

INVITE sips:bob@biloxi.example.com SIP/2.0
Via: SIP/2.0/TLS pc33.atlanta.example.com;branch=z9hG4bK74bf9
Max-Forwards: 70
To: Bob <sips:bob@biloxi.example.com>
From: Alice <sips:alice@atlanta.example.com>;tag=9fxced76sl
Call-ID: 3848276298220188511@atlanta.example.com
Call-Info: <http://www.example.com/alice/photo.jpg> ;purpose=icon,
          <http://www.example.com/alice/> ;purpose=info,
          <https://www.example.com/abc123456/>
          ;purpose=emergencyCallData.ProviderInfo
Geolocation: <cid:target123@atlanta.example.com>

```

```

Geolocation-Routing: no
Accept: application/sdp, application/pidf+xml,
       application/emergencyCallProviderinfo+xml
CSeq: 31862 INVITE
Contact: <sips:alice@atlanta.example.com>
Content-Type: multipart/mixed; boundary=boundary1

Content-Length: ...

--boundary1

Content-Type: application/sdp

...SDP goes here

--boundary1

Content-Type: application/pidf+xml
Content-ID: <target123@atlanta.example.com>
Content-Disposition: by-reference;handling=optional

...PIDF-LO goes here

--boundary1--

```

Figure 12: Example: Attaching Additional Data per Reference in a SIP INVITE.

```

<?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:gpb="urn:ietf:params:xml:ns:pidf:geopriv10:basicPolicy"
  xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model"
  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>
          <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>2013-07-10T20:00:00Z
</gbp:retention-expiry>
</gp:usage-rules>
<gp:method>802.11</gp:method>
<provided-by
xmlns="urn:ietf:params:xml:ns:pidf:geopriv10:emergencyCallData">

    <emergencyCallDataReference purpose="emergencyCallData.SvcInfo"
        ref="https://example.com/ref2"/>

    <emergencyCallDataValue>
        <emergencyCall.ProviderInfo
            xmlns="urn:ietf:params:xml:ns:emergencyCall.ProviderInfo">
            <DataProviderString>University of California, Irvine
            </DataProviderString>
            <ProviderID>urn:nena:companyid:uci</ProviderID>
            <ProviderIDSeries>NENA</ProviderIDSeries>
            <TypeOfProvider>Other</TypeOfProvider>
            <ContactURI>tel:+1 9498245222</ContactURI>
            <Language>EN</Language>
        </emergencyCall.ProviderInfo>

        <emergencyCall.Comment
            xmlns="urn:ietf:params:xml:ns:emergencyCall.Comment">
            <Comment xml:lang="en">This is an example text.</Comment>
        </emergencyCall.Comment>

    </emergencyCallDataValue>
</provided-by>
</gp:geopriv>
<dm:deviceID>mac:1234567890ab</dm:deviceID>
<dm:timestamp>2013-07-09T20:57:29Z</dm:timestamp>
</dm:device>
</presence>

```

Figure 13: Example: Including Additional Data via the Provided-By Element in a PIDF-LO.

6. XML Schemas

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

6.1. emergencyCall.ProviderInfo XML Schema

```
<?xml version="1.0"?>
<xs:schema
  targetNamespace="urn:ietf:params:xml:ns:emergencyCall.ProviderInfo"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:pi="urn:ietf:params:xml:ns:emergencyCall.ProviderInfo"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  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:simpleType name="iso3166a2">
      <xs:restriction base="xs:token">
        <xs:pattern value="[A-Z]{2}"/>
      </xs:restriction>
    </xs:simpleType>

    <xs:element
      name="emergencyCall.ProviderInfo"
      type="pi:ProviderInfoType"/>

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

    <xs:complexType name="ProviderInfoType">
      <xs:sequence>
        <xs:element name="DataProviderString"
          type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="ProviderID">
```

```

        type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="ProviderIDSeries"
  type="xs:string" minOccurs="0" 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" type="pi:iso3166a2"
  minOccurs="0" maxOccurs="unbounded" />
<xs:element name="DataProviderContact"
  type="xc:vcardType" minOccurs="0"
  maxOccurs="1"/>
<xs:element name="SubcontratorPrincipal"
  type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="SubcontractorPriority"
  type="pi:SubcontractorPriorityType" minOccurs="0" maxOccur
s="1"/>

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

</xs:schema>

```

Figure 14: emergencyCall.ProviderInfo XML Schema.

6.2. emergencyCall.SvcInfo XML Schema

```

<?xml version="1.0"?>
<xs:schema
  targetNamespace="urn:ietf:params:xml:ns:emergencyCall.SvcInfo"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:svc="urn:ietf:params:xml:ns:emergencyCall.SvcInfo"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  elementFormDefault="qualified" attributeFormDefault="unqualified">

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

  <xs:element name="emergencyCall.SvcInfo" type="svc:SvcInfoType"/>

  <xs:complexType name="SvcInfoType">
    <xs:sequence>
      <xs:element name="SvcEnvironment"
        type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="SvcDelByProvider"
        type="xs:string" minOccurs="1" maxOccurs="1"/>
    
```

```

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

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

</xs:schema>

```

Figure 15: emergencyCall.SvcInfo XML Schema.

6.3. emergencyCall.DevInfo XML Schema

```

<?xml version="1.0"?>
<xs:schema
    targetNamespace="urn:ietf:params:xml:ns:emergencyCall.DevInfo"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:dev="urn:ietf:params:xml:ns:emergencyCall.DevInfo"
    xmlns:xml="http://www.w3.org/XML/1998/namespace"
    elementFormDefault="qualified" attributeFormDefault="unqualified">

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

    <xs:element name="emergencyCall.DevInfo" type="dev:DevInfoType"/>

    <xs:complexType name="DevInfoType">
        <xs:sequence>
            <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"
                type="xs:string" minOccurs="0" maxOccurs="1"/>
            <xs:element name="TypeOfDeviceID"
                type="xs:string" minOccurs="0" maxOccurs="1"/>
            <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>

</xs:schema>

```

Figure 16: emergencyCall.DevInfo XML Schema.

6.4. emergencyCall.SubInfo XML Schema

```

<?xml version="1.0"?>
<xs:schema
  targetNamespace="urn:ietf:params:xml:ns:emergencyCall.SubInfo"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:sub="urn:ietf:params:xml:ns:emergencyCall.SubInfo"
  xmlns:xc="urn:ietf:params:xml:ns:vcard-4.0"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  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="emergencyCall.SubInfo" type="sub:SubInfoType" />

  <xs:complexType name="SubInfoType">
    <xs:complexContent>
      <xs:sequence>
        <xs:element name="SubscriberData" type="xc:vcardType"
          minOccurs="0" maxOccurs="1" />

        <xs:any namespace="##other" processContents="lax"
          minOccurs="0" maxOccurs="unbounded" />
      </xs:sequence>
      <xs:attribute name="privacyRequested" type="xs:boolean" use="requi
red" />
    </xs:complexContent>
  </xs:complexType>

</xs:schema>

```

Figure 17: emergencyCall.SubInfo XML Schema.

6.5. emergencyCall.Comment XML Schema

```

<?xml version="1.0"?>

```

```

<xs:schema
  targetNamespace="urn:ietf:params:xml:ns:emergencyCall.Comment"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:com="urn:ietf:params:xml:ns:emergencyCall.Comment"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  elementFormDefault="qualified" attributeFormDefault="unqualified">

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

  <xs:element name="emergencyCall.Comment" type="com:CommentType"/>

  <xs:complexType name="CommentType">
    <xs:sequence>
      <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 18: EmergencyCall.Comment XML Schema.

6.6. Provided-By XML Schema

This section defines the Provided-By schema.

```

<?xml version="1.0"?>
<xs:schema
  targetNamespace=
    "urn:ietf:params:xml:ns:pidf:geopriv10:emergencyCallData"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ad="urn:ietf:params:xml:ns:pidf:geopriv10:emergencyCallData"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:pi="urn:ietf:params:xml:ns:emergencyCall.ProviderInfo"
  xmlns:svc="urn:ietf:params:xml:ns:emergencyCall.SvcInfo"

```



```
xmlns:dev="urn:ietf:params:xml:ns:emergencyCall.DevInfo"
xmlns:sub="urn:ietf:params:xml:ns:emergencyCall.SubInfo"
xmlns:com="urn:ietf:params:xml:ns:emergencyCall.Comment"
elementFormDefault="qualified" attributeFormDefault="unqualified">

<xs:import namespace="urn:ietf:params:xml:ns:emergencyCall.Comment"/>
<xs:import namespace="urn:ietf:params:xml:ns:emergencyCall.SubInfo"/>
<xs:import namespace="urn:ietf:params:xml:ns:emergencyCall.DevInfo"/>
<xs:import namespace="urn:ietf:params:xml:ns:emergencyCall.SvcInfo"/>
<xs:import namespace="urn:ietf:params:xml:ns:emergencyCall.ProviderInfo"/>

<xs:element name="provided-by" type="ad:provided-by-Type"/>

<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>

<!-- 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:anyURI"
        use="required"/>
      <xs:attribute name="ref" type="xs:anyURI"
        use="required"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
```

```
<!-- Additional Data By Value -->

<xs:complexType name="emergencyCallDataValueType">
  <xs:sequence>
    <xs:element name="emergencyCall.ProviderInfo"
      type="pi:ProviderInfoType"
      minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="emergencyCall.SvcInfo"
      type="svc:SvcInfoType"
      minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="emergencyCall.DevInfo"
      type="dev:DevInfoType"
      minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="emergencyCall.SubInfo"
      type="sub:SubInfoType"
      minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="emergencyCall.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 19: Provided-By XML Schema.

7. Security Considerations

The information in this data structure will usually be considered private. HTTPS is specified to require the provider of the information 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 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 authorities, globally recognized credentials are useful. This might be accomplished by extending the notion of the "forest guide" described in [RFC5222] to allow the forest guide to provide the credential of the PKI root for areas that it has coverage information for, but standards for such a mechanism are not yet available. In its absence, the data provider will need to obtain the root CA credentials for any areas it is willing to provide additional data by out of band means. 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 the right server. The emergency authorities could provide credentials, distinguishable from credentials it provides 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 above. In the absence of such credentials, the emergency authorities could maintain a list of local data providers' credentials provided to it out of band. At a minimum, the emergency 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 to service providers, and the solutions are the same.

8. Privacy Considerations

This document enables functionality for conveying additional information about the caller 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 callers identity is defined and use is mandatory. However, observance of this request for privacy and what information it relates to is controlled by the destination jurisdiction.

There are a number of privacy concerns with regular 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 in comparison to the benefit for the emergency caller

are often towards the latter. Hence, the data protection rights of individuals are often 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 Plain old telephone system (POTS). For this reason there are additional privacy threats to consider, which are described in more detail in [RFC6973].

Stored Data Compromise: First, 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 operators, 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: Furthermore, there is the risk of secondary use. Secondary use 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 security (while in transit with traditional communication security techniques, and while at rest using access control mechanisms) there is the risk of disclosure, which is the revelation of information about an individual that affects the way others judge the individual.

To mitigate these privacy risks the following countermeasures can be taken.

In regions where callers can elect to suppress certain personally identifying information, the 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 constitutes personal data. Local regulations may govern what data must be provided in emergency calls, but in general, the emergency call system is often aided by the kinds of information described in this document. There is a tradeoff between the privacy considerations and the utility of the data. For adequate protection this specification requires all data exchanges to be secured via communication security techniques (namely TLS) against eavesdropping and inception. Furthermore, security safeguards are required to prevent unauthorized access to data at rest. Various security incidents over the last 10 years have shown data breaches are not not uncommon and are often caused by lack of proper access control frameworks, software bugs (buffer overflows), or missing input parsing (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 to highlight 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 value (URL in 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 raises privacy concerns and intermediaries are involved transmitting a reference is more appropriate (assuming 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.

9. IANA Considerations

9.1. Registry creation

This document creates a new registry called 'Emergency Call Additional Data'. The following sub-registries are created in Emergency Call Additional Data:

9.1.1. 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.

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 20.

Name	Source	URL
NENA	National Emergency Number Association	http://www.nena.org
EENA	European Emergency Number Association	http://www.eena.org

Figure 20: Provider ID Series Registry.

9.1.2. 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 TypeOfServiceProvider element

The content of this registry includes:

Name: Value to be used in TypeOfServiceProvider.

Description: A short description of the type of service provider

The initial set of values is defined in Figure 1.

9.1.3. 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: Enumeration token of the service.

Description: Short description identifying the service.

The initial set of values are defined in Figure 3.

9.1.4. 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:

Name: Enumeration token of the device classification.

Description: Short description identifying the device type.

The initial set of values are defined in Figure 5.

9.1.5. 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:

Name: Enumeration token of the device id type.

Description: Short description identifying type of device id.

The initial set of values are defined in Figure 6.

9.1.6. 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:

Name: Enumeration token of the data type.

Description: Short description identifying the the data.

Specification: Citation for the specification of the data.

The initial set of values are listed in Figure 21.

Token	Description	Specification
IEE1512	Common Incident Management Message Set	IEEE 1512-2006
VEDS	Vehicle Emergency Data Set	APCO/NENA VEDS

Figure 21: Device/Service Data Type Registry.

9.1.7. Additional Data Blocks Registry

This document creates a new sub-registry called 'Additional Data Blocks' 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 content of this registry includes:

Name: Element Name of enclosing block.

Reference: The document that describes the block

The initial set of values are listed in Figure 22.

Token	Reference
ProviderInfo	[This RFC]
SvcInfo	[This RFC]
DevInfo	[This RFC]
Subscriber	[This RFC]
Comment	[This RFC]

Figure 22: Additional Data Blocks Registry.

9.2. 'emergencyCallData' Purpose Parameter Value

This document defines the 'emergencyCall' 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].

Header Field	Parameter Name	New Value	Reference
Call-Info	purpose	emergencyCall	[This RFC]

9.3. URN Sub-Namespace Registration for provided-by Registry Entry

This section registers the namespace specified in Section 9.5.1 in the provided-by registry established by RFC 4119, for usage within the <provided-by> element of a PIDF-LO.

The schema for the provided-by schema used by this document is specified in Section 6.6.

9.4. MIME Registrations

9.4.1. MIME Content-type Registration for 'application/emergencyCall.ProviderInfo+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.ProviderInfo+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 the data provider information, which is a sub-category of additional data about an emergency call. Since this data contains personal information appropriate precautions have 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 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>

9.4.2. 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.SvcInfo+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 the service information, which is a sub-category of additional data about an emergency call. Since this data contains personal information appropriate precautions have 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 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>

9.4.3. MIME Content-type Registration for 'application/emergencyCall.DevInfo+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.DevInfo+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 the device information information, which is a sub-category of additional data about an emergency call. Since this data contains personal information appropriate precautions have 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 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>

9.4.4. MIME Content-type Registration for 'application/emergencyCall.SubInfo+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.SubInfo+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 owner/subscriber information, which is a sub-category of additional data about an emergency call. Since this data contains personal information appropriate precautions have 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 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>

9.4.5. MIME Content-type Registration for 'application/emergencyCall.Comment+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.Comment+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 a comment, which is a sub-category of additional data about an emergency call. This data may contain personal information. Appropriate precautions may have 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 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>

9.5. URN Sub-Namespace Registration

9.5.1. Registration for urn:ietf:params:xml:ns:emergencyCallAddlData

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

URI: urn:ietf:params:xml:ns:emergencyCallAddlData

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
```

9.5.2. Registration for urn:ietf:params:xml:ns:emergencyCallProviderInfo

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

URI: urn:ietf:params:xml:ns:emergencyCallProviderInfo

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:
        Data Provider Information</title>
</head>
<body>
  <h1>Namespace for Additional Data related to an Emergency Call</h1>
  <h2>Data Provider Information</h2>
  <p>See [TBD: This document].</p>
</body>
</html>
END
```

9.5.3. Registration for urn:ietf:params:xml:ns:emergencyCall.SvcInfo

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

URI: urn:ietf:params:xml:ns:emergencyCall.SvcInfo

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:
        Service Information</title>
</head>
<body>
  <h1>Namespace for Additional Data related to an Emergency Call</h1>
  <h2>Service Information</h2>
  <p>See [TBD: This document].</p>
</body>
</html>
END
```


9.5.4. Registration for urn:ietf:params:xml:ns:emergencyCall.DevInfo

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

URI: urn:ietf:params:xml:ns:emergencyCall.DevInfo

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:
    Device Information</title>
</head>
<body>
  <h1>Namespace for Additional Data related to an Emergency Call</h1>
  <h2>Device Information</h2>
  <p>See [TBD: This document].</p>
</body>
</html>
END
```

9.5.5. Registration for urn:ietf:params:xml:ns:emergencyCall.SubInfo

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

URI: urn:ietf:params:xml:ns:emergencyCall.SubInfo

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:
    Owner/Subscriber Information</title>
</head>
<body>
  <h1>Namespace for Additional Data related to an Emergency Call</h1>
  <h2> Owner/Subscriber Information</h2>
  <p>See [TBD: This document].</p>
</body>
</html>
END
```

9.5.6. Registration for urn:ietf:params:xml:ns:emergencyCall.Comment

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

URI: urn:ietf:params:xml:ns:emergencyCall.Comment

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:Comment</title>
</head>
<body>
  <h1>Namespace for Additional Data related to an Emergency Call</h1>
  <h2> Comment</h2>
  <p>See [TBD: This document].</p>
</body>
```

</html>
END

9.6. Schema Registrations

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

URI: urn:ietf:params:xml:schema:additional-data:emergencyCallProviderInfo

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 14.

URI: urn:ietf:params:xml:schema:additional-data:addCallSvcInfo

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 15.

URI: urn:ietf:params:xml:schema:additional-data:emergencyCallDevInfo

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 16.

URI: urn:ietf:params:xml:schema:additional-data:emergencyCall.SubInfo

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 Section 6.4.

URI: urn:ietf:params:xml:schema:additional-data:emergencyCall.Comment

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 Section 6.5.

9.7. 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

10. 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.

We would also like to thank Paul Kyzivat, Gunnar Hellstrom, Martin Thomson, Keith Drage, Laura Liess, and Barbara Stark for their review comments.

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2392] Levinson, E., "Content-ID and Message-ID Uniform Resource Locators", RFC 2392, August 1998.
- [RFC3023] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", RFC 3023, January 2001.
- [RFC3204] Zimmerer, E., Peterson, J., Vemuri, A., Ong, L., Audet, F., Watson, M., and M. Zonoun, "MIME media types for ISUP and QSIG Objects", RFC 3204, December 2001.

- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.
- [RFC3325] Jennings, C., Peterson, J., and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", RFC 3325, November 2002.
- [RFC3459] Burger, E., "Critical Content Multi-purpose Internet Mail Extensions (MIME) Parameter", RFC 3459, January 2003.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.
- [RFC4119] Peterson, J., "A Presence-based GEOPRIV Location Object Format", RFC 4119, December 2005.
- [RFC4288] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", RFC 4288, December 2005.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC5621] Camarillo, G., "Message Body Handling in the Session Initiation Protocol (SIP)", RFC 5621, September 2009.
- [RFC6350] Perreault, S., "vCard Format Specification", RFC 6350, August 2011.
- [RFC6351] Perreault, S., "xCard: vCard XML Representation", RFC 6351, August 2011.

11.2. Informational References

- [I-D.ietf-geopriv-relative-location]
Thomson, M., Rosen, B., Stanley, D., Bajko, G., and A. Thomson, "Relative Location Representation", draft-ietf-geopriv-relative-location-08 (work in progress), September 2013.
- [RFC5012] Schulzrinne, H. and R. Marshall, "Requirements for Emergency Context Resolution with Internet Technologies", RFC 5012, January 2008.

- [RFC5139] Thomson, M. and J. Winterbottom, "Revised Civic Location Format for Presence Information Data Format Location Object (PIDF-LO)", RFC 5139, February 2008.
- [RFC5222] Hardie, T., Newton, A., Schulzrinne, H., and H. Tschofenig, "LoST: A Location-to-Service Translation Protocol", RFC 5222, August 2008.
- [RFC5491] Winterbottom, J., Thomson, M., and H. Tschofenig, "GEOPRIV Presence Information Data Format Location Object (PIDF-LO) Usage Clarification, Considerations, and Recommendations", RFC 5491, March 2009.
- [RFC5962] Schulzrinne, H., Singh, V., Tschofenig, H., and M. Thomson, "Dynamic Extensions to the Presence Information Data Format Location Object (PIDF-LO)", RFC 5962, September 2010.
- [RFC5985] Barnes, M., "HTTP-Enabled Location Delivery (HELD)", RFC 5985, September 2010.
- [RFC6443] Rosen, B., Schulzrinne, H., Polk, J., and A. Newton, "Framework for Emergency Calling Using Internet Multimedia", RFC 6443, December 2011.
- [RFC6848] Winterbottom, J., Thomson, M., Barnes, R., Rosen, B., and R. George, "Specifying Civic Address Extensions in the Presence Information Data Format Location Object (PIDF-LO)", RFC 6848, January 2013.
- [RFC6881] Rosen, B. and J. Polk, "Best Current Practice for Communications Services in Support of Emergency Calling", BCP 181, RFC 6881, March 2013.
- [RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, July 2013.

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.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  targetNamespace="urn:ietf:params:xml:ns:vcard-4.0"
  xmlns:ns1="urn:ietf:params:xml:ns:vcard-4.0">
  <!--

    3.3
    iana-token = xsd:string { pattern = "[a-zA-Z0-9-]+" }
    x-name = xsd:string { pattern = "x-[a-zA-Z0-9-]+" }
  -->
  <xs:simpleType name="iana-token">
    <xs:annotation>
      <xs:documentation>vCard Format Specification
    </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
  <xs:simpleType name="x-name">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
  <!--

    4.1
  -->
  <xs:element name="text" type="xs:string"/>
  <xs:group name="value-text-list">
    <xs:sequence>
      <xs:element maxOccurs="unbounded" ref="ns1:text"/>
    </xs:sequence>
  </xs:group>
  <!-- 4.2 -->
  <xs:element name="uri" type="xs:anyURI"/>
  <!-- 4.3.1 -->
  <xs:element name="date"
    substitutionGroup="ns1:value-date-and-or-time">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:pattern value="\d{8}|\d{4}-\d\d|
          --\d\d(\d\d)?|---\d\d"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <!-- 4.3.2 -->
  <xs:element name="time"
    substitutionGroup="ns1:value-date-and-or-time">
    <xs:simpleType>
      <xs:restriction base="xs:string">
```

```

        <xs:pattern value="(\d\d(\d\d(\d\d)?)?|-\d\d(\d\d)?)|--\d\d)
        (Z|[+|-]\d\d(\d\d)?)" />
    </xs:restriction>
</xs:simpleType>
</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)?)?(Z|[+|-]\d\d(\d\d)?)" />
        </xs:restriction>
    </xs:simpleType>
</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 name="timestamp">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:pattern value="\d{8}T\d{6}(Z|[+|-]\d\d(\d\d)?)" />
        </xs:restriction>
    </xs:simpleType>
</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:simpleType>
</xs:element>
<!-- 4.8 -->
<xs:element name="language-tag">
    <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[0-9a-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>
</xs:element>
<!--

    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>
<!-- 5.4 -->
<xs:group name="param-altid">
  <xs:sequence>
    <xs:element minOccurs="0" ref="nsl:altid"/>
  </xs:sequence>
</xs:group>
<xs:element name="altid">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="nsl:text"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!-- 5.5 -->
<xs:group name="param-pid">
  <xs:sequence>
    <xs:element minOccurs="0" ref="nsl: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="nsl: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:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<!-- 5.7 -->
<xs:group name="param-mediatype">
  <xs:sequence>
    <xs:element minOccurs="0" ref="ns1:mediatype"/>
  </xs:sequence>
</xs:group>
<xs:element name="mediatype">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ns1:text"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!-- 5.8 -->
<xs:group name="param-calscale">
  <xs:sequence>
    <xs:element minOccurs="0" ref="ns1:calscale"/>
  </xs:sequence>
</xs:group>
<xs:element name="calscale">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="text">
        <xs:simpleType>
          <xs:restriction base="xs:token">
            <xs:enumeration value="gregorian"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!-- 5.9 -->
<xs:group name="param-sort-as">
  <xs:sequence>
    <xs:element minOccurs="0" ref="ns1:sort-as"/>
  </xs:sequence>
</xs:group>
<xs:element name="sort-as">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" ref="ns1:text"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

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

```
<!-- 6.1.4 -->
<xs:element name="kind">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="text">
        <xs:simpleType>
          <xs:union memberTypes="nsl:x-name nsl:iana-token">
            <xs:simpleType>
              <xs:restriction base="xs:token">
                <xs:enumeration value="individual"/>
              </xs:restriction>
            </xs:simpleType>
            <xs:simpleType>
              <xs:restriction base="xs:token">
                <xs:enumeration value="group"/>
              </xs:restriction>
            </xs:simpleType>
            <xs:simpleType>
              <xs:restriction base="xs:token">
                <xs:enumeration value="org"/>
              </xs:restriction>
            </xs:simpleType>
            <xs:simpleType>
              <xs:restriction base="xs:token">
                <xs:enumeration value="location"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:union>
        </xs:simpleType>
      </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="nsl:param-language"/>
            <xs:group ref="nsl:param-altid"/>
            <xs:group ref="nsl:param-pid"/>
            <xs:group ref="nsl:param-pref"/>
            <xs:group ref="nsl:param-type"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
        <xs:element ref="nsl:text"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <!-- 6.2.2 -->
  <xs:element name="n">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="parameters">
          <xs:complexType>
            <xs:sequence>
              <xs:group ref="nsl:param-language"/>
              <xs:group ref="nsl:param-sort-as"/>
              <xs:group ref="nsl:param-altid"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element maxOccurs="unbounded" ref="nsl:surname"/>
        <xs:element maxOccurs="unbounded" ref="nsl:given"/>
        <xs:element maxOccurs="unbounded" ref="nsl:additional"/>
        <xs:element maxOccurs="unbounded" ref="nsl:prefix"/>
        <xs:element maxOccurs="unbounded" ref="nsl: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 name="suffix" type="xs:string"/>
  <!-- 6.2.3 -->
  <xs:element name="nickname">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="parameters">
          <xs:complexType>
            <xs:sequence>
              <xs:group ref="nsl:param-language"/>
              <xs:group ref="nsl:param-altid"/>
              <xs:group ref="nsl:param-pid"/>
              <xs:group ref="nsl:param-pref"/>
              <xs:group ref="nsl:param-type"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:group ref="nsl:value-text-list"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

```
</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="nsl:param-altid"/>
            <xs:group ref="nsl:param-pid"/>
            <xs:group ref="nsl:param-pref"/>
            <xs:group ref="nsl:param-type"/>
            <xs:group ref="nsl:param-mediatype"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element ref="nsl: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="nsl:param-altid"/>
            <xs:group ref="nsl:param-calscale"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:choice>
        <xs:element ref="nsl:value-date-and-or-time"/>
        <xs:element ref="nsl: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="nsl:param-altid"/>
            <xs:group ref="nsl:param-calscale"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:choice>
      <xs:element ref="nsl:value-date-and-or-time"/>
      <xs:element ref="nsl:text"/>
    </xs:choice>
  </xs:sequence>
</xs:complexType>
</xs:element>
<!-- 6.2.7 -->
<xs:element name="gender">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="nsl:sex"/>
      <xs:element minOccurs="0" ref="nsl: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="nsl:label"/>
  </xs:sequence>
</xs:group>
<xs:element name="label">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="nsl:text"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="adr">
  <xs:complexType>
    <xs:sequence>
```



```
<xs:element minOccurs="0" name="parameters">
  <xs:complexType>
    <xs:sequence>
      <xs:group ref="nsl:param-language"/>
      <xs:group ref="nsl:param-altid"/>
      <xs:group ref="nsl:param-pid"/>
      <xs:group ref="nsl:param-pref"/>
      <xs:group ref="nsl:param-type"/>
      <xs:group ref="nsl:param-geo"/>
      <xs:group ref="nsl:param-tz"/>
      <xs:group ref="nsl:param-label"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element maxOccurs="unbounded" ref="nsl:pobox"/>
<xs:element maxOccurs="unbounded" ref="nsl:ext"/>
<xs:element maxOccurs="unbounded" ref="nsl:street"/>
<xs:element maxOccurs="unbounded" ref="nsl:locality"/>
<xs:element maxOccurs="unbounded" ref="nsl:region"/>
<xs:element maxOccurs="unbounded" ref="nsl:code"/>
<xs:element maxOccurs="unbounded" ref="nsl:country"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="pobox" type="xs:string"/>
<xs:element name="ext" type="xs:string"/>
<xs:element name="street" type="xs:string"/>
<xs:element name="locality" type="xs:string"/>
<xs:element name="region" type="xs:string"/>
<xs:element name="code" type="xs:string"/>
<xs:element name="country" type="xs:string"/>
<!-- 6.4.1 -->
<xs:element name="tel">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="parameters">
        <xs:complexType>
          <xs:sequence>
            <xs:group ref="nsl:param-altid"/>
            <xs:group ref="nsl:param-pid"/>
            <xs:group ref="nsl: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: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:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:group ref="nsl:param-mediatype"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:choice>
    <xs:element ref="nsl:text"/>
    <xs:element ref="nsl:uri"/>
</xs:choice>
</xs:sequence>
</xs:complexType>
</xs:element>
<!-- 6.4.2 -->
<xs:element name="email">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="parameters">
                <xs:complexType>
                    <xs:sequence>
                        <xs:group ref="nsl:param-altid"/>
                        <xs:group ref="nsl:param-pid"/>
                        <xs:group ref="nsl:param-pref"/>
                        <xs:group ref="nsl:param-type"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element ref="nsl:text"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<!-- 6.4.3 -->
<xs:element name="impp">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="parameters">
```

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

```
        </xs:element>
        <xs:choice>
          <xs:element ref="ns1:text"/>
          <xs:element ref="ns1:uri"/>
          <xs:element ref="ns1:utc-offset"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:sequence>
</xs:group>
<!-- 6.5.2 -->
<xs:group name="property-geo">
  <xs:sequence>
    <xs:element name="geo">
      <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:sequence>
</xs:group>
<!-- 6.6.1 -->
<xs:element name="title">
  <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>
    <xs:element ref="nsl:text"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
<!-- 6.6.2 -->
<xs:element name="role">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="parameters">
        <xs:complexType>
          <xs:sequence>
            <xs:group ref="nsl:param-language"/>
            <xs:group ref="nsl:param-altid"/>
            <xs:group ref="nsl:param-pid"/>
            <xs:group ref="nsl:param-pref"/>
            <xs:group ref="nsl:param-type"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element ref="nsl:text"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!-- 6.6.3 -->
<xs:element name="logo">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="parameters">
        <xs:complexType>
          <xs:sequence>
            <xs:group ref="nsl:param-language"/>
            <xs:group ref="nsl:param-altid"/>
            <xs:group ref="nsl:param-pid"/>
            <xs:group ref="nsl:param-pref"/>
            <xs:group ref="nsl:param-type"/>
            <xs:group ref="nsl:param-mediatype"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element ref="nsl:uri"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!-- 6.6.4 -->
<xs:element name="org">
  <xs:complexType>
    <xs:sequence>
```

```
<xs:element minOccurs="0" name="parameters">
  <xs:complexType>
    <xs:sequence>
      <xs:group ref="nsl:param-language"/>
      <xs:group ref="nsl:param-altid"/>
      <xs:group ref="nsl:param-pid"/>
      <xs:group ref="nsl:param-pref"/>
      <xs:group ref="nsl:param-type"/>
      <xs:group ref="nsl:param-sort-as"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:group ref="nsl:value-text-list"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<!-- 6.6.5 -->
<xs:element name="member">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="parameters">
        <xs:complexType>
          <xs:sequence>
            <xs:group ref="nsl:param-altid"/>
            <xs:group ref="nsl:param-pid"/>
            <xs:group ref="nsl:param-pref"/>
            <xs:group ref="nsl:param-mediatype"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element ref="nsl:uri"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!-- 6.6.6 -->
<xs:element name="related">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="parameters">
        <xs:complexType>
          <xs:sequence>
            <xs:group ref="nsl:param-altid"/>
            <xs:group ref="nsl:param-pid"/>
            <xs:group ref="nsl: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:enumeration value="home"/>
    <xs:enumeration value="contact"/>
    <xs:enumeration value="acquaintance"/>
    <xs:enumeration value="friend"/>
    <xs:enumeration value="met"/>
    <xs:enumeration value="co-worker"/>
    <xs:enumeration value="colleague"/>
    <xs:enumeration value="co-resident"/>
    <xs:enumeration value="neighbor"/>
    <xs:enumeration value="child"/>
    <xs:enumeration value="parent"/>
    <xs:enumeration value="sibling"/>
    <xs:enumeration value="spouse"/>
    <xs:enumeration value="kin"/>
    <xs:enumeration value="muse"/>
    <xs:enumeration value="crush"/>
    <xs:enumeration value="date"/>
    <xs:enumeration value="sweetheart"/>
    <xs:enumeration value="me"/>
    <xs:enumeration value="agent"/>
    <xs:enumeration value="emergency"/>
  </xs:restriction>
</xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:group ref="nsl:param-mediatype"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:choice>
  <xs:element ref="nsl:uri"/>
  <xs:element ref="nsl:text"/>
</xs:choice>
</xs:sequence>
</xs:complexType>
</xs:element>
<!-- 6.7.1 -->
<xs:element name="categories">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="parameters">
        <xs:complexType>
          <xs:sequence>
```

```
        <xs:group ref="nsl:param-altid"/>
        <xs:group ref="nsl:param-pid"/>
        <xs:group ref="nsl:param-pref"/>
        <xs:group ref="nsl:param-type"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:group ref="nsl:value-text-list"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<!-- 6.7.2 -->
<xs:element name="note">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="parameters">
        <xs:complexType>
          <xs:sequence>
            <xs:group ref="nsl:param-language"/>
            <xs:group ref="nsl:param-altid"/>
            <xs:group ref="nsl:param-pid"/>
            <xs:group ref="nsl:param-pref"/>
            <xs:group ref="nsl:param-type"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element ref="nsl:text"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!-- 6.7.3 -->
<xs:element name="prodid">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="nsl:text"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!-- 6.7.4 -->
<xs:element name="rev" type="nsl:value-timestamp"/>
<!-- 6.7.5 -->
<xs:element name="sound">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="parameters">
        <xs:complexType>
          <xs:sequence>
            <xs:group ref="nsl:param-language"/>
```



```
        <xs:group ref="nsl:param-altid"/>
        <xs:group ref="nsl:param-pid"/>
        <xs:group ref="nsl:param-pref"/>
        <xs:group ref="nsl:param-type"/>
        <xs:group ref="nsl:param-mediatype"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element ref="nsl:uri"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<!-- 6.7.6 -->
<xs:element name="uid">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="nsl:uri"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!-- 6.7.7 -->
<xs:element name="clientpidmap">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="nsl:sourceid"/>
      <xs:element ref="nsl: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="nsl:param-altid"/>
            <xs:group ref="nsl:param-pid"/>
            <xs:group ref="nsl:param-pref"/>
            <xs:group ref="nsl:param-type"/>
            <xs:group ref="nsl:param-mediatype"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element ref="nsl:uri"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
</xs:element>
<!-- 6.8.1 -->
<xs:element name="key">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="parameters">
        <xs:complexType>
          <xs:sequence>
            <xs:group ref="nsl:param-altid"/>
            <xs:group ref="nsl:param-pid"/>
            <xs:group ref="nsl:param-pref"/>
            <xs:group ref="nsl:param-type"/>
            <xs:group ref="nsl:param-mediatype"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:choice>
        <xs:element ref="nsl:uri"/>
        <xs:element ref="nsl:text"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!-- 6.9.1 -->
<xs:element name="fburl">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="parameters">
        <xs:complexType>
          <xs:sequence>
            <xs:group ref="nsl:param-altid"/>
            <xs:group ref="nsl:param-pid"/>
            <xs:group ref="nsl:param-pref"/>
            <xs:group ref="nsl:param-type"/>
            <xs:group ref="nsl:param-mediatype"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element ref="nsl:uri"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!-- 6.9.2 -->
<xs:element name="caladruri">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="parameters">
        <xs:complexType>
```

```
<xs:sequence>
  <xs:group ref="nsl:param-altid"/>
  <xs:group ref="nsl:param-pid"/>
  <xs:group ref="nsl:param-pref"/>
  <xs:group ref="nsl:param-type"/>
  <xs:group ref="nsl:param-mediatype"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element ref="nsl:uri"/>
</xs:sequence>
</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="nsl:param-altid"/>
            <xs:group ref="nsl:param-pid"/>
            <xs:group ref="nsl:param-pref"/>
            <xs:group ref="nsl:param-type"/>
            <xs:group ref="nsl:param-mediatype"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element ref="nsl:uri"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!-- Top-level grammar -->
<xs:group name="property">
  <xs:choice>
    <xs:element ref="nsl:adr"/>
    <xs:element ref="nsl:anniversary"/>
    <xs:element ref="nsl:bday"/>
    <xs:element ref="nsl:caladruri"/>
    <xs:element ref="nsl:caluri"/>
    <xs:element ref="nsl:categories"/>
    <xs:element ref="nsl:clientpidmap"/>
    <xs:element ref="nsl:email"/>
    <xs:element ref="nsl:fburl"/>
    <xs:element ref="nsl:fn"/>
    <xs:group ref="nsl:property-geo"/>
    <xs:element ref="nsl:impp"/>
    <xs:element ref="nsl:key"/>
```

```
<xs:element ref="nsl:kind"/>
<xs:element ref="nsl:lang"/>
<xs:element ref="nsl:logo"/>
<xs:element ref="nsl:member"/>
<xs:element ref="nsl:n"/>
<xs:element ref="nsl:nickname"/>
<xs:element ref="nsl:note"/>
<xs:element ref="nsl:org"/>
<xs:element ref="nsl:photo"/>
<xs:element ref="nsl:prodid"/>
<xs:element ref="nsl:related"/>
<xs:element ref="nsl:rev"/>
<xs:element ref="nsl:role"/>
<xs:element ref="nsl:gender"/>
<xs:element ref="nsl:sound"/>
<xs:element ref="nsl:source"/>
<xs:element ref="nsl:tel"/>
<xs:element ref="nsl:title"/>
<xs:group ref="nsl:property-tz"/>
<xs:element ref="nsl:uid"/>
<xs:element ref="nsl:url"/>
</xs:choice>
</xs:group>

<xs:element name="vcards">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" ref="nsl:vcard"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:complexType name="vcardType">
  <xs:complexContent>
    <xs:restriction base="xs:anyType">
      <xs:choice maxOccurs="unbounded">
        <xs:group ref="nsl:property"/>
        <xs:element ref="nsl:group"/>
      </xs:choice>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:element name="vcard" type="nsl:vcardType"/>

<xs:element name="group">
  <xs:complexType>
```

```
<xs:group minOccurs="0" maxOccurs="unbounded"
  ref="ns1:property"/>
  <xs:attribute name="name" use="required"/>
</xs:complexType>
</xs:element>
</xs:schema>
```

Authors' Addresses

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

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

Hannes Tschofenig
Nokia Solutions and Networks
Linnoitustie 6
Espoo 02600
Finland

Phone: +358 (50) 4871445
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>

Randall Gellens
Qualcomm Technologies, Inc.
5775 Morehouse Drive
San Diego, CA 92121
US

Email: rg+ietf@qti.qualcomm.com

James Winterbottom
AU

Email: a.james.winterbottom@gmail.com

ECRIT
Internet-Draft
Intended status: Standards Track
Expires: January 16, 2014

B. Rosen
NeuStar, Inc.
H. Schulzrinne
Columbia U.
H. Tschofenig
Nokia Siemens Networks
July 15, 2013

Data-Only Emergency Calls
draft-ietf-ecrit-data-only-ea-06.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 a 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 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.

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	6
4.3. Sending a Data-Only Emergency Call	7
5. Error Handling	8
5.1. 425 (Bad Alert Message) Response Code	8
5.2. The AlertMsg-Error Header Field	8
6. Updates to the CAP Message	10
7. Example	10
8. Security Considerations	14
9. IANA Considerations	16
9.1. Registration of the 'application/emergencyCall.cap+xml' MIME type	16
9.2. IANA Registration of Additional Data Block	17
9.3. IANA Registration for 425 Response Code	17
9.4. IANA Registration of New AlertMsg-Error Header Field	18
9.5. IANA Registration for the SIP AlertMsg-Error Codes	18
10. Acknowledgments	19
11. References	19
11.1. Normative References	19
11.2. Informative References	20
Authors' Addresses	21

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.

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 an alert 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 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 and a PIDF-LO payload 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.

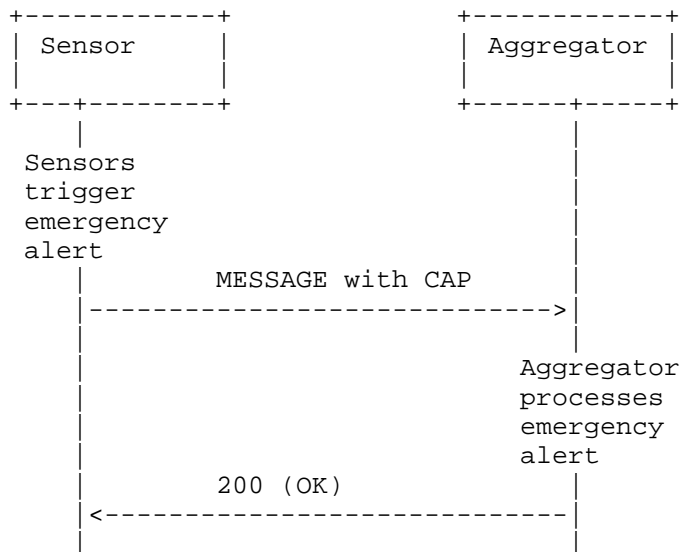


Figure 1: Targeted Emergency Alert Routing

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.

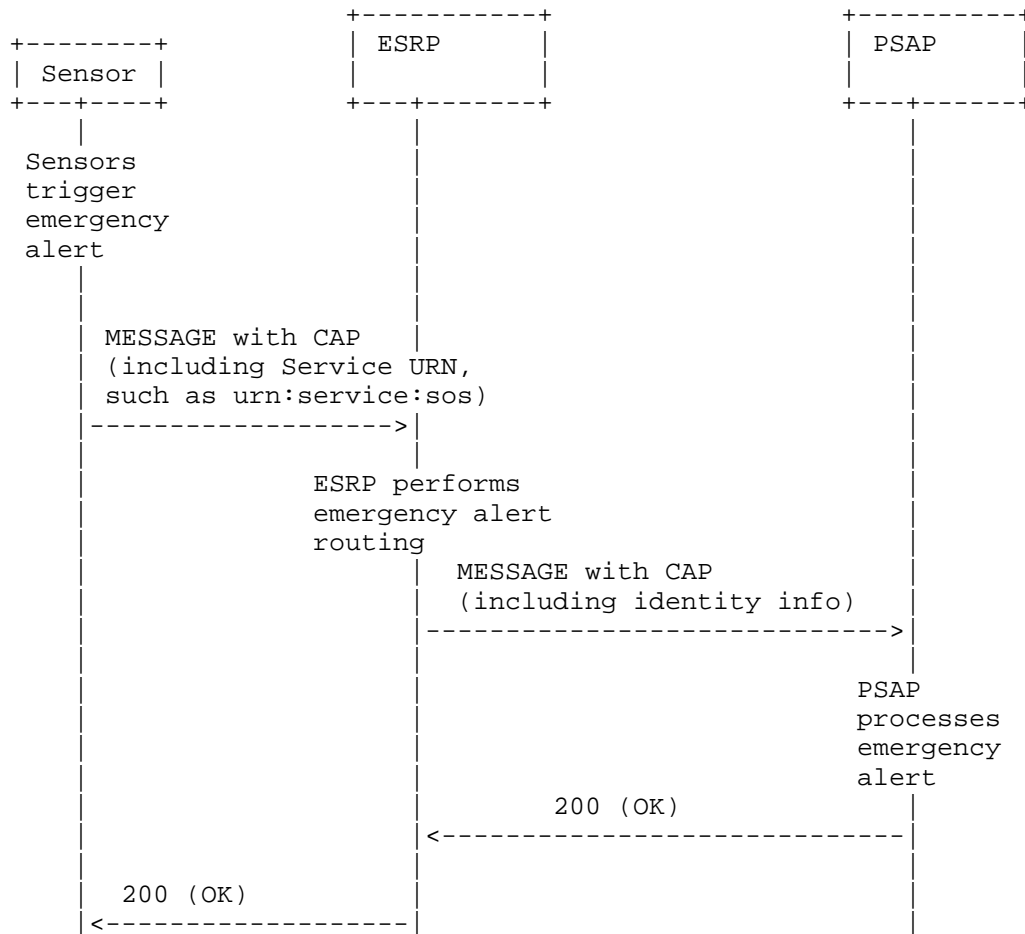




Figure 2: Location-Based Emergency Alert Routing

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 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. Alternatively, 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          = 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 send a 425 with a `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. 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:gbp="
    urn:ietf:params:xml:ns:pidf:geopriv10:basicPolicy"
  xmlns:cl="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model"
  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@example.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"
 xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
 xmlns:gpp="urn:ietf:params:xml:ns:pidf:geopriv10:basicPolicy"
 xmlns:cl="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr"

```
xmlns:gml="http://www.opengis.net/gml"
xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model"
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 4: Example Message conveying an Alert to a PSAP

8. 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.

9. IANA Considerations

9.1. 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.

9.2. 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.

9.3. 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

Registry:

Response Code	Reference
Request Failure 4xx	
425 Bad Alert Message	[this doc]

This SIP Response code is defined in Section 5.

9.4. 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

Registry:

Header Name	compact	Reference
-----	-----	-----
AlertMsg-Error		[this doc]

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

Header Field	Parameter Name	Predefined Values	Reference
-----	-----	-----	-----
AlertMsg-Error	code	yes	[this doc]

9.5. 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

Code	Default Reason Phrase	Reference
----	-----	-----
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.

10. 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.

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", March 1997.
- [cap] Jones, E. and A. Botterell, "Common Alerting Protocol v. 1.1 ", October 2005.
- [RFC2392] Levinson, E., "Content-ID and Message-ID Uniform Resource Locators", RFC 2392, August 1998.
- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.
- [RFC3428] Campbell, B., Rosenberg, J., Schulzrinne, H., Huitema, C., and D. Gurle, "Session Initiation Protocol (SIP) Extension for Instant Messaging", RFC 3428, December 2002.
- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.
- [RFC3903] Niemi, A., "Session Initiation Protocol (SIP) Extension for Event State Publication", RFC 3903, October 2004.

- [RFC3023] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", RFC 3023, January 2001.
- [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.
- [RFC6442] Polk, J., Rosen, B., and J. Peterson, "Location Conveyance for the Session Initiation Protocol", RFC 6442, December 2011.
- [RFC6665] Roach, A., "SIP-Specific Event Notification", RFC 6665, July 2012.
- [RFC6881] Rosen, B. and J. Polk, "Best Current Practice for Communications Services in Support of Emergency Calling", BCP 181, RFC 6881, March 2013.
- [I-D.ietf-ecrit-additional-data]
Rosen, B., Tschofenig, H., Marshall, R., Randy, R., and J. Winterbottom, "Additional Data related to an Emergency Call", draft-ietf-ecrit-additional-data-10 (work in progress), July 2013.
- [I-D.rosen-ecrit-addldata-subnot]
Rosen, B., "Updating Additional Data related to an Emergency Call using Subscribe/ Notify", draft-rosen-ecrit-addldata-subnot-00 (work in progress), July 2013.

11.2. Informative References

- [I-D.ietf-ecrit-trustworthy-location]
Tschofenig, H., Schulzrinne, H., and B. Aboba, "Trustworthy Location", draft-ietf-ecrit-trustworthy-location-06 (work in progress), July 2013.
- [RFC4474] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, August 2006.
- [RFC3325] Jennings, C., Peterson, J., and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", RFC 3325, November 2002.
- [RFC6443] Rosen, B., Schulzrinne, H., Polk, J., and A. Newton, "Framework for Emergency Calling Using Internet Multimedia", RFC 6443, December 2011.

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>

ECRIT
Internet-Draft
Intended status: Standards Track
Expires: January 17, 2014

R. Marshall
J. Martin
TCS
B. Rosen
Neustar
July 16, 2013

A LoST extension to support return of complete and similar location info
draft-marshall-ecrit-similar-location-02

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 a valid or invalid location is 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 location element set for a valid response, and one or more suggested sets of civic location information for invalid LoST responses. These two types of civic addresses are referred to as either "complete" or "similar" locations, and are included as compilation of ca type xml elements within the existing response 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 17, 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.

Table of Contents

1. Introduction	2
2. Terminology	3
3. Overview of Returned Location Information	4
4. Returned Location Information	6
5. Complete Location returned for Valid response	6
6. Similar Location returned for Invalid Response	8
7. Relax NG schema	10
8. Security Considerations	12
9. IANA Considerations	12
10. Acknowledgements	12
11. References	12
11.1. Normative References	12
11.2. Informative References	12
Authors' Addresses	12

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 element as a part of the whole address, the mechanism is insufficient in providing either the complete set of address elements that the LoST server contains, or of providing alternate suggestions (hints) as to which civic address is intended.

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 location information for those valid or invalid cases.

This enhancement to the validation feature within LoST is required in order to ensure a high level of address matching, to overcome user and system input errors, and to support the usefulness of location-based systems in general.

The structure of this document includes terminology, Section 2, followed by a discussion of the basic elements involved in location validation. These 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:

Address: The term Address is used interchangeably with the concept of Civic Location.

Invalid: The result of the attempt to match an individual input data as part of a larger set of data that has already been successfully matched.

Invalid Civic Element: An unmatched result of an individual civic location element as part of a broader set of elements that make up a civic location.

Invalid Civic Location: An unmatched result of an input civic location, when taken as a whole, based on one or more individual unmatched civic address elements.

Complete Location: An expanded civic location that includes additional address elements in addition to the existing validated civic elements provided.

Similar Location: A suggested civic location that is comparatively close to the civic location which was input, but which had one or more invalid element.

Returned Location Information: A set of standard civic location 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 location, its goal is to take the set of elements in the location information in the request, and find a unique location in its database that matches the information in the request. Uniqueness may not require values for all possible elements in the civic address that the database may hold. Further, the input location information may not represent the form of location the users of the LoST service prefer to have. As an example, there are LoST 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 may 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 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 subsequent emergency call.

Since a LoST server often contains more data than what is often included within a `findService` request, it is expected that this additional location information could be returned within response messages that may be both valid and invalid. For valid responses, where a LoST server contains additional location information relating to that civic address, the `findServiceResponse` message can return additional location information along with the original validated elements in order to form a complete civic location.

On the other hand, for an invalid LoST response that contains address elements returned with one or more of them marked as invalid, and constituting an invalid location, this document introduces the idea of reusing this same mechanism, but for a different purpose - to supply similar location information - again, information that is contained within the LoST server, but is provided as a complete "similar" civic location put forward as a suggested alternative address that is also a valid location.

In valid location responses, when a LoST server returns a response to a `findService` request that contains a set of CAType elements

considered valid, the location information in the `findServiceResponse` is extended to include additional location information specific for that location. As an example, the query may contain a HNO (house number), RD (road name) and A3 (city) but may not contain A1, A2, PC (Postal Code) CAtypes. The RD and PC elements may be sufficient to locate the address specified in the request and thus be considered valid. Yet, downstream entities may find it helpful to have the additional A1, A2, and PC location elements that exist, and so the mechanism described here supports their inclusion. 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 it can include a `completeLocation` element within the `findServiceResponse` message, representing a "complete" civic location.

input address: 6000 15th Ave NW Seattle

completed address: 6000 15th Ave NW Seattle, WA 98105 US

When invalid location responses are received, the same mechanism works as follows: when a LoST server returns a response to a `findService` request that contains a set of CAtype elements with one or more that are tagged as invalid, the location information in the `findServiceResponse` is extended to include additional location information specific for that location. Differing results in the same data used in the above example, where the RD and PC elements are not sufficient to locate a unique address leads to an "invalid" result. This is the case, despite the fact that the LoST server typically contains additional location elements which could have resulted in a uniquely identifiable location if additional data had been supplied in 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 it can include one or more `similarLocation` elements within the `findServiceResponse` message representing "similar" civic locations.

To show this, suppose that a similar address as above is inserted within a Lost `findService` request:

input address: 6000 15th Ave Seattle, WA.

Different from the above case, this time we make the assumption that the address is deemed "invalid" by the LoST server because there is no plain "15th Ave" in the city of Seattle with a house number that matches 6000. However 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 elements can be provided either as part of a complete civic location (CCL) or a similar civic location (SCL). The inclusion of either CCL or SCL 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 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.

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

```
<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">

      <A3>Seattle</A3>
      <A6>15th</A6>
      <STS>Ave</STS>
      <POD>NW</POD>
      <HNO>6000</HNO>

    </civicAddress>
  </location>

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

```
</findService>

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

<findServiceResponse >
  xmlns="urn:ietf:params:xml:ns:lost1"
  xmlns:rli="urn:ietf:params:xml:ns:lost-rli1">
    xmlns:ca="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr">

    <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:A6 ca:STS ca:POD ca:HNO</valid>
      <invalid></invalid>
      <unchecked></unchecked>

      <rli:completeLocation> <!-- completed address -->
        <ca:civicAddress>
          <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>NW</ca:POD>
          <ca:HNO>6000</ca:HNO>
          <ca:PC>98106</ca:PC>
          <ca:PCN>SEATTLE</ca:PCN>
        </ca:civicAddress>

      </rli:completeLocation>

    </locationValidation>

    <path>
```

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

  <locationUsed id="587cd3880"/>

</findServiceResponse>

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

6. Similar Location returned for Invalid Response

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

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

<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>
      <A1>WA</A1>
      <A3>Seattle</A3>
      <A6>15th Ave</A6>
      <HNO>6000</HNO>

    </civicAddress>
  </location>

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

</findService>

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

```
<findServiceResponse>
  xmlns="urn:ietf:params:xml:ns:lost1"
  xmlns:rli="urn:ietf:params:xml:ns:lost-rli1">
  xmlns:ca="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr">

  <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:country ca:A1 ca:A3</valid>
    <invalid>ca:A6</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:POD>NW</ca:POD>
        <ca:HNO>6000</ca:HNO>
        <ca:PC>98106</ca:PC>
        <ca:PCN>SEATTLE</ca:PCN>
      </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:POD>NE</ca:POD>
        <ca:HNO>6000</ca:HNO>
        <ca:PC>98105</ca:PC>
        <ca:PCN>SEATTLE</ca:PCN>
      </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 [TBA].

```
namespace a = "http://relaxng.org/ns/compatibility/annotations/1.0"
default namespace ns1 = "urn:ietf:params:xml:ns:lost-rl11"
```

```
##
##      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.
##
div {

    ##
    ##      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. In the case where the input location is valid, the requester still may not actually understand where that location is. For valid location use cases, this extension returns more location information than the requester may have had which, in turn, may reveal more about the location. While this may be very desirable when, for example, supporting an emergency call, it may not be as desirable for other services. The LoST server implementation should consider the risk of releasing more detail versus the value in doing so. Generally, we do not believe this is 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.

9. IANA Considerations

10. Acknowledgements

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

11.2. Informative References

[RFC5222] Hardie, T., Newton, A., Schulzrinne, H., and H. Tschofenig, "LoST: A Location-to-Service Translation Protocol", RFC 5222, August 2008.

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

ECRIT
Internet-Draft
Intended status: Standards Track
Expires: January 09, 2014

B. Rosen
NeuStar
July 08, 2013

Updating Additional Data related to an Emergency Call using Subscribe/
Notify
draft-rosen-ecrit-addldata-subnot-00.txt

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.

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 09, 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.

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 define 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: <UpdateSubscribeURII>

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]

Rosen, B., Tschofenig, H., Marshall, R., and R. Randy,
"Additional Data related to an Emergency Call", draft-
ietf-ecrit-additional-data-09 (work in progress), May
2013.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3023] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", RFC 3023, January 2001.
- [RFC4288] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", RFC 4288, December 2005.
- [RFC6446] Niemi, A., Kiss, K., and S. Loreto, "Session Initiation Protocol (SIP) Event Notification Extension for Notification Rate Control", RFC 6446, January 2012.
- [RFC6655] McGrew, D. and D. Bailey, "AES-CCM Cipher Suites for Transport Layer Security (TLS)", RFC 6655, July 2012.

Author's Address

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

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