ECRIT
Internet-Draft
Intended status: Standards Track
Expires: January 5, 2016

R. Gellens
Qualcomm Technologies, Inc.
B. Rosen
NeuStar
H. Tschofenig

R. Marshall
TeleCommunication Systems, Inc.
J. Winterbottom

July 4, 2015

Additional Data Related to an Emergency Call
draft-ietf-ecrit-additional-data-32.txt

Abstract

When an emergency call is sent to a Public Safety Answering Point (PSAP), the originating device, the access network provider to which the device is connected, and all service providers in the path of the call have information about the call, the caller or the location which is helpful for the PSAP to have in handling the emergency. This document describes data structures and mechanisms to convey such data to the PSAP. The intent is that every emergency call carry the information described here using the mechanisms described here.

The mechanisms permit the data to be conveyed by reference (as an external resource) or by value (within the body of a SIP message or a location object). This follows the tradition of prior emergency services standardization work where data can be conveyed by value within the call signaling (i.e., in the body of the SIP message) or 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 January 5, 2016.

Table of Contents

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 request for emergency services, location information, and meta-data about the sources of the data. In addition, the device, the access network provider, and any service provider in the call path has even more information that is useful for a PSAP when handling an emergency.

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 is 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. The intent is that every emergency call carry the information described here using the mechanisms described here.

This document defines three categories of this additional data that can be transmitted with an emergency call:

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

Data Associated with a Call: While some information is carried in the call setup procedure itself (as part of the SIP headers as well as in the body of the SIP message), there is additional data known by the device making the call, the access network to which the device is connected, and service providers along the path of the call. This information includes service provider contact information, subscriber identity and contact information, the type of service the service provider and the access network provide, what type of device is being used, etc. Some data is broadly applicable, while other data is dependent on the type of device or service. For example, a medical monitoring device may have sensor data. The data structures defined in this document (Data Provider Information, Device Information, and Owner/Subscriber Information) all fall into the category of "Data Associated with a Call". Note that the Owner/Subscriber Information includes the subscriber's vCard, which may contain personal information such as birthday, anniversary, etc., but the data block itself is still considered to be about the call, not the caller.

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

While this document defines data structures only within the category of Data Associated with a Call, by establishing the overall framework of Additional Data, along with general mechanisms for transport of such data, extension points and procedures for future extensions, it

minimizes the work needed to carry data in the other categories. Other specifications may make use of the facilities provided here.

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

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

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

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

2. Terminology

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

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

involved in providing emergency services. This includes PSAPs, dispatch, police, fire, emergency medical, other responders, and other similar agencies.

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

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

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

'Optional': means it MAY be present.

vCard [RFC6350] 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 reason, an XML-based encoding of the information elements defined in the vCard specification has been defined and the name of that specification is xCard [RFC6351]. Since the term vCard is more familiar to most readers, we use the terms xCard and vCard interchangeably.

3. Document Scope

The scope of this document is explicitly limited to emergency calls. The data structures defined here are not appropriate to be conveyed with non-emergency calls because they carry sensitive and private data. However, in certain private-use situations where the endpoints have a preexisting relationship and privacy issues are addressed within the relationship or do not apply because of it, the mechanisms and data structures defined here MAY be used.

4. Data Structures

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

'Data Provider': This block supplies name and contact information for the entity that created the data. Section 4.1 provides the details.

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

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

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

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

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

Each block is added to the Additional Data Blocks Registry created in Section 10.1.9 and categorized as providing data about the caller. New blocks added to the registry in the future MUST also be categorized per the description of the three categories in Section 1. See Section 4.3.7 and Section 4.3.8 for additional considerations when adding new blocks or types of data.

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

4.1. Data Provider Information

This block is intended to be supplied by any service provider in the path of the call, or the access network provider, and the device. It includes identification and contact information. This block MUST be supplied by any entity that provides any other block; it SHOULD be supplied by every service provider in the call path and by the access network provider if those entities do not add any other blocks. Devices SHOULD use this block to provide identifying information. The MIME subtype is "application/EmergencyCallData.ProviderInfo+xml". An access network provider SHOULD provide this block either by value or by reference in the <provided-by> element of a PIDF-LO

4.1.1. Data Provider String

Data Element: Data Provider String

Use: Conditional

XML Element: <DataProviderString>

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

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

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

4.1.2. Data Provider ID

Data Element: Data Provider ID

Use: Conditional. Optional for blocks supplied by the originating device, mandatory otherwise. This data MUST be provided by all entities other than the originating device in order to uniquely identify the service provider or access provider.

XML Element: <ProviderID>

Description: A jurisdiction-specific code for, or the fully-qualified domain name of, the access network provider or service provider shown in the <DataProvidedBy> element that created the structure. NOTE: The value SHOULD be assigned by an organization

appropriate for the jurisdiction. In the U.S., if the provider is registered with NENA, the provider's NENA Company ID MUST appear here. Additional information can be found at NENA Company Identifier Program [1] or NENA Company ID [2]. The NENA Company ID MUST be in the form of a URI in the following format: urn:nena:companyid:<NENA Company ID>. If the organization does not have an identifier registered with a jurisdiction-specific emergency services registrar (such as NENA), then the value MAY be the fully-qualified domain name of the service provider or access provider. The device MAY use its IP address or fully-qualified domain name (and set the "Data Provider ID Series" element to "domain").

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

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

4.1.3. Data Provider ID Series

Data Element: Data Provider ID Series

Use: Conditional. Optional for blocks supplied by the originating device, mandatory otherwise.

XML Element: <ProviderIDSeries>

Description: Identifies the issuer of the <ProviderID>. The Provider ID Series Registry created in Section 10.1.1 initially contains the entries shown in Figure 1.

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

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

Name	Source	URL
NENA	National Emergency Number Association	http://www.nena.org
EENA	European Emergency Number Association	http://www.eena.org
domain	(The ID is a fully-qualified domain name)	(not applicable)

Figure 1: Provider ID Series Registry

4.1.4. Type of Data Provider

Data Element: Type of Data Provider

Use: Required.

XML Element: <TypeOfProvider>

Description: Identifies the type of data provider supplying the data. The registry containing all valid values is created in Section 10.1.5 and the initial set of values is shown in Figure 2.

Reason for Need: Identifies the category of data provider.

How Used by Call Taker: This information may be helpful when deciding whom to contact when further information is needed.

Token	Description
Client	Originating client/device
Access Network Provider	Access network service provider
Telecom Provider	Telecom service provider (including over-the-top VoIP services)
Telematics Provider	A sensor-based service provider, especially vehicle-based
Language Translation Provider	A spoken language translation service
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 service, e.g., video relay for sign language interpreting
Other	Any other type of service provider

Figure 2: Type of Data Provider Registry

4.1.5. Data Provider Contact URI

Data Element: Data Provider Contact URI

Use: Required

XML Element: <ContactURI>

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

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

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

4.1.6. Data Provider Languages(s) Supported

Data Element: Data Provider Language(s) supported

Use: Required.

XML Element: <Language>

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

(Note that this field informs the PSAP of the language(s) used by the data provider. If the PSAP needs to contact the data provider, it can be helpful to know in advance the language(s) used by the data provider. If the PSAP uses a communication protocol to reach the data provider, that protocol may have language facilities of its own (such as the 'language' media feature tag, defined in RFC 3840 [RFC3840] and the more extensive language negotiation mechanism proposed with [I-D.gellens-slim-negotiating-human-language]), and if so, those are independent of this field.)

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

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

4.1.7. xCard of Data Provider

Data Element: xCard of Data Provider

Use: Optional

XML Element: <DataProviderContact>

Description: Per [RFC6351] the xCard structure is represented within a <vcard> element. Although multiple <vcard> elements may be contained in a structure only one <vcard> element SHOULD be provided. If more than one appears, the first SHOULD be used. There are many fields in the xCard and the creator of the data structure is encouraged to provide all available information. N, ORG, ADR, TEL, EMAIL are suggested at a minimum. N SHOULD contain the name of the support group or device owner as appropriate. If more than one TEL property is provided, a parameter from the vCard Property Value registry MUST be specified for each TEL. For encoding of the vCard this specification uses the XML-based encoding specified in [RFC6351], referred to in this document as "xCard".

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

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

4.1.8. Subcontractor Principal

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

Data Element: Subcontractor Principal

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

XML Element: <SubcontractorPrincipal>

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

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

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

4.1.9. Subcontractor Priority

Data Element: Subcontractor Priority

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

XML Element: <SubcontractorPriority>

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

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

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

4.1.10. ProviderInfo Example

```
<?xml version="1.0" encoding="UTF-8"?>
<ad:EmergencyCallData.ProviderInfo
    xmlns:ad="urn:ietf:params:xml:ns:EmergencyCallData:ProviderInfo">
    <ad:DataProviderReference>string0987654321@example.org
    </ad:DataProviderReference>
    <ad:DataProviderString>Example VoIP Provider
      </ad:DataProviderString>
    <ad:ProviderID>urn:nena:companyid:ID123</ad:ProviderID>
    <ad:ProviderIDSeries>NENA</ad:ProviderIDSeries>
    <ad:TypeOfProvider>Telecom Provider</ad:TypeOfProvider>
    <ad:ContactURI>tel:+1-201-555-0123</ad:ContactURI>
    <ad:Language>en</ad:Language>
    <ad:DataProviderContact
        xmlns="urn:ietf:params:xml:ns:vcard-4.0">
            <vcard>
                <fn><text>Hannes Tschofenig</text></fn>
                <n>
                    <surname>Hannes</surname>
                    <given>Tschofenig</given>
                    <additional/>
                    <prefix/>
                    <suffix>Dipl. Ing.</suffix>
                </n>
                <bday><date>--0203</date></bday>
                <anniversary>
```

```
    <date-time>20090808T1430-0500</date-time>
</anniversary>
<gender><sex>M</sex></gender>
<lang>
    <parameters><pref><integer>1</integer></pref>
    </parameters>
    <language-tag>de</language-tag>
</lang>
<lang>
    <parameters><pref><integer>2</integer></pref>
    </parameters>
    <language-tag>en</language-tag>
</lang>
<org>
    <parameters><type><text>work</text></type>
    </parameters>
    <text>Example VoIP Provider</text>
</org>
<adr>
    <parameters>
        <type><text>work</text></type>
        <label><text>Hannes Tschofenig
            Linnoitustie 6
            Espoo , Finland
            02600</text></label>
    </parameters>
    <pobox/>
    <ext/>
    <street>Linnoitustie 6</street>
    <locality>Espoo</locality>
    <region>Uusimaa</region>
    <code>02600</code>
    <country>Finland</country>
</adr>
<tel>
    <parameters>
        <type>
            <text>work</text>
            <text>voice</text>
        </type>
    </parameters>
    <uri>tel:+358 50 4871445</uri>
</tel>
<email>
    <parameters><type><text>work</text></type>
    </parameters>
    <text>hannes.tschofenig@nsn.com</text>
</email>
```

```
                        <geo>
                            <parameters><type><text>work</text></type>
                            </parameters>
                            <uri>geo:60.210796,24.812924</uri>
                        </geo>
                        <key>
                            <parameters><type><text>home</text></type>
                            </parameters>
                            <uri>
                            http://www.tschofenig.priv.at/key.asc
                            </uri>
                        </key>
                        <tz><text>Finland/Helsinki</text></tz>
                        <url>
                            <parameters><type><text>home</text></type>
                            </parameters>
                            <uri>http://www.tschofenig.priv.at</uri>
                        </url>
                    </vcard>
        </ad:DataProviderContact>
    </ad:EmergencyCallData.ProviderInfo>
```

Figure 3: EmergencyCallData.ProviderInfo Example.

4.2. Service Information

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

4.2.1. Service Environment

Data Element: Service Environment

Use: Conditional: Required unless the 'ServiceType' value is 'wireless'.

XML Element: <ServiceEnvironment>

Description: This element indicates whether a call is from a business or residence. Currently, the only valid entries are 'Business', 'Residence', and 'unknown', as shown in Figure 4. New values can be defined via the registry created in Section 10.1.2.

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

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

Token	Description
Business	Business service
Residence	Residential service
unknown	Type of service unknown (e.g., anonymous pre-paid service)

Figure 4: Service Environment Registry

4.2.2. Service Type

Data Element: Service Delivered by Provider to End User

Use: Required

XML Element: <ServiceType>

Description: This defines the type of service over which the call is placed (similar to the Class of Service delivered with legacy emergency calls in some some regions). The implied mobility of this service cannot be relied upon. A registry is created in Section 10.1.3. The initial set of values is shown in Figure 5. 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 in 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 legacy emergency calling systems, it is known to be valuable.

Name	Description
wireless	Wireless Telephone Service: Includes CDMA, GSM, Wi-Fi, WiMAX, LTE (but not satellite)
coin	Fixed public pay/coin telephones: Any coin or credit card operated device
one-way	One way outbound service
prison	Inmate call/service
temp	Soft dial tone/quick service/warm disconnect/suspended
MLTS-hosted	Hosted multi-line telephone system such as Centrex
MLTS-local	Local multi-line telephone system, includes all PBX, key systems, Shared Tenant Service
sensor-unattended	These are devices that generate DATA ONLY. This is a one-way information transmit without interactive media
sensor-attended	Devices that are supported by a monitoring service provider or that are capable of supporting interactive media
POTS	Wireline: Plain Old Telephone Service
VOIP	An over-the-top service that provides communication over arbitrary Internet access (fixed, nomadic, mobile)
remote	Off-premise extension
relay	A service where a human third-party agent provides additional assistance This includes sign language relay/ interpretation, telematics services that provide a human on the call, and similar services.

Figure 5: Service Delivered by Provider to End User Registry

4.2.3. Service Mobility Environment

Data Element: Service Mobility Environment

Use: Required

XML Element: <ServiceMobility>

Description: This provides the service provider's view of the mobility of the caller's device. As the service provider might not know the characteristics of the actual device or access network used, the value should be treated as advisory and not be relied upon. A registry is created in Section 10.1.4 with the initial valid entries shown in Figure 6.

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.

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

Figure 6: Service Environment Registry

4.2.4. EmergencyCallData.ServiceInfo Example

```
<?xml version="1.0" encoding="UTF-8"?>
<svc:EmergencyCallData.ServiceInfo
     xmlns:svc="urn:ietf:params:xml:ns:EmergencyCallData:ServiceInfo">
    <svc:DataProviderReference>2468.IBOC.MLTS.1359@example.org
    </svc:DataProviderReference>
    <svc:ServiceEnvironment>Business</svc:ServiceEnvironment>
    <svc:ServiceType>MLTS-hosted</svc:ServiceType>
    <svc:ServiceMobility>Fixed</svc:ServiceMobility>
</svc:EmergencyCallData.ServiceInfo>
```

Figure 7: EmergencyCallData.ServiceInfo Example.

4.3. Device Information

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

4.3.1. Device Classification

Data Element: Device Classification

Use: Optional

XML Element: <DeviceClassification>

Description: This data element defines the kind of device making the emergency call. If the device provides the data structure, the device information SHOULD be provided. If the service provider provides the structure and it knows what the device is, the service provider SHOULD provide the device information. Often the carrier does not know what the device is. It is possible to receive two Additional Data Associated with a Call data structures, one created by the device and one created by the service provider. This information describes the device, not how it is being used. This data element defines the kind of device making the emergency call. A registry is created in Section 10.1.6 with the initial set of values as shown in Figure 8.

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

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

Token	Description
cordless	Cordless handset
fixed	Fixed phone
satellite	Satellite phone
sensor-fixed	Fixed (non mobile) sensor/alarm device
desktop	Soft client on desktop PC
laptop	Soft client on laptop type device
tablet	Soft client on tablet type device
alarm-monitored	Alarm system
sensor-mobile	Mobile sensor device
aircraft	Aircraft telematics device
automobile	Automobile/cycle/off-road telematics
truck	Truck/construction telematics
farm	Farm equipment telematics
marine	Marine telematics
personal	Personal telematics device
feature-phone	Feature- (not smart-) cellular phone
smart-phone	Smart-phone cellular phone (native)
smart-phone-app	Soft client app on smart-phone
unknown-device	Soft client on unknown device type
game	Gaming console
text-only	Other text device
NA	Not Available

Figure 8: Device Classification Registry Initial Values

4.3.2. Device Manufacturer

Data Element: Device Manufacturer

Use: Optional

XML Element: <DeviceMfgr>

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

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

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

4.3.3. Device Model Number

Data Element: Device Model Number

Use: Optional

XML Element: <DeviceModelNr>

Description: Model number of the device.

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

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

4.3.4. Unique Device Identifier

Data Element: Unique Device Identifier

Use: Optional

XML Element: <UniqueDeviceID>

XML Attribute: <TypeOfDeviceID>

Description: A string that identifies the specific device (or the device's current SIM) making the call or creating an event. Note that more than one <UniqueDeviceID> may be present, to supply more than one of the identifying values.

The <TypeOfDeviceID> attribute identifies the type of device identifier. A registry is created in Section 10.1.7 with an initial set of values shown in Figure 9.

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

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

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

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)
IMSI	International Mobile Subscriber ID (GSM)
UDI	Unique Device Identifier
RFID	Radio Frequency Identification
SN	Manufacturer Serial Number

Figure 9: Registry of Device Identifier Types

4.3.5. Device/Service-Specific Additional Data Structure

Data Element: Device/service-specific additional data structure

Use: Optional

XML Element: <DeviceSpecificData>

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

Reason for Need: Provides device/service specific data that may be used by the call taker and/or responders.

How Used by Call Taker: Provide information to guide call takers to select appropriate responders, give appropriate pre-arrival instructions to callers, and advise responders of what to be prepared for. May be used by responders to guide assistance provided.

4.3.6. Device/Service Specific Additional Data Structure Type

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

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

XML Element: <DeviceSpecificType>

Description: A value from the registry defined in Section 10.1.8 to describe the type of data located at the device/service-specific additional data structure. The initial values shown in Figure 10 currently only include IEEE 1512, which is the USDoT model for traffic incidents.

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

How Used by Call Taker: This data element allows the end user (call taker 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.

Token	Description	Specification
IEEE1512	Common Incident Management Message Set (USDoT model for traffic incidents)	IEEE 1512-2006 https://standards.ieee.org/findstds/standard/1512-2006.html

Figure 10: Device/Service Data Type Registry

4.3.7. Issues with getting new types of data into use

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

The mechanisms this document describes are meant to encourage development of widely supported, common data formats for classes of devices. If all manufacturers of a class of device use the same

format, and the data can be shown to improve outcomes, then PSAPs and responders may be encouraged to upgrade their systems and train their staff to use the data. Variations, however well intentioned, are unlikely to be supported.

Implementers should consider that data from sensor-based devices in some cases may not be useful to call takers or PSAPs (and privacy or other considerations may preclude the PSAP from touching the data), but may be of use to responders. Each data item provided with the call in conformance with this document can be accessed by responders or other entities in the emergency services, whether or not the data is accessed by the PSAP.

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

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

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

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

Size: Information which may be very large may be better sent in the DeviceInfo block, rather than a new block so that implementations are not tempted to send the data by value. Conversely, data which is small may best be sent in a separate block so that it can be sent by value

Availability of a server: Providing the data via the device block requires a server be made available to retrieve the data. Providing the data via new block allows it to be sent by value (CID).

4.3.9. EmergencyCallData.DeviceInfo Example

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

Figure 11: EmergencyCallData.DeviceInfo Example.

4.4. Owner/Subscriber Information

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

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

4.4.1. Subscriber Data Privacy Indicator

Attribute: 'privacyRequested', Boolean.

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

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

jurisdiction (in keeping with the semantics of the privacy indicator provided in some legacy emergency call systems).

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

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

4.4.2. xCard for Subscriber's Data

Data Element: xCARD for Subscriber's Data

Use: Conditional. Subscriber data MUST be provided unless it is not available. Some services, such as 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>. While some data (such as <anniversary>) might not seem obviously relevant for emergency services, any data is potentially useful in some emergency circumstances.

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. While the location here is not necessarily that of caller, in some circumstances it can be helpful in locating the caller when other means have failed.

4.4.3. EmergencyCallData.SubscriberInfo Example

```
<?xml version="1.0" encoding="UTF-8"?>
<sub:EmergencyCallData.SubscriberInfo
     xmlns:sub=
         "urn:ietf:params:xml:ns:EmergencyCallData:SubscriberInfo"
     privacyRequested="false">
    <sub:DataProviderReference>FEABFECD901@example.org
    </sub:DataProviderReference>
```

```
<sub:SubscriberData xmlns="urn:ietf:params:xml:ns:vcard-4.0">
     <vcard>
            <fn><text>Simon Perreault</text></fn>
            <n>
                <surname>Perreault</surname>
                <given>Simon</given>
                <additional/>
                <prefix/>
                <suffix>ing. jr</suffix>
                <suffix>M.Sc.</suffix>
            </n>
            <bday><date>--0203</date></bday>
            <anniversary>
                <date-time>20090808T1430-0500</date-time>
            </anniversary>
            <gender><sex>M</sex></gender>
            <lang>
                <parameters><pref><integer>1</integer></pref>
                </parameters>
                <language-tag>fr</language-tag>
            </lang>
            <lang>
                <parameters><pref><integer>2</integer></pref>
                </parameters>
                <language-tag>en</language-tag>
            </lang>
            <org>
                <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>
        </sub:SubscriberData>
    </sub:EmergencyCallData.SubscriberInfo>
```

Figure 12: EmergencyCallData.SubscriberInfo Example.

4.5. Comment

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

4.5.1. Comment

Data Element: EmergencyCallData.Comment

Use: Optional

XML Element: <Comment>

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

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

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

4.5.2. EmergencyCallData.Comment Example

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

Figure 13: EmergencyCallData.Comment Example.

5. Data Transport Mechanisms

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

1. First, the ability to embed a Uniform Resource Identifier (URI) in an existing SIP header field, the Call-Info header, is defined. The URI points to the additional data structure. The Call-Info header is specified in Section 20.9 of [RFC3261]. This

document adds a new compound token starting with the value 'EmergencyCallData' for the Call-Info "purpose" parameter. If the "purpose" parameter is set to a value starting with 'EmergencyCallData', then the Call-Info header contains either an HTTPS URL pointing to an external resource or a CID (content indirection) URI that allows the data structure to be placed in the body of the SIP message. The "purpose" parameter also indicates the kind of data (by its MIME subtype) 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 the <provided-by> element of a PIDF-LO), each HTTPS URL references one block; the data is retrieved with an HTTPS GET operation, which returns the block as an object (the blocks defined here are returned as XML objects).

2. Second, the ability to embed additional data structures in the <provided-by> element of a PIDF-LO [RFC4119] is defined. In addition to service providers in the call path, the access network provider may also have similar information that can be valuable to the PSAP. When the access network provider supplies location information in the form of a PIDF-LO from a location server via a location configuration protocol, it has the ability to add the data structures defined in this document within the PIDF-LO. The data in these data structures is not specific to the location itself, but rather provides descriptive information having to do with the immediate circumstances about the provision of the location (e.g., the identity of the access network

provider, 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 can provide location information in a PIDF-LO. The <provided-by> element of the PIDF-LO is a mechanism for the access network provider to supply the information. For this reason, this document describes a namespace per [RFC4119] for inclusion in the <provided-by> element of a PIDF-LO for adding information known to the access network provider. The access network provider SHOULD provide additional data within a <provided-by> element of a PDIF-LO it returns for emergency use (e.g., if requested with a HELD "responseTime" attribute of "emergencyRouting" or "emergencyDispatch" [RFC5985]).

One or more blocks of data registered in the Emergency Call Additional Data registry, as defined in Section 10.1.9, can be included or referenced in the SIP signaling (using the Call-Info header field) or in the <provided-by> element of a PIDF-LO. For interoperability, only blocks in the registry are permitted to be sent using the mechanisms specified in this document. Since multiple entities are expected to provide sets of data, the data itself needs information describing the source. Consequently, each entity adding additional data MUST supply a "Data Provider" block. All other blocks are optional, but each entity SHOULD supply all blocks where it has at least some of the information in the block.

5.1. Transmitting Blocks using the Call-Info Header

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

```
Call-Info:  https://www.example.com/23sedde3;
    purpose="EmergencyCallData.ProviderInfo"
```

A Call-info header with a purpose value starting with 'EmergencyCallData' only has meaning in the context of an emergency

call (as ascertained by the presence of an emergency service URN in a Request-URI header of a SIP message), test emergency calls (using an appropriate service URN), and some private-use calls where the endpoints have a preexisting relationship and privacy concerns do not apply because of the relationship; use in other contexts is undefined and is likely to unnecessarily expose confidential data.

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

The data may also be supplied by value in any SIP request or response method that is permitted to contain a body (i.e., not a BYE request). In this case, Content Indirection (CID) [RFC2392] is used, with the CID URL referencing the MIME body part containing the data. Note that [RFC3261] forbids proxies from altering message bodies, so entities in the call path that add blocks by value need to do so using an appropriate SIP entity (e.g., a back-to-back user agent).

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

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

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

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

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

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

The <EmergencyCallDataReference> element transmits one data block; multiple data blocks may be transmitted by using multiple <EmergencyCallDataReference> elements. Multiple <EmergencyCallDataReference> elements MAY be included as child elements inside the <provided-by> element.

The following is a simplified example:

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

        <EmergencyCallDataReference
                purpose="EmergencyCallData.ProviderInfo"
                ref="https://example.com/ref3" />

        <EmergencyCallDataReference
                purpose="EmergencyCallData.Comment"
                ref="https://example.com/ref4" />
</provided-by>
```

Example <provided-by> by Reference

For an example in context, Figure 18 shows a PIDF-LO example with an <EmergencyCallDataReference> element pointing to an EmergencyCallData.ServiceInfo data block with the URL in the 'ref' attribute and the purpose attribute set to "EmergencyCallData.ServiceInfo".

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

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

The <EmergencyCallDataValue> element is used to transmit one or more additional data blocks by value within a <provided-by> element of a PIDF-LO. Each block being transmitted is placed (as a child element)

inside the <EmergencyCallDataValue> element. (The same XML structure as would be contained in the corresponding MIME type body part is placed inside the <EmergencyCallDataValue> element.) Multiple <EmergencyCallDataValue> elements MAY be included as child elements in the <provided-by> element.

The following is a simplified example:

```
<provided-by

        <EmergencyCallDataValue>

          <EmergencyCallData.ProviderInfo
             xmlns=
             "urn:ietf:params:xml:ns:EmergencyCallData:ProviderInfo">
             <DataProviderReference>flurbit735@es.example.com
               </DataProviderReference>
             <DataProviderString>Access Network Examples, Inc
               </DataProviderString>
             <ProviderID>urn:nena:companyid:Test</ProviderID>
             <ProviderIDSeries>NENA</ProviderIDSeries>
             <TypeOfProvider>Access Network Provider
               </TypeOfProvider>
             <ContactURI>tel:+1-555-555-0897</ContactURI>
             <Language>en</Language>
           </EmergencyCallData.ProviderInfo>

           <EmergencyCallData.Comment
              xmlns=
              "urn:ietf:params:xml:ns:EmergencyCallData:Comment">
              <DataProviderReference>flurbit735@es.example.com
                </DataProviderReference>
              <Comment xml:lang="en">This is an example text.
                </Comment>
           </EmergencyCallData.Comment>

        </EmergencyCallDataValue>

</provided-by>
```

Example <provided-by> by Value

For an example in context, Figure 18 shows a PIDF-LO example that contains a <provided-by> element with the <EmergencyCallData.ProviderInfo> and the <EmergencyCallData.Comment> elements as child elements of an <EmergencyCallDataValue> element.

5.4. The Content-Disposition Parameter

RFC 5621 [RFC5621] discusses the handling of message bodies in SIP. It updates and clarifies handling originally defined in RFC 3261 [RFC3261] based on implementation experience. While RFC 3261 did not mandate support for 'multipart' message bodies, 'multipart/mixed' MIME bodies are used by many extensions (including this document) today. For example, adding a PIDF-LO, SDP, and additional data in body of a SIP message requires a 'multipart' message body.

RFC 3204 [RFC3204] and RFC 3459 [RFC3459] define the 'handling' parameter for the Content-Disposition header field. These RFCs describe how a UAS reacts if it receives a message body whose content type or disposition type it does not understand. If the 'handling' parameter has the value "optional", the UAS ignores the message body. If the 'handling' parameter has the value "required", the UAS returns a 415 (Unsupported Media Type) response. The 'by-reference' disposition type allows a SIP message to contain a reference to the body part, and the SIP UA processes the body part according to the reference. This is the case for the Call-info header containing a Content Indirection (CID) URL.

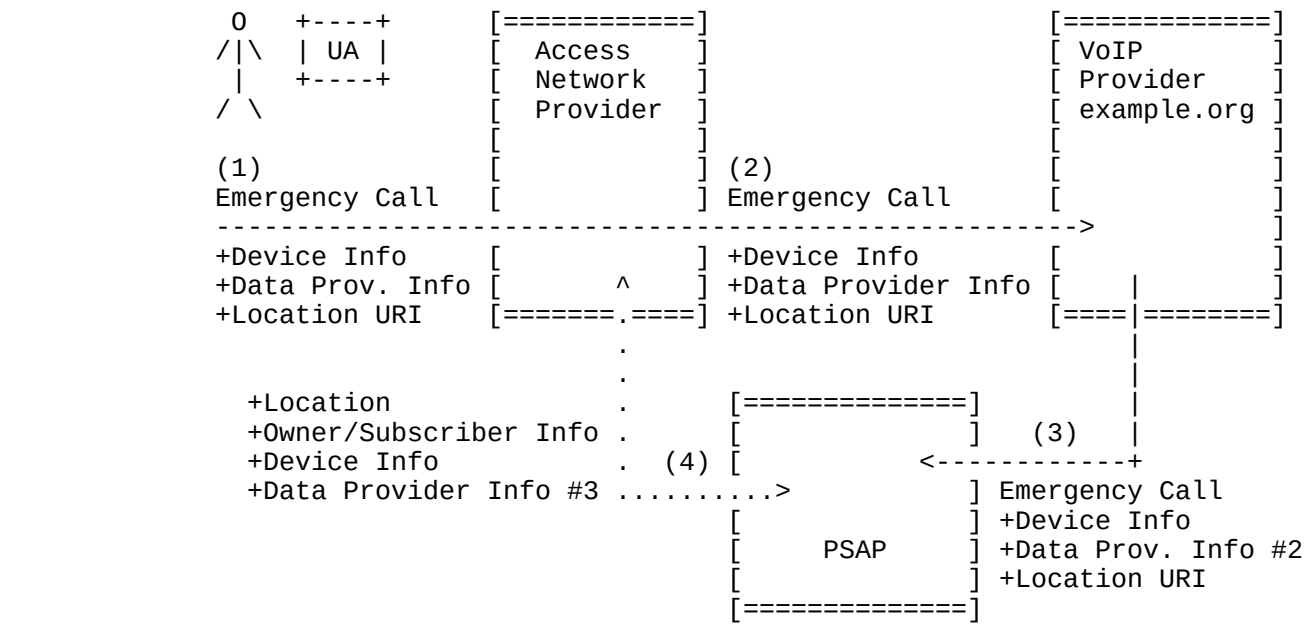
As an example, a SIP message indicates the Content-Disposition parameter in the body of the SIP message as shown in Figure 14.

```
Content-Type: application/sdp

...Omit Content-Disposition here; defaults are ok
...SDP goes in here

--boundary1

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

...PIDF-LO goes in here

--boundary1--

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

...Data provider information data goes in here

--boundary1--
```

Figure 14: Example for use of the Content-Disposition Parameter in SIP

6. Examples

This section illustrates a longer and more complex example, as shown in Figure 15. In this example additional data is added by the end device, included by the VoIP provider, and provided by the access network provider (via the PIDF-LO).



Legend:

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

Figure 15: Additional Data Example Flow

The example scenario starts with the end device itself adding device information, owner/subscriber information, a location URI, and data provider information to the outgoing emergency call setup message (see step #1 in Figure 15). The SIP INVITE example is shown in Figure 16.

```
INVITE urn:service:sos SIP/2.0
Via: SIPS/2.0/TLS server.example.com;branch=z9hG4bK74bf9
Max-Forwards: 70
To: <urn:service:sos>
From: Hannes Tschofenig <sips:hannes@example.com>;tag=9fxced76sl
Call-ID: 3848276298220188511@example.com
Call-Info: <http://wwww.example.com/hannes/photo.jpg>
               ;purpose=icon,
  <http://www.example.com/hannes/> ;purpose=info,
  <cid:1234567890@atlanta.example.com>
      ;purpose=EmergencyCallData.ProviderInfo,
```

```
        <cid:0123456789@atlanta.example.com>
            ;purpose=EmergencyCallData.DeviceInfo
      Geolocation: <https://ls.example.net:9768/357yc6s64ceyoiuy5ax3o>
      Geolocation-Routing: yes
      Accept: application/sdp, application/pidf+xml,
         application/EmergencyCallData.ProviderInfo+xml
      CSeq: 31862 INVITE
      Contact: <sips:hannes@example.com>
      Content-Type: multipart/mixed; boundary=boundary1

      Content-Length: ...

      --boundary1

      Content-Type: application/sdp

      ...SDP goes here

      --boundary1--

      Content-Type: application/EmergencyCallData.DeviceInfo+xml
      Content-ID: <0123456789@atlanta.example.com>
      Content-Disposition: by-reference;handling=optional
   <?xml version="1.0" encoding="UTF-8"?>

   <dev:EmergencyCallData.DeviceInfo
        xmlns:dev="urn:ietf:params:xml:ns:EmergencyCallData:DeviceInfo">
       <dev:DataProviderReference>d4b3072df09876543@[93.184.216.119]
       </dev:DataProviderReference>
       <dev:DeviceClassification>laptop</dev:DeviceClassification>
       <dev:UniqueDeviceID
           TypeOfDeviceID="MAC">00-0d-4b-30-72-df</dev:UniqueDeviceID>
   </dev:EmergencyCallData.DeviceInfo>


      --boundary1--

      Content-Type: application/EmergencyCallData.ProviderInfo+xml
      Content-ID: <1234567890@atlanta.example.com>
      Content-Disposition: by-reference;handling=optional
   <?xml version="1.0" encoding="UTF-8"?>
   <pi:EmergencyCallData.ProviderInfo
       xmlns:pi="urn:ietf:params:xml:ns:EmergencyCallData:ProviderInfo">
         <pi:DataProviderReference>d4b3072df09876543@[93.184.216.119]
         </pi:DataProviderReference>
         <pi:DataProviderString>Hannes Tschofenig
         </pi:DataProviderString>
       <pi:TypeOfProvider>Client</pi:TypeOfProvider>
```

```
<pi:ContactURI>tel:+1-555-555-0123</pi:ContactURI>
<pi:Language>en</pi:Language>
<pi:DataProviderContact
    xmlns="urn:ietf:params:xml:ns:vcard-4.0">
            <vcard>
                <fn><text>Hannes Tschofenig</text></fn>
                <n>
                    <surname>Hannes</surname>
                    <given>Tschofenig</given>
                    <additional/>
                    <prefix/>
                    <suffix>Dipl. Ing.</suffix>
                </n>
                <bday><date>--0203</date></bday>
                <anniversary>
                    <date-time>20090808T1430-0500</date-time>
                </anniversary>
                <gender><sex>M</sex></gender>
                <lang>
                    <parameters><pref><integer>1</integer></pref>
                    </parameters>
                    <language-tag>de</language-tag>
                </lang>
                <lang>
                    <parameters><pref><integer>2</integer></pref>
                    </parameters>
                    <language-tag>en</language-tag>
                </lang>
                <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>
                <adr>
                    <parameters>
                        <type><text>home</text></type>
                        <label><text>Hannes Tschofenig
```

```
                c/o Hotel DuPont
                42 W 11th St
                Wilmington, DE 19801
                USA</text></label>
        </parameters>
        <pobox/>
        <ext/>
        <street>42 W 11th St</street>
        <locality>Wilmington</locality>
        <region>DE</region>
        <code>19801</code>
        <country>USA</country>
    </adr>
    <tel>
        <parameters>
            <type>
                <text>work</text>
                <text>voice</text>
            </type>
        </parameters>
        <uri>tel:+358 50 4871445</uri>
    </tel>
    <tel>
        <parameters>
            <type>
                <text>home</text>
                <text>voice</text>
            </type>
        </parameters>
        <uri>tel:+1 555 555 0123</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>
    <geo>
        <parameters><type><text>home</text></type>
        </parameters>
        <uri>geo:39.746537,-75.548027</uri>
    </geo>
    <key>
        <parameters>
```

```
                                    <type><text>home</text></type>
                                </parameters>
                                <uri>https://www.example.com/key.asc
                                    </uri>
                            </key>
                            <tz><text>Finland/Helsinki</text></tz>
                            <url>
                                <parameters><type><text>home</text></type>
                                    </parameters>
                                <uri>http://example.com/hannes.tschofenig
                                    </uri>
                            </url>
                        </vcard>
    </pi:DataProviderContact>
</pi:EmergencyCallData.ProviderInfo>
   --boundary1--
```

Figure 16: End Device sending SIP INVITE with Additional Data

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

The VoIP service provider receives the message and determines, based on the Service URN, that the incoming request is an emergency call. It performs typical emergency services related tasks (such as location-based routing), and adds additional data, namely service and subscriber information as well as data provider information #2, to the outgoing message. For the example we assume a VoIP service provider that deploys a back-to-back user agent allowing additional data to be included in the body of the SIP message (rather than by reference), which allows us to illustrate the use of multiple data provider info blocks. The resulting message is shown in Figure 17. The SIP INVITE is sent to the PSAP in step #3.

```
INVITE sips:psap@example.org SIP/2.0
Via: SIPS/2.0/TLS server.example.com;branch=z9hG4bK74bf9
Max-Forwards: 70
To: <urn:service:sos>
From: Hannes Tschofenig <sips:hannes@example.com>;tag=9fxced76sl
Call-ID: 3848276298220188511@example.com
```

```
     Call-Info: <http://wwww.example.com/hannes/photo.jpg>
                    ;purpose=icon,
       <http://www.example.com/hannes/> ;purpose=info,
       <cid:1234567890@atlanta.example.com>
           ;purpose=EmergencyCallData.ProviderInfo
       <cid:0123456789@atlanta.example.com>
           ;purpose=EmergencyCallData.DeviceInfo
     Call-Info: <cid:bloorpyhex@atlanta.example.com>
           ;purpose=EmergencyCallData.ServiceInfo
     Call-Info: <cid:aaabbb@atlanta.example.com>
           ;purpose=EmergencyCallData.ProviderInfo
     Geolocation: <https://ls.example.net:9768/357yc6s64ceyoiuy5ax3o>
     Geolocation-Routing: yes
     Accept: application/sdp, application/pidf+xml,
        application/EmergencyCallData.ProviderInfo+xml
     CSeq: 31862 INVITE
     Contact: <sips:hannes@example.com>
     Content-Type: multipart/mixed; boundary=boundary1

     Content-Length: ...

     --boundary1

     Content-Type: application/sdp

     ...SDP goes here

     --boundary1--

     Content-Type: application/EmergencyCallData.DeviceInfo+xml
     Content-ID: <0123456789@atlanta.example.com>
     Content-Disposition: by-reference;handling=optional
  <?xml version="1.0" encoding="UTF-8"?>

  <dev:EmergencyCallData.DeviceInfo
       xmlns:dev="urn:ietf:params:xml:ns:EmergencyCallData:DeviceInfo">
      <dev:DataProviderReference>d4b3072df09876543@[93.184.216.119]
      </dev:DataProviderReference>
      <dev:DeviceClassification>laptop</dev:DeviceClassification>
      <dev:UniqueDeviceID
          TypeOfDeviceID="MAC">00-0d-4b-30-72-df</dev:UniqueDeviceID>
  </dev:EmergencyCallData.DeviceInfo>


     --boundary1--

     Content-Type: application/EmergencyCallData.ProviderInfo+xml
     Content-ID: <1234567890@atlanta.example.com>
```

```
    Content-Disposition: by-reference;handling=optional
<?xml version="1.0" encoding="UTF-8"?>
<pi:EmergencyCallData.ProviderInfo
    xmlns:pi="urn:ietf:params:xml:ns:EmergencyCallData:ProviderInfo">
      <pi:DataProviderReference>d4b3072df09876543@[93.184.216.119]
      </pi:DataProviderReference>
      <pi:DataProviderString>Hannes Tschofenig
      </pi:DataProviderString>
    <pi:TypeOfProvider>Client</pi:TypeOfProvider>
    <pi:ContactURI>tel:+1-555-555-0123</pi:ContactURI>
    <pi:Language>en</pi:Language>
    <pi:DataProviderContact
        xmlns="urn:ietf:params:xml:ns:vcard-4.0">
            <vcard>
                <fn><text>Hannes Tschofenig</text></fn>
                <n>
                    <surname>Hannes</surname>
                    <given>Tschofenig</given>
                    <additional/>
                    <prefix/>
                    <suffix>Dipl. Ing.</suffix>
                </n>
                <bday><date>--0203</date></bday>
                <anniversary>
                    <date-time>20090808T1430-0500</date-time>
                </anniversary>
                <gender><sex>M</sex></gender>
                <lang>
                    <parameters><pref><integer>1</integer></pref>
                    </parameters>
                    <language-tag>de</language-tag>
                </lang>
                <lang>
                    <parameters><pref><integer>2</integer></pref>
                    </parameters>
                    <language-tag>en</language-tag>
                </lang>
                <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>
                <adr>
                    <parameters>
                        <type><text>home</text></type>
                        <label><text>Hannes Tschofenig
                            c/o Hotel DuPont
                            42 W 11th St
                            Wilmington, DE 19801
                            USA</text></label>
                    </parameters>
                    <pobox/>
                    <ext/>
                    <street>42 W 11th St</street>
                    <locality>Wilmington</locality>
                    <region>DE</region>
                    <code>19801</code>
                    <country>USA</country>
                </adr>
                <tel>
                    <parameters>
                        <type>
                            <text>work</text>
                            <text>voice</text>
                        </type>
                    </parameters>
                    <uri>tel:+358 50 4871445</uri>
                </tel>
                <tel>
                    <parameters>
                        <type>
                            <text>home</text>
                            <text>voice</text>
                        </type>
                    </parameters>
                    <uri>tel:+1 555 555 0123</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>
                    <geo>
                        <parameters><type><text>home</text></type>
                        </parameters>
                        <uri>geo:39.746537,-75.548027</uri>
                    </geo>
                    <key>
                        <parameters>
                                    <type><text>home</text></type>
                        </parameters>
                        <uri>https://www.example.com/key.asc
                                     </uri>
                    </key>
                    <tz><text>Finland/Helsinki</text></tz>
                    <url>
                        <parameters><type><text>home</text></type>
                            </parameters>
                        <uri>http://example.com/hannes.tschofenig
                            </uri>
                    </url>
                </vcard>
     </pi:DataProviderContact>
  </pi:EmergencyCallData.ProviderInfo>

     --boundary1--

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

     --boundary1--

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

```
      <pi:DataProviderReference>string0987654321@example.org
      </pi:DataProviderReference>
        <pi:DataProviderString>Exemplar VoIP Provider
        </pi:DataProviderString>
      <pi:ProviderID>urn:nena:companyid:ID123</pi:ProviderID>
      <pi:ProviderIDSeries>NENA</pi:ProviderIDSeries>
      <pi:TypeOfProvider>Service Provider</pi:TypeOfProvider>
      <pi:ContactURI>sip:voip-provider@example.com</pi:ContactURI>
      <pi:Language>en</pi:Language>
      <pi:DataProviderContact
          xmlns:xc="urn:ietf:params:xml:ns:vcard-4.0">
                  <vcard>
                      <fn><text>John Doe</text></fn>
                      <n>
                          <surname>John</surname>
                          <given>Doe</given>
                          <additional/>
                          <prefix/>
                          <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>en</language-tag>
                      </lang>
                      <org>
                          <parameters><type><text>work</text></type>
                          </parameters>
                          <text>Exemplar VoIP Provider</text>
                      </org>
                      <adr>
                          <parameters>
                              <type><text>work</text></type>
                              <label><text>John Doe
                                  123 Middle Street
                                  The Sticks, IA 50055</text></label>
                          </parameters>
                          <pobox/>
                          <ext/>
                          <street>123 Middle Street</street>
                          <locality>the Sticks</locality>
                          <region>IA</region>
                          <code>50055</code>
```

```
                        <country>USA</country>
                    </adr>
                    <tel>
                        <parameters>
                            <type>
                                <text>work</text>
                                <text>voice</text>
                            </type>
                        </parameters>
                        <uri>sips:john.doe@example.com</uri>
                    </tel>
                    <email>
                        <parameters><type><text>work</text></type>
                        </parameters>
                        <text>john.doe@example.com</text>
                    </email>
                    <geo>
                        <parameters><type><text>work</text></type>
                        </parameters>
                        <uri>geo:41.761838,-92.963268</uri>
                    </geo>
                    <tz><text>America/Chicago</text></tz>
                    <url>
                        <parameters><type><text>home</text></type>
                        </parameters>
                        <uri>http://www.example.com/john.doe</uri>
                    </url>
                </vcard>
     </pi:DataProviderContact>
  </pi:EmergencyCallData.ProviderInfo>
```

Figure 17: VoIP Provider sending SIP INVITE with Additional Data

Finally, the PSAP requests location information from the access network provider. The response is shown in Figure 18. Along with the location information, additional data is provided in the <provided-by> element of the PIDF-LO. This request and response is step #4.

```
<?xml version="1.0" encoding="UTF-8"?>
<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: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>US</country>
            <A1>DE</A1>
            <A3>Wilmington</A3>
            <PRD>W</PRD>
            <RD>11th</RD>
            <STS>Street</STS>
            <HNO>42</HNO>
            <NAM>The Hotel DuPont</NAM>
            <PC>19801</PC>
          </civicAddress>
          </gp:location-info>
          <gp:usage-rules>
            <gbp:retransmission-allowed>true
            </gbp:retransmission-allowed>
            <gbp:retention-expiry>2013-12-10T20:00:00Z
            </gbp:retention-expiry>
          </gp:usage-rules>
          <gp:method>802.11</gp:method>

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

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

           <EmergencyCallDataValue>
            <EmergencyCallData.ProviderInfo
              xmlns=
                "urn:ietf:params:xml:ns:EmergencyCallData:ProviderInfo">
              <DataProviderReference>88QV4FpfZ976T@example.com
              </DataProviderReference>
                <DataProviderString>Diamond State Exemplar
                </DataProviderString>
              <ProviderID>urn:nena:companyid:diamond</ProviderID>
              <ProviderIDSeries>NENA</ProviderIDSeries>
              <TypeOfProvider>Access Network Provider</TypeOfProvider>
              <ContactURI>tel:+1-302-555-0000</ContactURI>
              <Language>en</Language>
            </EmergencyCallData.ProviderInfo>

            <EmergencyCallData.Comment
                xmlns="urn:ietf:params:xml:ns:EmergencyCallData:Comment">
                <DataProviderReference>88QV4FpfZ976T@example.com
                </DataProviderReference>
```

```
              <Comment xml:lang="en">This is an example text.</Comment>
           </EmergencyCallData.Comment>

          </EmergencyCallDataValue>
        </gp:provided-by>
      </gp:geopriv>
      <dm:deviceID>mac:00-0d-4b-30-72-df</dm:deviceID>
      <dm:timestamp>2013-07-09T20:57:29Z</dm:timestamp>
    </dm:device>
  </presence>
```

Figure 18: Access Network Provider returning PIDF-LO with Additional Data

7. XML Schemas

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

7.1. EmergencyCallData.ProviderInfo XML Schema

```
<?xml version="1.0"?>
<xs:schema
   targetNamespace=
        "urn:ietf:params:xml:ns:EmergencyCallData:ProviderInfo"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:pi="urn:ietf:params:xml:ns:EmergencyCallData:ProviderInfo"
   xmlns: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/2009/01/xml.xsd"/>

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

    <xs:element
        name="EmergencyCallData.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="DataProviderReference"
                       type="xs:token" minOccurs="1" maxOccurs="1"/>

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

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

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

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

                   <xs:element name="ContactURI" type="xs:anyURI"
                       minOccurs="1" maxOccurs="1"/>


   <xs:element name="Language" minOccurs="1" maxOccurs="unbounded">
       <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>


                   <xs:element name="DataProviderContact"
                       minOccurs="0" maxOccurs="1">
                     <xs:complexType>
                        <xs:sequence>
                          <xs:element minOccurs="0"
                              maxOccurs="unbounded" ref="xc:vcard"/>
                        </xs:sequence>
                     </xs:complexType>
                   </xs:element>
```

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

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

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

   </xs:schema>
```

Figure 19: EmergencyCallData.ProviderInfo XML Schema.

7.2. EmergencyCallData.ServiceInfo XML Schema

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

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

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

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

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

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

</xs:schema>
```

Figure 20: EmergencyCallData.ServiceInfo XML Schema.

7.3. EmergencyCallData.DeviceInfo XML Schema

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

```
        xmlns:dev="urn:ietf:params:xml:ns:EmergencyCallData:DeviceInfo"
        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="EmergencyCallData.DeviceInfo"
           type="dev:DeviceInfoType"/>

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

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

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

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

                   <xs:element name="UniqueDeviceID" minOccurs="0"
                               maxOccurs="unbounded">
                     <xs:complexType>
                       <xs:simpleContent>
                         <xs:extension base="xs:string">
                           <xs:attribute name="TypeOfDeviceID"
                                         type="xs:string"
                                         use="required"/>
                         </xs:extension>
                       </xs:simpleContent>
                     </xs:complexType>
                   </xs:element>

                   <xs:element name="DeviceSpecificData"
                       type="xs:anyURI" minOccurs="0" maxOccurs="1"/>

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

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

```
   </xs:schema>
```

Figure 21: EmergencyCallData.DeviceInfo XML Schema.

7.4. EmergencyCallData.SubscriberInfo XML Schema

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

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


           <xs:complexType name="SubscriberInfoType">
             <xs:complexContent>
              <xs:restriction base="xs:anyType">
               <xs:sequence>
                   <xs:element name="DataProviderReference"
                     type="xs:token" minOccurs="1" maxOccurs="1"/>

                   <xs:element name="SubscriberData">
                      <xs:complexType>
                        <xs:sequence>
                           <xs:element maxOccurs="unbounded"
                                       ref="xc:vcard"/>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>

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

```
               <xs:attribute name="privacyRequested" type="xs:boolean"
                   use="required"/>
              </xs:restriction>
             </xs:complexContent>
            </xs:complexType>

   </xs:schema>
```

Figure 22: EmergencyCallData.SubscriberInfo XML Schema.

7.5. EmergencyCallData.Comment XML Schema

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

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

                <xs:element name="Comment"
                    type="com:CommentSubType" minOccurs="0"
                    maxOccurs="unbounded"/>

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

    <xs:complexType name="CommentSubType">
     <xs:simpleContent>
      <xs:extension base="xs:string">
       <xs:attribute ref="xml:lang"/>
      </xs:extension>
     </xs:simpleContent>
    </xs:complexType>

</xs:schema>
```

Figure 23: EmergencyCallData.Comment XML Schema.

7.6. provided-by XML Schema

This section defines the provided-by schema.

```
 <?xml version="1.0"?>
```

```
 <xs:schema
     targetNamespace=
     "urn:ietf:params:xml:ns:EmergencyCallData"
     xmlns:xs="http://www.w3.org/2001/XMLSchema"
     xmlns:ad="urn:ietf:params:xml:ns:EmergencyCallData"
     xmlns:xml="http://www.w3.org/XML/1998/namespace"
     xmlns:pi="urn:ietf:params:xml:ns:EmergencyCallData:ProviderInfo"
     xmlns:svc="urn:ietf:params:xml:ns:EmergencyCallData:ServiceInfo"
     xmlns:dev="urn:ietf:params:xml:ns:EmergencyCallData:DeviceInfo"
     xmlns:sub=
         "urn:ietf:params:xml:ns:EmergencyCallData:SubscriberInfo"
     xmlns:com="urn:ietf:params:xml:ns:EmergencyCallData:Comment"
     elementFormDefault="qualified"
     attributeFormDefault="unqualified">

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

     <xs:element name="EmergencyCallDataReference"
                 type="ad:ByRefType"/>

     <xs:element name="EmergencyCallDataValue"
                 type="ad:EmergencyCallDataValueType"/>

     <!-- Additional Data By Reference -->

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

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

    <!-- Additional Data By Value -->

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

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

            </xs:sequence>
        </xs:complexType>

 </xs:schema>
```

Figure 24: provided-by XML Schema

8. Security Considerations

The data structures described in this document contain information usually considered private. When information is provided by value, entities that are a party to the SIP signaling (such as proxy servers and back-to-back user agents) will have access to it and need to protect it against inappropriate disclosure. An entity that is able to eavesdrop on the SIP signaling will also have access. Some access types (such as in-the-clear Wi-Fi) are more vulnerable than others (such as 3G or 4G cellular data traffic) to eavesdropping. Mechanisms that protect against eavesdropping (such as Transport Layer Security (TLS)) SHOULD be preferentially used whenever

feasible. (This requirement is not a "MUST" because there is an existing deployed base of clear-text SIP, and also because, as an emergency call, it is more important for the call to go through than for it to be protected; e.g., the call MUST proceed even if the TLS negotiation or certificate verification fails for whatever reason.) When information is provided by reference, HTTPS is REQUIRED for dereferencing, and the provider of the information is REQUIRED to validate the credentials of the requester. While the creation of a public key infrastructure (PKI) that has global scope may be difficult, the alternatives to creating devices and services that can provide critical information securely are more daunting. The provider of the information MAY enforce any policy it wishes to use, but PSAPs and responder agencies SHOULD deploy a PKI so that providers of additional data can check the certificate of the client (the requester) 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 are sufficient; the emergency services community within a country can arrange a PKI, data providers can be provisioned with the root CA public key for the country. Some nations are developing a PKI for this, and related, purposes. Since calls could be made from devices where the device and/or the service provider(s) are not local to the emergency services authorities, globally recognized credentials are useful. This might be accomplished by extending the notion of the "forest guide" described in [RFC5582] to allow the forest guide to provide the credential of the PKI root for areas for which it has coverage information, but standards for such a mechanism are not yet available. In its absence, the data provider needs to obtain by out of band means the root CA credentials for any areas to which it is willing to provide additional data. With the credential of the root CA for a national emergency services PKI, the data provider server can validate the credentials of an entity requesting additional data by reference.

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

At a minimum, the emergency services authorities could obtain a credential from the DNS entry of the domain in the Additional Data URI to at least validate that the server is known to the domain providing the URI.

Data provided by devices by reference have similar credential validation issues as for service providers, and while the solutions are the same, the challenges of doing so for every device are obviously more difficult, especially when considering root certificate updates, revocation lists, etc. However, in general, devices are not expected to provide data directly by reference, but rather, to either provide data by value, or upload the data to a server which can more reliably make it available and more easily enforce security policy. Devices which do provide data directly by reference, which might include fixed-location sensors, will need to be capable of handling this.

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

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

9. Privacy Considerations

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

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

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

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

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

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

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

Disclosure: When the data defined in this document is not properly protected (while in transit with traditional communication security techniques, and while stored using access control

mechanisms) there is the risk of disclosure, which is the revelation of private information about an individual.

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

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

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

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

policies any party in possession of the reference is able to resolve the reference and to obtain the data, including intermediaries.

10. IANA Considerations

10.1. Registry creation

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

10.1.1. Provider ID Series Registry

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

Private entities issuing or using internally-generated IDs are encouraged to register here and to ensure that all IDs they issue or use are unique. This guarantees that IDs issued or used by the entity are globally unique and distinguishable from other IDs issued or used by the same or a different entity. (Some organizations, such as NENA, issue IDs that are unique among all IDs they issue, so an entity using a combination of its NENA ID and the fact that it is from NENA is globally unique. Other entities might not have an ID issued by an organization such as NENA, so they are permitted to use their domain name, but if so, it needs to be unique.)

The content of this registry includes:

Name: An identifier to 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 1.

10.1.2. Service Environment Registry

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

The content of this registry includes:

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

Description: A short description of the value.

The initial set of values is listed in Figure 4.

10.1.3. Service Type Registry

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

The content of this registry includes:

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

Description: A short description of the value.

The initial set of values is listed in Figure 5.

10.1.4. Service Mobility Registry

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

The content of this registry includes:

Token: The value used in the <ServiceMobility> element.

Description: A short description of the value.

The initial set of values is listed in Figure 6.

10.1.5. Service Provider Type Registry

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

The content of this registry includes:

Token: The value used in the <TypeOfProvider> element.

Description: A short description of the type of service provider.

The initial set of values is defined in Figure 2.

10.1.6. Device Classification Registry

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

The content of this registry includes:

Token: Value used in the <DeviceClassification> element.

Description: Short description identifying the device type.

The initial set of values are defined in Figure 8.

10.1.7. 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 information which PSAPs and responders are able to use to uniquely identify a device. (For example, a biometric fingerprint used to authenticate a device would not normally be useful by a PSAP or responder to identify a device.)

The content of this registry includes:

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

Description: Short description identifying the type of the device ID.

The initial set of values are defined in Figure 9.

10.1.8. 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 "Specification Required" rules, which include an explicit expert review. The designated expert should ascertain that the proposed type is well understood, and provides information useful to PSAPs and responders. The specification must contain a complete description of the data, and a precise format specification suitable to allow interoperable implementations.

The content of this registry includes:

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

Description: Short description identifying the the data.

Specification: Citation for the specification of the data.

The initial set of values are listed in Figure 10.

10.1.9. Emergency Call Data Types Registry

This document creates a new sub-registry called 'Emergency Call Data Types' in the 'purpose' registry established by RFC 3261 [RFC3261]. As defined in [RFC5226], this registry operates under "Specification Required" rules, which include an explicit expert review. 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 expert is also responsible for verifying that the block is correctly categorized per the description of the categories in Section 1.

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

The content of this registry includes:

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

Data About: Indicates if the data describes the call, the caller, or the location (or is applicable to all), which helps PSAPs and other entities determine if they wish to process the block. The value MUST be either "The Call", "The Caller", "The Location", or "All". New values are created by extending this registry in a subsequent RFC.

Reference: The document that describes the data object

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

The initial set of values are listed in Figure 25.

Token	Data About	Reference
ProviderInfo	The Call	[This RFC]
ServiceInfo	The Call	[This RFC]
DeviceInfo	The Call	[This RFC]
SubscriberInfo	The Call	[This RFC]
Comment	The Call	[This RFC]

Figure 25: Additional Data Blocks Registry

10.2. 'EmergencyCallData' Purpose Parameter Value

This document defines the 'EmergencyCallData' value for the "purpose" parameter of the Call-Info header field. The Call-Info header and the corresponding registry for the 'purpose' parameter was established with RFC 3261 [RFC3261]. Note that 'EmergencyCallData' is a compound value; when used as a value of the 'purpose' parameter of the Call-Info header, 'EmergencyCallData' is immediately followed by a dot ('.') and a value from the 'Emergency Call Data Types' registry Section 10.1.9.

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

10.3. URN Sub-Namespace Registration for <provided-by> Registry Entry

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

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

10.4. MIME Registrations

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

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

MIME media type name: application

MIME subtype name: EmergencyCallData.ProviderInfo+xml

Mandatory parameters: none

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

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

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

Interoperability considerations: None

Published specification: [TBD: This specification]

Applications which use this media type: Emergency Services

Additional information:

Magic Number: None

File Extension: .xml

Macintosh file type code: 'TEXT'

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

Intended usage: LIMITED USE

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

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

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

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

MIME media type name: application

MIME subtype name: EmergencyCallData.ServiceInfo+xml

Mandatory parameters: none

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

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

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

eavesdropping of personal information. Please refer to Section 8 and Section 9 for more information.

Interoperability considerations: None

Published specification: [TBD: This specification]

Applications which use this media type: Emergency Services

Additional information:

Magic Number: None

File Extension: .xml

Macintosh file type code: 'TEXT'

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

Intended usage: LIMITED USE

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

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

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

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

MIME media type name: application

MIME subtype name: EmergencyCallData.DeviceInfo+xml

Mandatory parameters: none

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

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

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

Interoperability considerations: None

Published specification: [TBD: This specification]

Applications which use this media type: Emergency Services

Additional information:

Magic Number: None

File Extension: .xml

Macintosh file type code: 'TEXT'

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

Intended usage: LIMITED USE

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

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

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

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

MIME media type name: application

MIME subtype name: EmergencyCallData.SubscriberInfo+xml

Mandatory parameters: none

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

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

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

Interoperability considerations: None

Published specification: [TBD: This specification]

Applications which use this media type: Emergency Services

Additional information:

Magic Number: None

File Extension: .xml

Macintosh file type code: 'TEXT'

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

Intended usage: LIMITED USE

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

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

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

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

MIME media type name: application

MIME subtype name: EmergencyCallData.Comment+xml

Mandatory parameters: none

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

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

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

Interoperability considerations: None

Published specification: [TBD: This specification]

Applications which use this media type: Emergency Services

Additional information:

Magic Number: None

File Extension: .xml

Macintosh file type code: 'TEXT'

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

Intended usage: LIMITED USE

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

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

10.5. URN Sub-Namespace Registration

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

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

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

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

XML:

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

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

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

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

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
```

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

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

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

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
```

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

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

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

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
```

10.5.5. Registration for urn:ietf:params:xml:ns:EmergencyCallData:SubscriberInfo

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

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

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
```

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

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

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

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

XML:

```
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
```

10.6. Schema Registrations

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

URI: urn:ietf:params:xml:schema:emergencycalldata:ProviderInfo

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 19.

URI: urn:ietf:params:xml:schema:emergencycalldata:ServiceInfo

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 20.

URI: urn:ietf:params:xml:schema:emergencycalldata:DeviceInfo

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 21.

URI: urn:ietf:params:xml:schema:emergencycalldata:SubscriberInfo

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 7.4.

URI: urn:ietf:params:xml:schema:emergencycalldata: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 7.5.

10.7. VCard Parameter Value Registration

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

Value: main

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

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

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

11. Acknowledgments

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

We would also like to thank Paul Kyzivat, Gunnar Hellstrom, Martin Thomson, Keith Drage, Laura Liess, Chris Santer, Barbara Stark, Chris Santer, and Archie Cobbs for their review comments. Alissa Cooper and Guy Caron deserves special mention for their detailed and extensive review comments, which were very helpful and appreciated.

12. References

12.1. 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.

[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.

[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.

[RFC3966] Schulzrinne, H., "The tel URI for Telephone Numbers", RFC 3966, December 2004.

[RFC4119] Peterson, J., "A Presence-based GEOPRIV Location Object Format", RFC 4119, December 2005.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, October 2008.

[RFC5621] Camarillo, G., "Message Body Handling in the Session Initiation Protocol (SIP)", RFC 5621, September 2009.

[RFC5646] Phillips, A. and M. Davis, "Tags for Identifying Languages", BCP 47, RFC 5646, September 2009.

[RFC6350] Perreault, S., "vCard Format Specification", RFC 6350, August 2011.

[RFC6351] Perreault, S., "xCard: vCard XML Representation", RFC 6351, August 2011.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, January 2013.

[RFC7303] Thompson, H. and C. Lilley, "XML Media Types", RFC 7303, July 2014.

12.2. Informational References

[ECRIT-WG-wiki]
IETF, "ECRIT WG Wiki"", July 2015, <http://tools.ietf.org/wg/ecrit/trac/attachment/wiki/WikiStart/additional-data-examples.zip>.

[I-D.gellens-slim-negotiating-human-language]
Gellens, R., "Negotiating Human Language in Real-Time Communications", draft-gellens-slim-negotiating-human-language-01 (work in progress), April 2015.

[IANA-XML-Schemas]
IANA, "IANA XML Schemas"", July 2015, <http://www.iana.org/assignments/xml-registry/xml-registry.xhtml#schema>.

[LanguageTagRegistry]
IANA, "Language Subtag Registry", Feb 2015, <http://www.iana.org/assignments/language-subtag-registry/language-subtag-registry>.

[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.

[RFC3840] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", RFC 3840, August 2004.

[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.

[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.

[RFC5582] Schulzrinne, H., "Location-to-URL Mapping Architecture and Framework", RFC 5582, September 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.

[RFC7035] Thomson, M., Rosen, B., Stanley, D., Bajko, G., and A. Thomson, "Relative Location Representation", RFC 7035, October 2013.

[RFC7090] Schulzrinne, H., Tschofenig, H., Holmberg, C., and M. Patel, "Public Safety Answering Point (PSAP) Callback", RFC 7090, April 2014.

12.3. URIs

[1] http://www.nena.org/?page=cid2014

[2] http://www.nena.org/?page=CompanyID

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="ns1:altid"/>
      </xs:sequence>
    </xs:group>
    <xs:element name="altid">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="ns1:text"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!-- 5.5 -->
    <xs:group name="param-pid">
      <xs:sequence>
        <xs:element minOccurs="0" ref="ns1:pid"/>
      </xs:sequence>
    </xs:group>
    <xs:element name="pid">
      <xs:complexType>
        <xs:sequence>
          <xs:element maxOccurs="unbounded" name="text">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:pattern value="\d+(\.\d+)?"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!-- 5.6 -->
    <xs:group name="param-type">
      <xs:sequence>
        <xs:element minOccurs="0" ref="ns1:type"/>
      </xs:sequence>
    </xs:group>
    <xs:element name="type">
```

```
      <xs:complexType>
        <xs:sequence>
          <xs:element maxOccurs="unbounded" name="text">
            <xs:simpleType>
              <xs:restriction base="xs:token">
                <xs:enumeration value="work"/>
                <xs:enumeration value="home"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!-- 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>
    <!-- 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="ns1:x-name ns1: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="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:element ref="ns1: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="ns1:param-language"/>
                <xs:group ref="ns1:param-sort-as"/>
                <xs:group ref="ns1:param-altid"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element maxOccurs="unbounded" ref="ns1:surname"/>
          <xs:element maxOccurs="unbounded" ref="ns1:given"/>
          <xs:element maxOccurs="unbounded" ref="ns1:additional"/>
          <xs:element maxOccurs="unbounded" ref="ns1:prefix"/>
          <xs:element maxOccurs="unbounded" ref="ns1:suffix"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="surname" type="xs:string"/>
    <xs:element name="given" type="xs:string"/>
    <xs:element name="additional" type="xs:string"/>
    <xs:element name="prefix" type="xs:string"/>
    <xs:element 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="ns1:param-language"/>
                <xs:group ref="ns1:param-altid"/>
                <xs:group ref="ns1:param-pid"/>
```

```
                <xs:group ref="ns1:param-pref"/>
                <xs:group ref="ns1:param-type"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:group ref="ns1:value-text-list"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!-- 6.2.4 -->
    <xs:element name="photo">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="parameters">
            <xs:complexType>
              <xs:sequence>
                <xs:group ref="ns1:param-altid"/>
                <xs:group ref="ns1:param-pid"/>
                <xs:group ref="ns1:param-pref"/>
                <xs:group ref="ns1:param-type"/>
                <xs:group ref="ns1:param-mediatype"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element ref="ns1:uri"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!-- 6.2.5 -->
    <xs:element name="bday">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="parameters">
            <xs:complexType>
              <xs:sequence>
                <xs:group ref="ns1:param-altid"/>
                <xs:group ref="ns1:param-calscale"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:choice>
            <xs:element ref="ns1:value-date-and-or-time"/>
            <xs:element ref="ns1:text"/>
          </xs:choice>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!-- 6.2.6 -->
```

```
    <xs:element name="anniversary">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="parameters">
            <xs:complexType>
              <xs:sequence>
                <xs:group ref="ns1:param-altid"/>
                <xs:group ref="ns1:param-calscale"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:choice>
            <xs:element ref="ns1:value-date-and-or-time"/>
            <xs:element ref="ns1:text"/>
          </xs:choice>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!-- 6.2.7 -->
    <xs:element name="gender">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="ns1:sex"/>
          <xs:element minOccurs="0" ref="ns1:identity"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="sex">
      <xs:simpleType>
        <xs:restriction base="xs:token">
          <xs:enumeration value=""/>
          <xs:enumeration value="M"/>
          <xs:enumeration value="F"/>
          <xs:enumeration value="O"/>
          <xs:enumeration value="N"/>
          <xs:enumeration value="U"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="identity" type="xs:string"/>
    <!-- 6.3.1 -->
    <xs:group name="param-label">
      <xs:sequence>
        <xs:element minOccurs="0" ref="ns1:label"/>
      </xs:sequence>
    </xs:group>
    <xs:element name="label">
      <xs:complexType>
```

```
        <xs:sequence>
          <xs:element ref="ns1:text"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="adr">
      <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:group ref="ns1:param-geo"/>
                <xs:group ref="ns1:param-tz"/>
                <xs:group ref="ns1:param-label"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element maxOccurs="unbounded" ref="ns1:pobox"/>
          <xs:element maxOccurs="unbounded" ref="ns1:ext"/>
          <xs:element maxOccurs="unbounded" ref="ns1:street"/>
          <xs:element maxOccurs="unbounded" ref="ns1:locality"/>
          <xs:element maxOccurs="unbounded" ref="ns1:region"/>
          <xs:element maxOccurs="unbounded" ref="ns1:code"/>
          <xs:element maxOccurs="unbounded" ref="ns1:country"/>
        </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="ns1:param-altid"/>
                <xs:group ref="ns1:param-pid"/>
```

```
                <xs:group ref="ns1:param-pref"/>
                <xs:element minOccurs="0" name="type">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element maxOccurs="unbounded" name="text">
                        <xs:simpleType>
                          <xs:restriction base="xs:token">
                            <xs:enumeration value="work"/>
                            <xs: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="ns1:param-mediatype"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:choice>
            <xs:element ref="ns1:text"/>
            <xs:element ref="ns1: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="ns1:param-altid"/>
                <xs:group ref="ns1:param-pid"/>
                <xs:group ref="ns1:param-pref"/>
                <xs:group ref="ns1:param-type"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element ref="ns1: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="ns1:param-altid"/>
                <xs:group ref="ns1:param-pid"/>
                <xs:group ref="ns1:param-pref"/>
                <xs:group ref="ns1:param-type"/>
                <xs:group ref="ns1:param-mediatype"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element ref="ns1:uri"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!-- 6.4.4 -->
    <xs:element name="lang">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="parameters">
            <xs:complexType>
              <xs:sequence>
                <xs:group ref="ns1:param-altid"/>
                <xs:group ref="ns1:param-pid"/>
                <xs:group ref="ns1:param-pref"/>
                <xs:group ref="ns1:param-type"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element ref="ns1:language-tag"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!-- 6.5.1 -->
    <xs:group name="property-tz">
      <xs:sequence>
        <xs:element name="tz">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" name="parameters">
                <xs:complexType>
```

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

```
          <xs:element ref="ns1: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="ns1:param-language"/>
                <xs:group ref="ns1:param-altid"/>
                <xs:group ref="ns1:param-pid"/>
                <xs:group ref="ns1:param-pref"/>
                <xs:group ref="ns1:param-type"/>
                <xs:group ref="ns1:param-sort-as"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:group ref="ns1: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="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>
    <!-- 6.6.6 -->
    <xs:element name="related">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="parameters">
            <xs:complexType>
```

```
            <xs:sequence>
              <xs:group ref="ns1:param-altid"/>
              <xs:group ref="ns1:param-pid"/>
              <xs:group ref="ns1:param-pref"/>
              <xs:element minOccurs="0" name="type">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element maxOccurs="unbounded" name="text">
                      <xs: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="ns1:param-mediatype"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element ref="ns1:uri"/>
          <xs:element ref="ns1: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="ns1:param-altid"/>
                <xs:group ref="ns1:param-pid"/>
                <xs:group ref="ns1:param-pref"/>
                <xs:group ref="ns1:param-type"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:group ref="ns1:value-text-list"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!-- 6.7.2 -->
    <xs:element name="note">
      <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:element ref="ns1:text"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!-- 6.7.3 -->
    <xs:element name="prodid">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="ns1:text"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!-- 6.7.4 -->
    <xs:element name="rev" type="ns1: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="ns1:param-language"/>
            <xs:group ref="ns1:param-altid"/>
            <xs:group ref="ns1:param-pid"/>
            <xs:group ref="ns1:param-pref"/>
            <xs:group ref="ns1:param-type"/>
            <xs:group ref="ns1:param-mediatype"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element ref="ns1:uri"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!-- 6.7.6 -->
<xs:element name="uid">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ns1:uri"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!-- 6.7.7 -->
<xs:element name="clientpidmap">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ns1:sourceid"/>
      <xs:element ref="ns1:uri"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="sourceid" type="xs:positiveInteger"/>
<!-- 6.7.8 -->
<xs:element name="url">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="parameters">
        <xs:complexType>
          <xs:sequence>
            <xs:group ref="ns1:param-altid"/>
            <xs:group ref="ns1:param-pid"/>
            <xs:group ref="ns1:param-pref"/>
```

```
                <xs:group ref="ns1:param-type"/>
                <xs:group ref="ns1:param-mediatype"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element ref="ns1:uri"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!-- 6.8.1 -->
    <xs:element name="key">
      <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:choice>
            <xs:element ref="ns1:uri"/>
            <xs:element ref="ns1: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="ns1:param-altid"/>
                <xs:group ref="ns1:param-pid"/>
                <xs:group ref="ns1:param-pref"/>
                <xs:group ref="ns1:param-type"/>
                <xs:group ref="ns1:param-mediatype"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element ref="ns1:uri"/>
        </xs:sequence>
```

```
      </xs:complexType>
    </xs:element>
    <!-- 6.9.2 -->
    <xs:element name="caladruri">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="parameters">
            <xs:complexType>
              <xs:sequence>
                <xs:group ref="ns1:param-altid"/>
                <xs:group ref="ns1:param-pid"/>
                <xs:group ref="ns1:param-pref"/>
                <xs:group ref="ns1:param-type"/>
                <xs:group ref="ns1:param-mediatype"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element ref="ns1:uri"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!-- 6.9.3 -->
    <xs:element name="caluri">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="parameters">
            <xs:complexType>
              <xs:sequence>
                <xs:group ref="ns1:param-altid"/>
                <xs:group ref="ns1:param-pid"/>
                <xs:group ref="ns1:param-pref"/>
                <xs:group ref="ns1:param-type"/>
                <xs:group ref="ns1:param-mediatype"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element ref="ns1:uri"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!-- Top-level grammar -->
    <xs:group name="property">
      <xs:choice>
        <xs:element ref="ns1:adr"/>
        <xs:element ref="ns1:anniversary"/>
        <xs:element ref="ns1:bday"/>
        <xs:element ref="ns1:caladruri"/>
        <xs:element ref="ns1:caluri"/>
```

```
        <xs:element ref="ns1:categories"/>
        <xs:element ref="ns1:clientpidmap"/>
        <xs:element ref="ns1:email"/>
        <xs:element ref="ns1:fburl"/>
        <xs:element ref="ns1:fn"/>
        <xs:group ref="ns1:property-geo"/>
        <xs:element ref="ns1:impp"/>
        <xs:element ref="ns1:key"/>
        <xs:element ref="ns1:kind"/>
        <xs:element ref="ns1:lang"/>
        <xs:element ref="ns1:logo"/>
        <xs:element ref="ns1:member"/>
        <xs:element ref="ns1:n"/>
        <xs:element ref="ns1:nickname"/>
        <xs:element ref="ns1:note"/>
        <xs:element ref="ns1:org"/>
        <xs:element ref="ns1:photo"/>
        <xs:element ref="ns1:prodid"/>
        <xs:element ref="ns1:related"/>
        <xs:element ref="ns1:rev"/>
        <xs:element ref="ns1:role"/>
        <xs:element ref="ns1:gender"/>
        <xs:element ref="ns1:sound"/>
        <xs:element ref="ns1:source"/>
        <xs:element ref="ns1:tel"/>
        <xs:element ref="ns1:title"/>
        <xs:group ref="ns1:property-tz"/>
        <xs:element ref="ns1:uid"/>
        <xs:element ref="ns1:url"/>
      </xs:choice>
    </xs:group>

    <xs:element name="vcards">
      <xs:complexType>
        <xs:sequence>
          <xs:element maxOccurs="unbounded" ref="ns1: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="ns1:property"/>
            <xs:element ref="ns1:group"/>
          </xs:choice>
```

```
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:element name="vcard"  type="ns1: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>
```

Appendix B. XML Validation

This document defines a number of XML schemas and contains various examples. Extracting the XML and validating the examples against the schemas can be challenging, especially due to the formatting limitations introduced by IETF RFCs. For those readers who copy the XML schemas and examples directly from this document, please consider that errors might be introduced due to line breaks and extra whitespaces in the regular expressions contained in the vcard schema in Appendix A. To validate the PIDF-LO from Figure 18 it is also necessary to consult the referenced RFCs and copy the schemas necessary for successful validation.

The XML schemas found in this document include a 'SchemaLocation' attribute. Depending on the location of the downloaded schema files you may need to adjust this schema location or configure your XML editor to point to the location.

For convenience of readers, the schemas are available at http://ip-emergency.net/additional-data.zip and the XML examples are available at the IETF ECRIT Working Group wiki page [ECRIT-WG-wiki].

Note to RFC Editor: After IANA has published the schemas, the above link to the schemas should be replaced with [IANA-XML-Schemas].

Authors' Addresses

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

Email: rg+ietf@qti.qualcomm.com

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

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

Hannes Tschofenig
Hall in Tirol 6060
Austria

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

Roger Marshall
TeleCommunication Systems, Inc.
2401 Elliott Avenue
Seattle, WA 98121
US

Phone: +1 206 792 2424
Email: rmarshall@telecomsys.com
URI: http://www.telecomsys.com

James Winterbottom
AU

Email: a.james.winterbottom@gmail.com

ECRIT
Internet-Draft
Intended status: Informational
Expires: January 3, 2016

R. Gellens
Qualcomm Technologies, Inc
B. Rosen
NeuStar, Inc.
H. Tschofenig

July 4, 2015

Next-Generation Vehicle-Initiated Emergency Calls
draft-ietf-ecrit-car-crash-03.txt

Abstract

This document describes how to use IP-based emergency services mechanisms to support the next generation of emergency calls placed by vehicles (automatically in the event of a crash or serious incident, or manually invoked by a vehicle occupant) and conveying vehicle, sensor, and location data related to the crash or incident. Such calls are often referred to as "Automatic Crash Notification" (ACN), or "Advanced Automatic Crash Notification" (AACN), even in the case of manual trigger. The "Advanced" qualifier refers to the ability to carry a richer set of data.

This document also registers a MIME Content Type and an Emergency Call Additional Data Block for the vehicle, sensor, and location data (often referred to as "crash data" even though there is not necessarily a crash). An external specification for the data format, contents, and structure are referenced in this document.

Profiling and simplifications of the general emergency call mechanism, as described in [RFC6443] and [RFC6881], are possible due to the nature of the functionality that is provided in vehicles such as the usage of Global Satellite Navigation System (GNSS).

This document reuses the technical aspects of next-generation pan-European eCall (a mandated and standardized system for emergency calls by in-vehicle systems within Europe and other regions), as described in [I-D.ietf-ecrit-ecall]. However, this document specifies a different set of vehicle (crash) data, specifically, the Vehicle Emergency Data Set (VEDS) rather than the eCall Minimum Set of Data (MSD).

Status of This Memo

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

Copyright Notice

Table of Contents

1. Terminology

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

This document re-uses terminology defined in Section 3 of [RFC5012].

Additionally, we use the following abbreviations:

Term	Expansion
3GPP	3rd Generation Partnership Project
AACN	Advanced Automatic Crash Notification
ACN	Automatic Crash Notification
APCO	Association of Public-Safety Communications Officials
EENA	European Emergency Number Association
ESInet	Emergency Services IP network
GNSS	Global Satellite Navigation System (which includes the various such systems including the Global Positioning System or GPS)
IVS	In-Vehicle System
MNO	Mobile Network Operator
NENA	National Emergency Number Association
TSP	Telematics Service Provider
VEDS	Vehicle Emergency Data Set

2. Introduction

Emergency calls made by in-vehicle systems (e.g., in the event of a crash) assist in significantly reducing road deaths and injuries by allowing emergency services to respond quickly and often with better location.

Drivers often have a poor location awareness, especially outside of major cities, at night and when away from home (especially abroad).

In the most crucial cases, the victim(s) may not be able to call because they have been injured or trapped.

For more than a decade, some vehicles have been equipped with telematics systems that, among other features, place an emergency call automatically in the event of a crash or manually in response to an emergency call button. Such systems generally have on-board location determination systems that make use of satellite-based positioning technology, inertial sensors, gyroscopes, etc., to provide a fairly accurate position for the vehicle. Such built-in systems can take advantage of the benefits of being integrated into a vehicle, such as more reliable power, ability to have larger or specialized antenna, ability to be engineered to avoid or minimise degradation by vehicle glass coatings, interference from other vehicle systems, etc. Thus, the PSAP can be provided with a good estimate of where the vehicle is during an emergency. Vehicle manufacturers are increasingly adopting such systems, both for the safety benefits and for the additional features and services they enable (e.g., remote engine diagnostics, remote door unlock, stolen vehicle tracking and disabling, etc.).

The general term for such systems is Automatic Crash Notification (ACN) or "Advanced Automatic Crash Notification" (AACN). "ACN" is used in this document as a general term. ACN systems transmit some amount of data specific to the incident, referred to generally as "crash data" (the term is commonly used even though there might not have been a crash). While different systems transmit different amounts of crash data, standardized formats, structures, and mechanisms are needed to provide interoperability among systems and PSAPs.

Currently deployed in-vehicle telematics systems are circuit-switched and lack a standards-based ability to convey crash data directly to the PSAP (generally relying on either a human call taker or an automated system to provide the PSAP call taker with some crash data orally, or possibly a proprietary mechanism). The PSAP call taker needs to first realize that the call is related to a vehicle incident, and in most cases must then listen to the data and transcribe it.

The transition to next-generation calling in general, and emergency calling in particular, provides an opportunity to vastly improve the scope, breadth, reliability and usefulness of crash data during an emergency by allowing it to be presented alongside the call, and to be automatically processed by the PSAP and made available to the call taker in an integrated, automated way. In addition, vehicle manufacturers are provided an opportunity to take advantage of the same standardized mechanisms for data transmission for internal use

if they wish (such as telemetry between the vehicle and a service center for both emergency and non-emergency uses, including location-based services, multi-media entertainment systems, and road-side assistance applications).

Next-generation ACN provides an opportunity for such calls to be recognized and processed as such during call set-up, and optionally routed to an upgraded PSAP where the vehicle data is available to assist the call taker in assessing and responding to the situation.

An ACN call may be either occupant-initiated or automatically triggered. (The "A" in "ACN" does stand for "Automatic," but the term is often used to refer to the class of calls that are placed by an in-vehicle system (IVS) and that carry incident-related data as well as voice.) Automatically triggered calls indicate a car crash or some other serious incident (e.g., a fire) and carry a greater presumption of risk of injury. Manually triggered calls are often reports of serious hazards (such as impaired drivers or roadway debris) and may require different responses depending on the situation. Manually triggered calls are also more likely to be false (e.g., accidental) calls and may thus be subject to different handling by the PSAP.

This document describes how the IETF mechanisms for IP-based emergency calls, including [RFC6443] and [I-D.ietf-ecrit-additional-data], are used to provide the realization of next-generation ACN.

The Association of Public-Safety Communications Officials (APCO) and the National Emergency Number Association (NENA) have jointly developed a standardized set of incident-related vehicle data for ACN use, called the Vehicle Emergency Data Set (VEDS) [VEDS]. Such data is often referred to as crash data although it is applicable in incidents other than crashes.

VEDS provides a standard data set for the transmission, exchange, and interpretation of vehicle-related data. A standard data format allows the data to be generated by an IVS, and interpreted by PSAPs, emergency responders, and medical facilities (including those capable of providing trauma level patient care). It includes incident-related information such as airbag deployment, location of the vehicle, if the vehicle was involved in a rollover, various sensor data that can indicate the potential severity of the crash and the likelihood of severe injuries to the vehicle occupants, etc. This data better informs the PSAP and emergency responders as to the type of response that may be needed. This information was recently included in the federal guidelines for field triage of injured patients. These guidelines are designed to help responders at the

accident scene identify the potential existence of severe internal injuries and to make critical decisions about how and where a patient needs to be transported.

This document registers the 'application/EmergencyCallData.VEDS+xml' MIME content-type, and registers the 'VEDS' entry in the Emergency Call Additional Data registry.

VEDS is an XML structure (see [VEDS]). The 'application/ EmergencyCallData.VEDS+xml' MIME content-type is used to identify it. The 'VEDS' entry in the Emergency Call Additional Data registry is used to construct a 'purpose' parameter value for conveying VEDS data in a Call-Info header (as described in [I-D.ietf-ecrit-additional-data]).

VEDS is a versatile structure that can accomodate varied needs. However, if additional sets of data are determined to be needed (e.g., in the future or in different regions), the steps to enable each data block are very briefly summarized below:

- A standardized format and encoding (such as XML) is defined and published by a Standards Development Organization (SDO).
- A MIME Content-Type is registered for it (typically under the 'Application' media type and with a sub-type starting with 'EmergencyCallData.').
- An entry for the block is added to the Emergency Call Additional Data Blocks sub-registry (established by [I-D.ietf-ecrit-additional-data]); the registry entry is the root of the MIME sub-type (not including the 'EmergencyCallData' prefix and any suffix such as '+xml').

A next-generation In-Vehicle System (IVS) transmits crash data by encoding it in a standardized and registered format (such as VEDS) and attaching it to an INVITE as a MIME body part. The body part is identified by its MIME content-type (such as 'application/ EmergencyCallData.VEDS+xml') in the Content-Type header field of the body part. The body part is assigned a unique identifier which is listed in a Content-ID header field in the body part. The INVITE is marked as containing the crash data by adding a Call-Info header field at the top level of the INVITE. This Call-Info header field contains a CID URL referencing the body part's unique identifier, and a 'purpose' parameter identifying the data as the crash data per the registry entry; the 'purpose' parameter's value is 'EmergencyCallData.' and the root of the MIME type (not including the 'EmergencyCallData' prefix and any suffix such as '+xml' (e.g., 'purpose=EmergencyCallData.VEDS').

The mechanisms described here are thus used to place emergency calls that are identifiable as ACN calls and that carry one or more standardized crash data objects in an interoperable way.

3. Overview of Current Deployment Models

Current (circuit-switched or legacy) systems for placing emergency calls by in-vehicle systems, including automatic crash notification systems, generally have a limited ability to convey at least location and in some cases telematics data to the PSAP. Most such systems use one of three architectural models, which are described here as: "Telematics Service Provider" (TSP), "direct", and "paired handset". These three models are illustrated below.

In the TSP model, both emergency and non-emergency calls are placed to a Telematics Service Provider (TSP); a proprietary technique is used for data transfer (such as proprietary in-band modems) to the TSP.

In an emergency, the TSP call taker bridges in the PSAP and communicates location, crash data (such as impact severity and trauma prediction), and other data (such as the vehicle description) to the PSAP call taker verbally. Typically, a three-way voice call is established between the vehicle, the TSP, and the PSAP, allowing communication between the PSAP call taker, the TSP call taker, and the vehicle occupants (who might be unconscious).

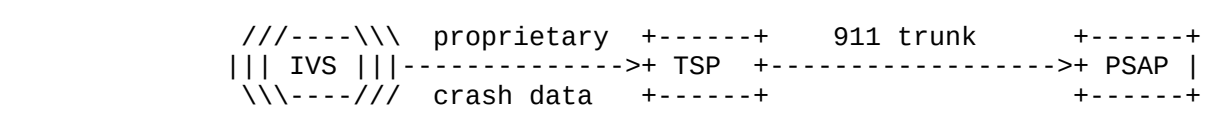


Figure 1: Legacy TSP Model.

In the paired model, the IVS uses a Bluetooth link with a previously-paired handset to establish an emergency call with the PSAP (by dialing a standard emergency number such as 9-1-1), and then communicates location data to the PSAP via text-to-speech; crash data is not conveyed. Some such systems use an automated voice prompt menu (e.g., "this is an automatic emergency call from a vehicle; press 1 to open a voice path to the vehicle; press 2 to hear the location read out") to allow the call taker to request location data via text-to-speech.

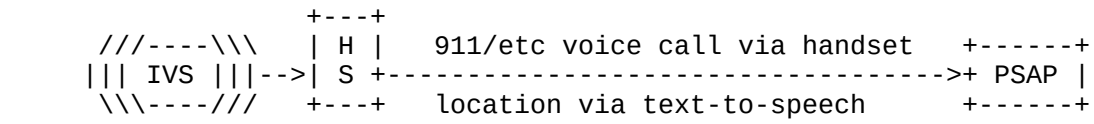


Figure 2: Legacy Paired Model

In the direct model, the IVS directly places an emergency call with the PSAP by dialing a standard emergency number such as 9-1-1. Such systems might communicate location data to the PSAP via text-to-speech; crash data might not be conveyed.

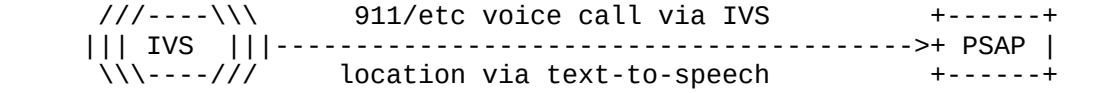


Figure 3: Legacy Direct Model

4. Document Scope

This document is focused on the interface to the PSAP, that is, how an ACN emergency call is setup and incident-related data (including vehicle, sensor, and location data) is transmitted to the PSAP using IETF specifications. (The goal is to re-use specifications rather than to invent new.) For the direct model, this is the end-to-end description (between the vehicle and the PSAP). For the TSP model, this describes the right-hand side (between the TSP and the PSAP), leaving the left-hand side (between the vehicle and the TSP) up to the entities involved (i.e., IVS and TSP vendors) who are then free to use the same mechanism as for the right-hand side (or not).

Note that while ACN systems in the U.S. and other regions are not currently mandated, Europe has a mandated and standardized system for emergency calls by in-vehicle systems. This pan-European system is known as "eCall" and is the subject of a separate document, [I-D.ietf-ecrit-ecall]. Vehicles designed to operate in multiple regions may need to support eCall as well as the ACN described here. If other regions devise their own specifications or data formats, a multi-region vehicle may need to support those as well. This document adopts the call set-up and other technical aspects of [I-D.ietf-ecrit-ecall], which uses [I-D.ietf-ecrit-additional-data], which makes it easy to substitute a different data set while keeping other technical aspects unchanged. Hence, both NG-eCall and the ACN mechanism described here are fully compatible, differing only in the specific data block that is sent (the eCall MSD in the case of NG-eCall, and the APCO/NENA VEDS used in this document). If other

regions adopt their own data set, this can be similarly accomodated without changing other technical aspects.

5. Migration to Next-Generation

Migration of emergency calls placed by in-vehicle systems to next-generation (all-IP) technology provides a standardized mechanism to identify such calls and to present crash data with the call. This allows ACN calls and crash data to be automatically processed by the PSAP and made available to the call taker in an integrated, automated way. Because the crash data is carried in the initial SIP INVITE (per [I-D.ietf-ecrit-additional-data]) the PSAP can present it to the call taker simultaneously with the appearance of the call.

Vehicle manufacturers using the TSP model may choose to take advantage of the same mechanism to carry telematics data between the vehicle and the TSP for both emergency and non-emergency calls.

A next-generation IVS establishes an emergency call using the emergency call solution as described in [RFC6443] and [RFC6881], with the difference that the Request-URI indicates an ACN type of emergency call and a Call-Info header field indicates that vehicle crash data is attached. When an ESInet is deployed the MNO only needs to recognize the call as an emergency call and route it to an ESInet. The ESInet may recognize the call as an ACN with vehicle data and may route the call to an NG-ACN capable PSAP. Such a PSAP would interpret the vehicle data sent with the call and make it available to the call taker.

Because of the need to identify and specially process Next-Generation ACN calls (as discussed above), [I-D.ietf-ecrit-ecall] registers new service URN children within the "sos" subservice. These URNs provide a mechanism by which an NG-ACN call is identified, and differentiate between manually and automatically triggered NG-ACN calls, which can be subject to different treatment, depending on policy. (The two service URNs registered in [I-D.ietf-ecrit-ecall] are: urn:service:sos.ecall.automatic and urn:service:sos.ecall.manual.)

Note that in North America, routing queries performed by clients outside of an ESInet are likely to treat all sub-services of "sos" identically to "sos" with no sub-service. However, the Request-URI header field retains the full sub-service; route and handling decisions within an ESInet or PSAP may take the sub-service into account. For example, in a region with multiple cooperating PSAPs, an NG-ACN call might be routed to a PSAP that is NG-ACN capable, or one that specializes in vehicle-related incidents.

Migration of the three architectural models to next-generation (all-IP) is described below.

In the TSP model, the IVS transmits crash and location data to the TSP using either a protocol that is based on a proprietary design or one that re-uses IETF specifications. In an emergency, the TSP call taker bridges in the PSAP and the TSP transmits crash and other data to the PSAP using IETF specifications. There is a three-way call between the vehicle, the TSP, and the PSAP, allowing communication between the PSAP call taker, the TSP call taker, and the vehicle occupants (who might be unconscious).

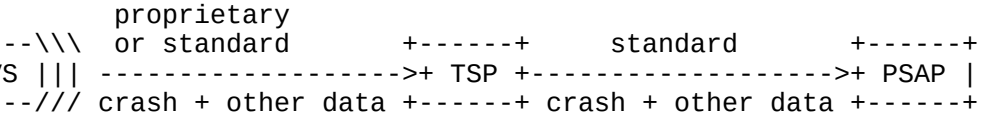


Figure 4: Next-Generation TSP Model

The vehicle manufacturer and the TSP may choose to use the same IETF specifications to transmit crash and location data from the vehicle to the TSP as is described here to transmit such data from the TSP to the PSAP.

In the paired model, the IVS uses a Bluetooth link to a previously-paired handset to establish an emergency call with the PSAP; it is not clear what facilities are or will be available for transmitting crash data through the Bluetooth link to the handset for inclusion in an NG emergency call.

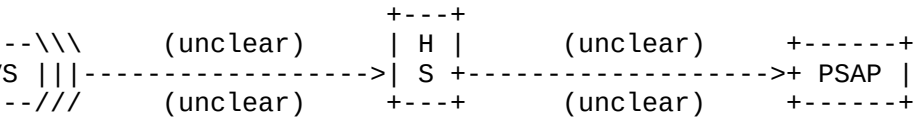


Figure 5: Next-Generation Paired Model

In the direct model, the IVS communicates crash data to the PSAP directly using IETF specifications.

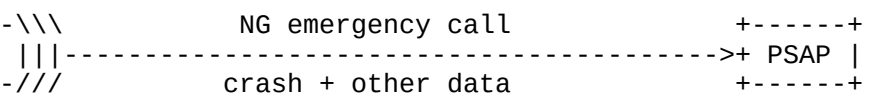


Figure 6: Next-Generation Model

If the call is routed to a PSAP that is not capable of processing the vehicle data, the PSAP ignores (or does not receive) the vehicle

data. This is detectable by the IVS or TSP when it receives a 200 OK to the INVITE that lacks an eCall control structure acknowledging receipt of the data [I-D.ietf-ecrit-ecall]. The IVS or TSP then proceeds as it would for a non-NG ACN call (e.g., verbal conveyance of data)

6. Profile

In the context of emergncy calls placed by an in-vehicle system it is assumed that the car is equipped with a built-in GNSS receiver. For this reason only geodetic location information will be sent within an emergency call. The following location shapes MUST be implemented: 2d and 3d Point (see Section 5.2.1 of [RFC5491]), Circle (see Section 5.2.3 of [RFC5491]), and Ellipsoid (see Section 5.2.7 of [RFC5491]). The coordinate reference systems (CRS) specified in [RFC5491] are also mandatory for this document. The <direction> element, as defined in [RFC5962] which indicates the direction of travel of the vehicle, is important for dispatch and hence it MUST be included in the PIDF-LO [RFC4119]. The <heading> element specified in [RFC5962] MUST be implemented and MAY be included.

Calls by in-vehicle systems are placed via cellular networks, which may ignore location sent by an originating device in an emergency call INVITE, instead attaching their own location (often determined in cooperation with the originating device). Standardized crash data structures often include location as determined by the IVS. A benefit of this is that it allows the PSAP to see both the location as determined by the cellular network (often in cooperation with the originating device) and the location as determined by the IVS.

This specification inherits the ability to utilize test call functionality from Section 15 of [RFC6881].

7. Call Setup

It is important that ACN calls be easily identifiable as such at all stages of call handling, and that automatic versus manual triggering be known. ACN calls differ from general emergency calls in several aspects, including the presence of standardized crash data, the fact that the call is known to be placed by an in-vehicle system (which has implications for PSAP operational processes), and, especially for automatic calls, information that may indicate a likelihood of severe injury and hence need for trauma services. Knowledge that a call is an ACN and further that it was automatically or manually invoked carries a range of implications about the call, the circumstances, and the vehicle occupants. Calls by in-vehicle systems may be considered a specific sub-class of general emergency calls and are optimally handled by a PSAP with the technical and operational

capabilities to serve such calls. (This is especially so in environments such as the U.S. where there are many PSAPs and where individual PSAPs have a range of capabilities.) Technical capabilities include the ability to recognize and process standardized crash data. Operational capabilities include training and processes for assessing severe injury likelihood and responding appropriately (e.g., dispatching trauma-capable medical responders or those trained and equipped to extract occupants from crashed vehicles and handle gasoline or other hazardous materials, transporting victims to a trauma center, alerting the receiving facility, etc.).

Because ACN calls differ in significant ways from general emergency calls, and because such calls should be handled by specialized PSAPs (equipped technically to interpet and make use of crash data, and operationally to handle emergency calls placed by in-vehicle systems), [I-D.ietf-ecrit-ecall] registers SOS sub-services. Using a sub-service makes it readily obvious that the call is an ACN; a further child element distinguishes calls automatically placed due to a crash or other serious incident (such as a fire) from those manually invoked by a vehicle occupant (specifically, "SOS.ecall.automatic" and "SOS.ecall.manual"). The distinction between automatic and manual invocation is also significant; automatically triggered calls indicate a car crash or some other serious incident (e.g., a fire) and carry a greater presumption of risk of injury and hence need for specific responders (such as trauma or fire). Manually triggered calls are often reports of serious hazards (such as drunk drivers) and may require different responses depending on the situation. Manually triggered calls are also more likely to be false (e.g., accidental) calls and may thus be subject to different handling by the PSAP.

A next-generation In-Vehicle System (IVS) transmits crash data by encoding it in a standardized and registered format and attaching it to an INVITE as an additional data block as specified in Section 4.1 of [I-D.ietf-ecrit-additional-data]. As described in that document, the block is identified by its MIME content-type, and pointed to by a CID URL in a Call-Info header with a ’purpose’ parameter value corresponding to the block.

Specifically, the steps required during standardization are:

- A set of crash data is standardized by an SDO or appropriate organization

- A MIME Content-Type for the crash data set is registered with IANA

    * If the data is specifically for use in emergency calling, the MIME type is normally under the 'application' type with a subtype starting with 'EmergencyCallData.'

    * If the data format is XML, then by convention the name has a suffix of '+xml'

- The item is registered in the Emergency Call Additional Data registry, as defined in Section 9.1.7 of [I-D.ietf-ecrit-additional-data]

    * For emergency-call-specific formats, the registered name is the root of the MIME Content-Type (not including the 'EmergencyCallData' prefix and any suffix such as '+xml') as described in Section 4.1 of [I-D.ietf-ecrit-additional-data]

When placing an emergency call:

- The crash data set is created and encoded per its specification

- The crash data set is attached to the emergency call INVITE as specified in Section 4.1 of [I-D.ietf-ecrit-additional-data], that is, as a MIME body part identified by its MIME Content-Type in the body part's Content-Type header field

- The body part is assigned a unique identifier label in a Content-ID header field of the body part

- A Call-Info header field at the top level of the INVITE is added that references the crash data and identifies it by its MIME root (as registered in the Emergency Call Additional Data registry)

    * The crash data is referenced in the Call-Info header field by a CID URL that contains the unique Content ID assigned to the crash data body part

    * The crash data is identified in the Call-Info header field by a 'purpose' parameter whose value is 'EmergencyCallData.' concatenated with the specific crash data entry in the Emergency Call Additional Data registry

    * The Call-Info header field MAY be either solely to reference the crash data (and hence have only the one URL) or may also contain other URLs referencing other data

- Additional crash data sets MAY be included by following the same steps

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

The VEDS data is attached as a body part with MIME content type 'application/EmergencyCallData.VEDS+xml' which is pointed at by a Call-Info URL of type CID with a 'purpose' parameter of 'EmergencyCallData.VEDS'.

Entities along the path between the vehicle and the PSAP are able to identify the call as an ACN call and handle it appropriately. The PSAP is able to identify the crash data as well as any other additional data attached to the INVITE by examining the Call-Info header fields for 'purpose' parameters whose values start with 'EmergencyCallData.' The PSAP is able to access and the data it is capable of handling and is interested in by checking the 'purpose' parameter values.

8. Call Routing

An Emergency Services IP Network (ESInet) is a network operated by or on behalf of emergency services authorities. It handles emergency call routing and processing before delivery to a PSAP. In the NG9-1-1 architecture adopted by NENA as well as the NG1-1-2 architecture adopted by EENA, each PSAP is connected to one or more ESInets. Each originating network is also connected to one or more ESInets. The ESInets maintain policy-based routing rules which control the routing and processing of emergency calls. The centralization of such rules within ESInets provides for a cleaner separation between the responsibilities of the originating network and that of the emergency services network, and provides greater flexibility and control over processing of emergency calls by the emergency services authorities. This makes it easier to react quickly to unusual situations that require changes in how emergency calls are routed or handled (e.g., a natural disaster closes a PSAP), as well as ease in making long-term changes that affect such routing (e.g., cooperative agreements to specially handle calls requiring translation or relay services).

In an environment that uses ESInets, the originating network need only detect that the service URN of an emergency call is or starts with "sos", passing all types of emergency calls to an ESInet. The ESInet is then responsible for routing such calls to an appropriate PSAP. In an environment without an ESInet, the emergency services

authorities and the originating carriers would need to determine how such calls are routed.

9. Test Calls

This document uses [I-D.ietf-ecrit-ecall], which inherits the ability to utilize test call functionality from Section 15 of [RFC6881].

A service URN starting with "test." indicates a request for an automated test. Per [I-D.ietf-ecrit-ecall], "urn:service:test.sos.ecall.automatic" indicates such a test feature. This functionality is defined in [RFC6881].

Note that since test calls are placed using "test" as the parent service URN and "sos" as a child, such calls are not treated as an emergency call and so some functionality will not apply (such as preemption or service availability for devices lacking service ("non-service-initialized" or "NSI") if those are available for emergency calls); this is by design. MNOs may recognize test calls and treat them in a way that tests as much functionality as desired, but this is outside the scope of this document.

10. Example

Figure 7 shows an emergency call placed by a vehicle whereby location information and VEDS crash data are both attached to the SIP INVITE message. The INVITE has a request URI containing the 'urn:service:sos.ecall.automatic' service URN and is thus recognized as an ACN type of emergency call, and is also recognized as a type of emergency call because the request URI starts with 'urn:service:sos'. The mobile network operator (MNO) routes the call to an Emergency services IP Network (ESInet), as for any emergency call. The ESInet processes the call as an ACN and routes the call to an appropriate ACN-capable PSAP (using location information and the fact that that it is an ACN). (In deployments where there is no ESInet, the MNO itself needs to route directly to an appropriate ACN-capable PSAP.) The call is processed by the Emergency Services Routing Proxy (ESRP), as the entry point to the ESInet. The ESRP routes the call to an appropriate ACN-capable PSAP, where the call is received by a call taker.

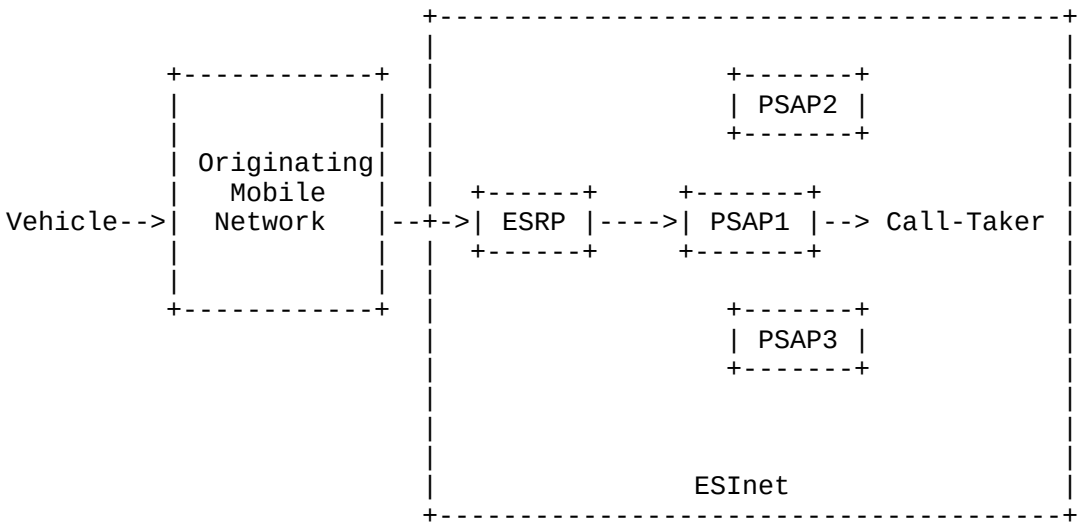


Figure 7: Example of Vehicle-Placed Emergency Call Message Flow

The example, shown in Figure 8, illustrates a SIP emergency call INVITE that is being conveyed with location information (a PIDF-LO) and crash data (as VEDS data).

```
INVITE urn:service:sos.ecall.automatic SIP/2.0
To: urn:service:sos.ecall.automatic
From: <sip:+13145551111@example.com>;tag=9fxced76sl
Call-ID: 3848276298220188511@atlanta.example.com
Geolocation: <cid:target123@example.com>
Geolocation-Routing: no
Call-Info: cid:1234567890@atlanta.example.com;
           purpose=EmergencyCallData.VEDS
Accept: application/sdp, application/pidf+xml
CSeq: 31862 INVITE
Content-Type: multipart/mixed; boundary=boundary1
Content-Length: ...

--boundary1

Content-Type: application/sdp

...Session Description Protocol (SDP) goes here

--boundary1

Content-Type: application/pidf+xml
Content-ID: <target123@atlanta.example.com>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<presence
       xmlns="urn:ietf:params:xml:ns:pidf"
       xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model"
       xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
       xmlns:dyn="urn:ietf:params:xml:ns:pidf:geopriv10:dynamic"
       xmlns:gml="http://www.opengis.net/gml"
       xmlns:gs="http://www.opengis.net/pidflo/1.0"
       entity="sip:+13145551111@example.com">
       <dm:device id="123">
           <gp:geopriv>
               <gp:location-info>
                   <gml:Point srsName="urn:ogc:def:crs:EPSG::4326">
                      <gml:pos>-34.407 150.883</gml:pos>
                   </gml:Point>
                    <dyn:Dynamic>
                       <dyn:heading>278</dyn:heading>
                       <dyn:direction><dyn:direction>
                    </dyn:Dynamic>
               </gp:location-info>
               <gp:usage-rules/>
               <method>gps</method>
           </gp:geopriv>
           <timestamp>2012-04-5T10:18:29Z</timestamp>
           <dm:deviceID>1M8GDM9A_KP042788</dm:deviceID>
       </dm:device>
</presence>

    --boundary1

Content-Type: application/EmergencyCallData.VEDS+xml
Content-ID: 1234567890@atlanta.example.com

...VEDS data object goes here

    --boundary1--
```

Figure 8: SIP INVITE indicating a Vehicule-Initated Emergency Call

11. Security Considerations

This document does not raise security considerations beyond those described in [RFC5069]. As with emergency service systems with end host provided location information there is the possibility that that location is incorrect, either intentially (in case of an a denial of service attack against the emergency services infrastructure) or due to a malfunctioning device. The reader is referred to

[I-D.ietf-ecrit-trustworthy-location] for a discussion of some of these vulnerabilities.

12. IANA Considerations

12.1. MIME Content-type Registration for 'application/EmergencyCall.VEDS+xml'

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

MIME media type name: application

MIME subtype name: EmergencyCallData.VEDS+xml

Mandatory parameters: none

Optional parameters: charset

Indicates the character encoding of enclosed XML.

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

Security considerations: This content type is designed to carry vehicle crash data during an emergency call. This data may contains personal information including vehicle VIN, location, direction, etc. appropriate precautions need to be taken to limit unauthorized access, inappropriate disclosure to third parties, and eavesdropping of this information. Please refer to Section 7 and Section 8 of [I-D.ietf-ecrit-additional-data] for more information.

Interoperability considerations: None

Published specification: [VEDS]

Applications which use this media type: Emergency Services

Additional information: None

Magic Number: None

File Extension: .xml

Macintosh file type code: 'TEXT'

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

Intended usage: LIMITED USE

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

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

12.2. Registration of the 'VEDS' entry in the Emergency Call Additional Data registry

This specification requests IANA to add the 'VEDS' entry to the Emergency Call Additional Data registry, with a reference to this document. The Emergency Call Additional Data registry has been established by [I-D.ietf-ecrit-additional-data].

13. Contributors

We would like to thank Ulrich Dietz for his help with earlier versions of the original version of this document.

14. Acknowledgements

We would like to thank Michael Montag, Arnoud van Wijk, Ban Al-Bakri, and Gunnar Hellstrom for their feedback.

15. Changes from Previous Versions

15.1. Changes from draft-ietf-01 to draft-ietf-02

- This document now refers to [I-D.ietf-ecrit-ecall] for technical aspects including the service URN; this document no longer proposes a unique service URN for non-eCall NG-ACN calls; the same service URN is now used for all NG-ACN calls including NG-eCall and non-eCall
- Added discussion of an NG-ACN call placed to a PSAP that doesn't support it
- Minor wording improvements and clarifications

15.2. Changes from draft-ietf-00 to draft-ietf-01

- Added further discussion of test calls
- Added further clarification to the document scope
- Mentioned that multi-region vehicles may need to support other crash notification specifications such as eCall
- Minor wording improvements and clarifications

15.3. Changes from draft-gellens-02 to draft-ietf-00

- Renamed from draft-gellens- to draft-ietf-
- Added text to Introduction to clarify that during a CS ACN, the PSAP call taker usually needs to listen to the data and transcribe it

15.4. Changes from draft-gellens-01 to -02

- Fixed case of 'EmergencyCallData', in accordance with changes to [I-D.ietf-ecrit-additional-data]

15.5. Changes from draft-gellens-00 to -01

- Now using 'EmergencyCallData' for purpose parameter values and MIME subtypes, in accordance with changes to [I-D.ietf-ecrit-additional-data]
- Added reference to RFC 6443
- Fixed bug that caused Figure captions to not appear

16. References

16.1. Normative References

[I-D.ietf-ecrit-additional-data]
Randy, R., Rosen, B., Tschofenig, H., Marshall, R., and J. Winterbottom, "Additional Data related to an Emergency Call", draft-ietf-ecrit-additional-data-24 (work in progress), October 2014.

[I-D.ietf-ecrit-ecall]
Gellens, R. and H. Tschofenig, "Next-Generation Pan-European eCall", draft-ietf-ecrit-ecall (work in progress), March 2015.

[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.

[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.

[RFC5031] Schulzrinne, H., "A Uniform Resource Name (URN) for Emergency and Other Well-Known Services", RFC 5031, January 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.

[RFC6443] Rosen, B., Schulzrinne, H., Polk, J., and A. Newton, "Framework for Emergency Calling Using Internet Multimedia", RFC 6443, December 2011.

[RFC6881] Rosen, B. and J. Polk, "Best Current Practice for Communications Services in Support of Emergency Calling", BCP 181, RFC 6881, March 2013.

[VEDS] "Vehicular Emergency Data Set (VEDS) version 3", July 2012, <http://apcointl.org/resources/aacn-and-veds/2012-07-25-19-24-06.html>.

16.2. Informative references

[I-D.ietf-ecrit-trustworthy-location] Tschofenig, H., Schulzrinne, H., and B. Aboba, "Trustworthy Location", draft-ietf-ecrit-trustworthy-location-14 (work in progress), July 2014.

[RFC5012] Schulzrinne, H. and R. Marshall, "Requirements for Emergency Context Resolution with Internet Technologies", RFC 5012, January 2008.

[RFC5069] Taylor, T., Tschofenig, H., Schulzrinne, H., and M. Shanmugam, "Security Threats and Requirements for Emergency Call Marking and Mapping", RFC 5069, January 2008.

Authors' Addresses

Randall Gellens
Qualcomm Technologies, Inc
5775 Morehouse Drive
San Diego 92651
US

Email: rg+ietf@qti.qualcomm.com

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

Email: br@brianrosen.net

Hannes Tschofenig

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

ECRIT
Internet-Draft
Intended status: Informational
Expires: January 5, 2016

R. Gellens
Qualcomm Technologies, Inc.
H. Tschofenig

July 4, 2015

Next-Generation Pan-European eCall
draft-ietf-ecrit-ecall-03.txt

Abstract

This document describes how to use IP-based emergency services mechanisms to support the next generation of the Pan European in-vehicle emergency call service defined under the eSafety initiative of the European Commission (generally referred to as "eCall"). eCall is a standardized and mandated system for a special form of emergency calls placed by vehicles. eCall deployment is required in the very near future in European Union member states, and eCall (and eCall-compatible systems) are also being deployed in other regions. eCall provides an integrated voice path and a standardized set of vehicle, sensor (e.g., crash related), and location data. An eCall is recognized and handled as a specialized form of emergency call and is routed to a specialized eCall-capable Public Safety Answering Point (PSAP) capable of processing the vehicle data and trained in handling emergency calls from vehicles.

Currently, eCall functions over circuit-switched cellular telephony; work on next-generation eCall (NG-eCall, sometimes called packet-switched eCall or PS-eCall) is now in process, and this document assists in that work by describing how to support eCall within the IP-based emergency services infrastructure.

This document also registers a MIME Content Type and an Emergency Call Additional Data Block for the eCall vehicle data.

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 5, 2016.

Table of Contents

1. Terminology

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

This document re-uses terminology defined in Section 3 of [RFC5012].

Additionally, we use the following abbreviations:

Term	Expansion
3GPP	3rd Generation Partnership Project
CEN	European Committee for Standardization
EENA	European Emergency Number Association
ESInet	Emergency Services IP network
IMS	Internet Multimedia Subsystem
IVS	In-Vehicle System
MNO	Mobile Network Operator
MSD	Minimum Set of Data
PSAP	Public Safety Answering Point

2. Document Scope

This document is limited to the signaling, data exchange, and protocol needs of next-generation eCall (NG-eCall, also referred to as packet-switched eCall (PS-eCall) and all-IP eCall) within the SIP framework for emergency calls, as described in [RFC6443] and [RFC6881]. eCall itself is specified by 3GPP and CEN and these specifications include far greater scope than is covered here.

The eCall service operates over cellular wireless communication, but this document does not address cellular-specific details, nor client domain selection (e.g., circuit-switched versus packet-switched). All such aspects are the purview of their respective standards bodies. The scope of this document is limited to eCall operating within a SIP-based environment (e.g., 3GPP IMS Emergency Calling).

The technical contents of this document may be suitable for use in other vehicle-initiated emergency call systems, but this is out of scope for this document.

Vehicles designed for multiple regions may need to support eCall and other Advanced Automatic Crash Notification systems, such as described in [draft-ietf-ecrit-car-crash]. That system is compatible with eCall, differing primarily in the specific data set that is sent.

3. Introduction

Emergency calls made from vehicles (e.g., in the event of a crash) assist in significantly reducing road deaths and injuries by allowing emergency services to be aware of the incident, the state of the vehicle, the location of the vehicle, and to have a voice channel with the vehicle occupants. This enables a quick and appropriate response.

The European Commission initiative of eCall was conceived in the late 1990s, and has evolved to a European Parliament decision requiring the implementation of compliant in-vehicle systems (IVS) in new vehicles and the deployment of eCall in the European Member States in the very near future. eCall (and eCall-compatible systems) are also being adopted in other regions.

The pan-European eCall system provides a standardized and mandated mechanism for emergency calls by vehicles. eCall establishes procedures for such calls to be placed by in-vehicle systems, recognized and processed by the network, and routed to a specialized PSAP where the vehicle data is available to assist the call taker in assessing and responding to the situation. eCall provides a standard set of vehicle, sensor (e.g., crash related), and location data.

An eCall may be either user-initiated or automatically triggered. Automatically triggered eCalls indicate a car crash or some other serious incident and carry a greater presumption of risk of injury. Manually triggered eCalls may be reports of serious hazards and are likely to require a different response than an automatically triggered eCall. Manually triggered eCalls are also more likely to be false (e.g., accidental) calls and may thus be subject to different handling by the PSAP.

Currently, eCall is standardized (by 3GPP [SDO-3GPP] and CEN [CEN]) as a 3GPP circuit-switched call over GSM (2G) or UMTS (3G). Flags in the call setup mark the call as an eCall, and further indicate if the call was automatically or manually triggered. The call is routed to an eCall-capable PSAP, a voice channel is established between the vehicle and the PSAP, and an eCall in-band modem is used to carry a defined set of vehicle, sensor (e.g., crash related), and location data (the Minimum Set of Data or MSD) within the voice channel. The same in-band mechanism is used for the PSAP to acknowledge successful receipt of the MSD, and to request the vehicle to send a new MSD (e.g., to check if the state of or location of the vehicle or its occupants has changed). Work on next-generation eCall (NG-eCall, also referred to as packet-switched eCall or PS eCall) is now in process. As part of this work, the European Telecommunications Standards Institute (ETSI) [SDO-ETSI] has published a Technical Report titled "Mobile Standards Group (MSG); eCall for VoIP" [MSG_TR] that presents findings and recommendations regarding support for eCall in an all-IP environment. NG-eCall moves from circuit switched to all-IP, and carries the vehicle data and other eCall-specific data as additional data associated with the call. This document describes how IETF mechanisms for IP-based emergency calls, including [RFC6443] and [additional-data-draft] are used to provide the signaling and data exchange of the next generation of pan-European eCall.

The [MSG_TR] recommendation for NG-eCall is to use 3GPP IMS emergency calling with additional elements identifying the call as an eCall and as carrying eCall data and with mechanisms for carrying the data. 3GPP IMS emergency services support multimedia, providing the ability to carry voice, text, and video. This capability is referred to within 3GPP as Multimedia Emergency Services (MMES).

A transition period will exist during which time the various entities involved in initiating and handling an eCall might support next-generation eCall, legacy eCall, or both. This transition period might last several years or longer. The issue of migration/co-existence during the transition period is very important but is outside the scope of this document. The ETSI TR "Mobile Standards Group (MSG); eCall for VoIP" [MSG_TR] discusses these issues in Clause 7.

4. eCall Requirements

Overall eCall requirements are specified by CEN in [EN_16072] and by 3GPP in [TS22.101] clauses 10.7 and A.27. Requirements specific to vehicle data are contained in EN 15722 [msd]. For convenience, the requirements most applicable to the limited scope of this document are summarized very briefly below.

eCall requires:

- The call be recognized as an eCall (which is inherently an emergency call)
- The call setup indicates if the call was manually or automatically triggered
- A voice channel between the vehicle and the PSAP
- Carrying the MSD intrinsically with the call (the MSD needs to be available to the same call-taker as the voice)
- The ability for the PSAP to acknowledge receipt of the MSD
- The ability for the PSAP to request that the vehicle generate and transmit a new MSD
- The ability of the PSAP to be able to re-contact the occupants of vehicle after the initial eCall is concluded
- The ability to perform a test call (which may be routed to a PSAP but is not treated as an emergency call and not handled by a call taker)

It is recognized that NG-eCall offers many potential enhancements, although these are not required by current EU regulations. For convenience, the enhancements most applicable to the limited scope of this document are summarized very briefly below.

NG-eCall is expected to offer:

- The ability to carry more data (e.g., an enhanced MSD or an MSD plus additional sets of data)
- The ability to handle video
- The ability to handle text
- The ability for the PSAP to access vehicle components (e.g., an onboard camera (such as rear facing or blind-spot cameras) for a visual assessment of the crash site situation)
- The ability for the PSAP to request the vehicle to take actions (e.g., sound the horn, disable the ignition, lock/unlock doors)
- The ability to avoid audio muting of the voice channel (because the MSD is not transferred using an in-band modem)

5. Vehicle Data

Pan-European eCall provides a standardized and mandated set of vehicle related data, known as the Minimum Set of Data (MSD). The European Committee for Standardization (CEN) has specified this data in EN 15722 [msd], along with both ASN.1 and XML encodings for the MSD [msd]. Circuit-switched eCall uses the ASN.1 encoding. The XML encoding is better suited for use in SIP messages and is used in this document. (The ASN.1 encoding is specified in Annex A of EN 15722 [msd], while the XML encoding is specified in Annex C.)

The "Additional Data related to an Emergency Call" document [additional-data-draft] establishes a general mechanism for attaching blocks of data to a SIP emergency call. This document makes use of that mechanism to carry the eCall MSD in a SIP emergency call.

This document registers the 'application/ emergencyCallData.eCall.MSD+xml' MIME Content-Type to enable the MSD to be carried in SIP. This document also adds the 'eCall.MSD' entry to the Emergency Call Additional Data Blocks registry (established by [additional-data-draft]) to enable the MSD to be recognized as such in a SIP-based eCall emergency call.

Note that if additional data sets are defined and registered (e.g., in the future or in other regions) and transmitted using the same mechanisms, the size and frequency of transmission during a session needs to be evaluated to be sure it is appropriate to use the signaling channel.

6. Call Setup

In circuit-switched eCall, the IVS places a special form of a 112 emergency call which carries an eCall flag (indicating that the call is an eCall and also if the call was manually or automatically triggered); the mobile network operator (MNO) recognizes the eCall flag and routes the call to an eCall-capable PSAP; vehicle data is

transmitted to the PSAP via the eCall in-band modem (in the voice channel).

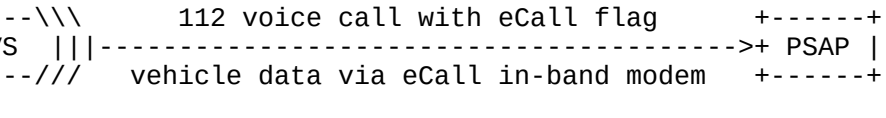


Figure 1: circuit-switched eCall

An In-Vehicle System (IVS) which supports NG-eCall transmits the MSD in accordance with [additional-data-draft] by encoding it as specified (per Appendix C of EN 15722 [msd]) and attaching it to an INVITE as a MIME body part. The body part is identified by its MIME content-type ('application/emergencyCallData.eCall.MSD+xml') in the Content-Type header field of the body part. The body part is assigned a unique identifier which is listed in a Content-ID header field in the body part. The INVITE is marked as containing the MSD by adding (or appending to) a Call-Info header field at the top level of the INVITE. This Call-Info header field contains a CID URL referencing the body part's unique identifier, and a 'purpose' parameter identifying the data as the eCall MSD per the registry entry; the 'purpose' parameter's value is 'emergencyCallData.' and the root of the MIME type (not including the 'emergencyCallData' prefix and any suffix such as '+xml' (e.g., 'purpose=emergencyCallData.eCall.MSD').

For NG-eCall, the IVS establishes an emergency call using the 3GPP IMS solution with a Request-URI indicating an eCall type of emergency call and with vehicle data attached; the MNO or ESInet recognizes the eCall URN and routes the call to a NG-eCall capable PSAP; the PSAP interpets the vehicle data sent with the call and makes it available to the call taker.

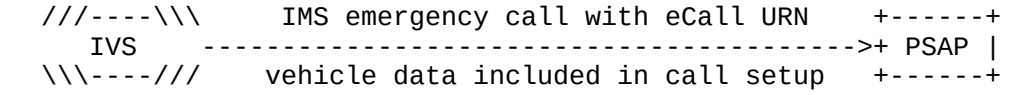


Figure 2: NG-eCall

This document registers new service URN children within the "sos" subservice. These URNs provide the mechanism by which an eCall is identified, and differentiate between manually and automatically triggered eCalls (which may be subject to different treatment, depending on policy). The two service URNs are: urn:service:sos.ecall.automatic and urn:service:sos.ecall.manual

7. Call Routing

The routing rules for eCalls are likely to differ from those of other emergency calls because eCalls are special types of emergency calls (with implications for the types of response required) and need to be handled by specially designated PSAPs. In an environment that uses ESInets, the originating network passes all types of emergency calls to an ESInet (which have a request URI containing the "SOS" service URN). The ESInet is then responsible for routing such calls to the appropriate PSAP. In an environment without an ESInet, the emergency services authorities and the originating network jointly determine how such calls are routed.

7.1. ESInets

This section provides background information on ESInets for information only.

An Emergency Services IP Network (ESInet) is a network operated by emergency services authorities. It handles emergency call routing and processing before delivery to a PSAP. In the NG1-1-2 architecture adopted by EENA, each PSAP is connected to one or more ESInets. Each originating network is also connected to one or more ESInets. The ESInets maintain policy-based routing rules which control the routing and processing of emergency calls. The centralization of such rules within ESInets provides for a cleaner separation between the responsibilities of the originating network and that of the emergency services network, and provides greater flexibility and control over processing of emergency calls by the emergency services authorities. This makes it easier to react quickly to unusual situations that require changes in how emergency calls are routed or handled (e.g., a natural disaster closes a PSAP), as well as ease in making long-term changes that affect such routing (e.g., cooperative agreements to specially handle calls requiring translation or relay services). ESInets may support the ability to interwork NG-eCall to legacy eCall to handle eCall-capable PSAPs that are not IP PSAPs (similarly to the ability to interwork IP emergency calls to legacy non-IP PSAPs). Note that in order to support legacy eCall-capable PSAPs that are not IP PSAPs and are not attached to an ESInet, an originating network may need the ability to route an eCall itself (e.g., to an interworking facility with interconnection to a suitable legacy eCall capable PSAP) based on the eCall and manual or automatic indications. The ETSI TR "Mobile Standards Group (MSG); eCall for VoIP" [MSG_TR] discusses transition issues in Clause 7.

8. Test Calls

eCall requires the ability to place test calls. These are calls that are recognized and treated to some extent as eCalls but are not given emergency call treatment and are not handled by call takers. The test call facility allows the IVS or user to verify that an eCall can be successfully established with voice communication. The IVS can also verify that the MSD was successfully received.

A service URN starting with "test." indicates a test call. For eCall, "urn:service:test.sos.ecall" indicates such a test feature. This functionality is defined in [RFC6881].

This document registers "urn:service:test.sos.ecall" for eCall test calls.

the current eCall test call facility is a non-emergency number so does not get treated as an emergency call. MNOs may treat a vehicle call in the "test" service URN in a way that tests as much functionality as desired, but this is outside the scope of this document.

PSAPs that have the ability to process NG-eCalls SHOULD accept test calls and send an acknowledgment if the MSD was successfully received, per this document. Such PSAPs MAY also play an audio clip (for example, saying that the call reached a PSAP) in addition to supporting media loopback per [RFC6881].

9. eCall-Specific Control/Metadata

eCall requires the ability for the PSAP to acknowledge successful receipt of an MSD sent by the IVS, and for the PSAP to request that the IVS send an MSD (e.g., the call taker may initiate a request for a new MSD to see if the vehicle’s state or location has changed). Future enhancements are desired to enable the PSAP to send other requests to the vehicle, such as locking or unlocking doors, sounding the horn, flashing the lights, starting a video stream from on-board cameras (such as rear focus or blind-spot), etc.

The mechanism established in [additional-data-draft], used in Section 5 of this document to carry the MSD from the IVS to the PSAP, is also used to carry a block of control data from the PSAP to the IVS. This eCall control block (sometimes referred to as eCall metadata) is an XML structure containing eCall-specific elements. When the PSAP needs to send an eCall control block that is in response to the MSD or other data sent by the IVS in a SIP request, the control block can be sent in the SIP response to that request (e.g., the INVITE). When the PSAP needs to send an eCall control

block that is not an immediate response to an MSD or other data sent by the IVS, the control block can be transmitted from the PSAP to the IVS in a SIP INFO message within the established session. The IVS can then send any requested data (such as a new MSD) in the reply to the INFO message. This mechanism flexibly allows the PSAP to send eCall-specific data to the IVS and the IVS to respond. If control data sent in a response message requests the IVS to send a new MSD or other data block, or to perform an action other than sending data, the IVS can send the requested data or an acknowledgment regarding the action in an INFO message within the session (it could also use re-INVITE but that is unnecessary when no aspect of the session or media is changing).

This mechanism requires

- An XML definition of the eCall control object
- An extension mechanism by which new elements can be added to the control object definition (e.g., permitting additional elements to be included by adding their namespace)
- A MIME type registration for the control object (so it can be carried in SIP messages and responses)
- An entry in the Emergency Call Additional Data Blocks sub-registry (established by [additional-data-draft]) so that the control block can be recognized as emergency call specific data within the SIP messages
- An Info-Package registration per [RFC6086] permitting the control block within Info messages

9.1. The eCall Control Block

The eCall control block is an XML data structure allowing for acknowledgments, requests, and capabilities information. It is carried in a SIP body part with a specific MIME content type. Three top-level elements are defined for use within an eCall control block:

ack Used in a control block sent by either side. The PSAP uses this to acknowledge receipt of data set sent by the IVS. The IVS uses this to acknowledge receipt of a request by the PSAP when that request would not otherwise be acknowledged (if the PSAP requests the vehicle to send data and the vehicle does so, the data serves as a success acknowledgement).

capabilities: Used in a control block sent from the IVS to the PSAP (e.g., in the initial INVITE) to inform the PSAP of the vehicle capabilities. Child elements contain all actions and data types supported by the vehicle and all available lamps (lights) and cameras.

request Used in a control block sent by the PSAP to the IVS, to request the vehicle to perform an action.

Mandatory Actions (the IVS and the PSAP MUST support):

- Transmit data object

Optional Actions (the IVS and the PSAP MAY support):

- Play and/or display static (pre-defined) message
- Speak/display dynamic text (text supplied in action)
- Flash or turn on or off a lamp (light)
- Honk horn
- Enable a camera

The <ack> element indicates the object being acknowledged (i.e., a data object or a <request> element), and reports success or failure.

The <capabilities> element has child <request> elements to indicate the actions supported by the IVS.

The <request> element contains attributes to indicate the request and to supply any needed information, and MAY contain a <text> child element to contain the text for a dynamic message. The 'action' attribute is mandatory and indicates the specific action. An IANA registry is created in Section 13.8.1 to contain the allowed values.

Extensibility: New elements, child elements, and attributes can be defined in new namespaces. IANA registries are used to specify the permitted values of several elements and attributes. These mechanisms allow for extension.

The control block does not contain a 'request' action to play dynamic media (such as a pre-recorded audio message). The SIP re-INVITE mechanism can be used to establish a one-way media stream for this purpose.

9.1.1. The <ack> element

The <ack> element is transmitted by the PSAP to acknowledge receipt of an eCall data object. An <ack> element sent by a PSAP references the unique ID of the data object that was sent by the IVS, and further indicates if the PSAP considers the receipt successful or not. The <ack> element is also transmitted by the IVS to the PSAP to acknowledge receipt of a <request> element that requested the IVS to perform an action other than transmitting a data object (e.g., a request to display a message would be acknowledged, but a request to transmit a data object would not result in a separate <ack> element

being sent, since the data object itself serves as acknowledgment.) An <ack> element sent by an IVS references the unique ID of the request being acknowledged, indicates whether the request was successfully performed, and if not, optionally includes an explanation.

The <ack> element has the following attributes and child elements:

9.1.1.1. Attributes of the <ack> element

The <ack> element has the following attributes:

Name: ref
Usage: Mandatory
Type: anyURI
Description: References the Content-ID of the body part that contained the data object or control object being acknowledged.
Example: <ack received="yes" ref="1234567890@atlanta.example.com"/>

Name: received
Usage: Conditional: mandatory in an >ack< element sent by a PSAP; not applicable in an >ack< element sent by an IVS
Type: Boolean
Description: Indicates if the referenced object was successfully received or not
Example: <ack received="yes" ref="1234567890@atlanta.example.com"/>

9.1.1.2. Child Elements of the <ack> element

The <ack> element has the following child elements:

Name: actionResult
Usage: Optional
Description: An <actionResult> element indicates the result of an action (other than a 'send-data' action). It has the following attributes:

Name: action
Usage: Mandatory
Type: token
Description: Contains the value of the 'action' attribute of the <request> element

Name: success
Usage: Mandatory
Type: Boolean

Description: Indicates if the action was successfully accomplished

Name: reason
Usage: Conditional
Type: token
Description: Used when 'success' is "False", this attribute contains a reason code for a failure. A registry for reason codes is defined in Section 13.8.3.

Name: details
Usage: optional
Type: string
Description: Contains further explanation of the circumstances of a success or failure. The contents are implementation-specific and human-readable.

Example: <actionResult action="msg-dynamic" success="true"/>

Example: <actionResult action="lamp" success="false" reason="unable" details="The requested lamp is inoperable"/>

9.1.1.3. Ack Examples

```
<?xml version="1.0" encoding="UTF-8"?>
<EmergencyCallData.eCallControl
    xmlns="urn:ietf:params:xml:ns:EmergencyCallData:eCall-control"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:ietf:params:xml:ns:EmergencyCallData:
        eCall-control">

<ack received="true" ref="1234567890@atlanta.example.com"/>

</EmergencyCallData.eCallControl>
```

Figure 3: Ack Example from PSAP to IVS

```
<?xml version="1.0" encoding="UTF-8"?>
<EmergencyCallData.eCallControl
    xmlns="urn:ietf:params:xml:ns:EmergencyCallData:eCall-control"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:ietf:params:xml:ns:EmergencyCallData:
        eCall-control">

<ack ref="1234567890@atlanta.example.com">
    <actionResult action="msg-dynamic" success="true"/>
    <actionResult action="lamp" success="false" reason="unable"
                  details="The requested lamp is inoperable"/>
</ack>

</EmergencyCallData.eCallControl>
```

Figure 4: Ack Example from IVS to PSAP

9.1.2. The <capabilities> element

The <capabilities> element is transmitted by the IVS to indicate to the PSAP its capabilities. No attributes for this element are currently defined. The following child elements are defined:

9.1.2.1. Child Elements of the <capabilities> element

The <capabilities> element has the following child elements:

Name: request
Usage: Mandatory
Description: The <capabilities> element contains a <request> child element per action supported by the vehicle.

Because support for a 'send-data' action is REQUIRED, a <request> child element with a "send-data" 'action' attribute is also REQUIRED. The 'supported-datatypes' attribute is REQUIRED in this <request> element within a <capabilities> element, and MUST contain at a minimum the 'eCall.MSD' data block value; it SHOULD contain all data blocks supported by the IVS.

All other actions are OPTIONAL.

If the "msg-static" action is supported, a <request> child element with a "msg-static" 'action' attribute is sent, with a 'msgid' attribute set to the highest supported static message supported by the vehicle.

If the "lamp" action is supported, a <request> child element with a "lamp" 'action' is sent, with a 'supported-lamps' attribute set to all supported lamp IDs.

If the "enable-camera" action is supported, a <request> child element with an "enable-camera" 'action' is sent, with a 'supported-cameras' attribute set to all supported camera IDs.

Examples:

```
<request action="send-data" supported-datatypes="eCall.MSD" />
<request action="send-data" supported-datatypes="eCall.MSD; VEDS;
eCall.type2" />
<request action="msg-dynamic"/>
<request action="msg.static" msgid="17" />
```

9.1.2.2. Capabilities Example

```
<?xml version="1.0" encoding="UTF-8"?>
<EmergencyCallData.eCallControl
    xmlns="urn:ietf:params:xml:ns:EmergencyCallData:eCall-control"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:ietf:params:xml:ns:EmergencyCallData:
        eCall-control">

<capabilities>
    <request action="send-data" supported-datatypes="eCall.MSD"/>
    <request action="lamp"
             supported-lamps="head;interior;fog-front;fog-rear;brake;
             position-front;position-rear;turn-left;turn-right;hazard"/>
    <request action="msg-static" msgid="3"/>
    <request action="msg-dynamic"/>
    <request action="honk"/>
    <request action="enable-camera" supported-cameras="backup; interior"/>
</capabilities>

</EmergencyCallData.eCallControl>
```

Figure 5: Capabilities Example

9.1.3. The <request> element

A <request> element appears one or more times on its own or as a child of a <capabilities> element. The following attributes and child elements may be used:

9.1.3.1. Attributes of the <request> element

The <request> element has the following attributes:

Name: action
Usage: Mandatory
Type: token
Description: Identifies the action that the vehicle is requested to perform. An IANA registry is established in Section 13.8.1 to contain the allowed values.
Example: action="send-data"

Name: msgid
Usage: Conditional
Type: int
Description: Mandatory with a "msg-static" action. Indicates the identifier of the static message to be displayed and/or spoken for the vehicle occupants. This document established an IANA registry for messages and their IDs, in Section 13.8.2
Example: msgid="3"

Name: persistance
Usage: Optional
Type: duration
Description: Specifies how long to carry on the specified action, for example, how long to continue honking or flashing. If absent, the default is indefinitely.
Example: persistance="PT1H"

Name: datatype
Usage: Conditional
Type: token
Description: Mandatory with a "send-data" action. Specifies the data block that the IVS is requested to transmit, using the same identifier as in the ’purpose’ attribute set in a Call-Info header field to point to the data block. Permitted values are contained in the ’Emergency Call Data Types’ IANA registry established in [additional-data-draft].
Example: datatype="eCall.MSD"

Name: supported-datatypes
Usage: Conditional
Type: string
Description: Used with a ’send-data’ action in a <request> element that is a child of a <capability> element, this attribute lists all data blocks that the vehicle can transmit, using the same identifier as in the ’purpose’ attribute in a Call-Info header field to point to the data block. Permitted values are contained

in the 'Emergency Call Data Types' IANA registry established in [additional-data-draft]. Multiple values are separated with a semicolon.
Example: supported-datatypes="eCall.MSD; VEDS; eCall.foo"

Name: lamp-action
Usage: Conditional
Type: token
Description: Used with a 'lamp' action, indicates if the lamp should be illuminated, turned off, or flashed. Permitted values are 'on', 'off', and 'flash'.
Example: lamp-action="flash"

Name: lamp-ID
Usage: Conditional
Type: token
Description: Used with a 'lamp' action, indicates which lamp the action affects. Permitted values are contained in the registry of lamp-ID tokens created in Section 13.8.4
Example: lamp-ID="hazard"

Name: supported-lamps
Usage: Conditional
Type: string
Description: Used with a 'lamp' action in a <request> element that is a child of a <capability> element, this attribute lists all supported lamps, using values in the registry of lamp-ID tokens created in Section 13.8.4. Multiple values are separated with a semicolon.
Example: supported-lamps="head; interior; fog-front; fog-rear; brake; position-front; position-rear; turn-left; turn-right; hazard"

Name: camera-ID
Usage: Conditional
Type: token
Description: Used with an 'enable-camera' action, indicates which camera to enable. Permitted values are contained in the registry of camera-ID tokens created in Section 13.8.5. When a vehicle camera is enabled, the IVS sends a re-INVITE to negotiate a one-way media stream for the camera.
Example: camera-ID="backup"

Name: supported-cameras
Usage: Conditional
Type: string
Description: Used with an 'enable-camera' action in a <request> element that is a child of a <capability> element, this attribute

lists all cameras that the vehicle supports (can add as a video feed in the current session), using the same identifiers as are used in the 'camera-ID' attribute (contained in the camera ID registry in Section 13.8.5). Multiple values are separated with a semicolon.

Example: supported-cameras="backup; interior"

9.1.3.2. Child Elements of the <request> element

The <request> element has the following child elements:

Name: text
Usage: Conditional
Type: string
Description: Used within a <request action="msg-dynamic"> element to contain the text to be displayed and/or spoken (via text-to-speech) for the vehicle occupants.
Example: <text>Emergency authorities are aware of your incident and location. Due to a multi-vehicle incident in your area, no one is able to speak with you right now. Please remain calm. We will assist you soon.</text>

9.1.3.3. Request Example

```
<?xml version="1.0" encoding="UTF-8"?>
<EmergencyCallData.eCallControl
    xmlns="urn:ietf:params:xml:ns:EmergencyCallData:eCall-control"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:ietf:params:xml:ns:EmergencyCallData:
        eCall-control">

<request action="send-data" datatype="eCall.MSD"/>
<request action="lamp" lamp-id="hazard"
         lamp-action="flash" persistance="PT1H"/>
<request action="msg-static" msgid="1"/>
<request action="msg-dynamic">
    <text>Remain calm.  Help is on the way.</text>
</request>

</EmergencyCallData.eCallControl>
```

Figure 6: Request Example

9.2. The emergencyCallData.eCall INFO package

This document registers the 'emergencyCallData.eCall' INFO package. Both endpoints (the IVS and the PSAP equipment) set the Recv-Info header field to 'emergencyCallData.eCall' per [RFC6086] to indicate ability to receive INFO messages carrying eCall data or control blocks.

Support for the 'emergencyCallData.eCall' INFO package indicates the ability to receive eCall data and control blocks, which are carried in a body part whose subtype starts with 'emergencyCallData.eCall.'. At present there is only one defined eCall data block, which has the 'application/emergencyCallData.eCall.MSD+xml' MIME type, and one eCall control block, which has the 'application/ emergencyCallData.eCall.control+xml' MIME type. The eCall control block includes the ability for the IVS to indicate its capabilities, so in the event additional eCall blocks are defined, the IVS can indicate which it supports.

The use of INFO is based on an analysis of the requirements against the intent and effects of INFO versus other approaches (such as SIP MESSAGE, media plane, or non-SIP protocols). In particular, the transport of eCall data and control blocks is done only during an emergency session established with SIP, using the mechanism established in [additional-data-draft], and is normally carried in the initial INVITE and its response; the use of INFO only occurs when a data block or request needs to be sent subsequently during the call. While MESSAGE could be used, it is not tied to a SIP session as is INFO. REINVITE could also be used, but is normally used to modify the session. SUBSCRIBE/NOTIFY could be coerced into service, but the semantics are not a clean fit. Hence, INFO is appropriate.

An INFO request message carrying an eCall data or control block has an Info-Package header field set to 'emergencyCallData.eCall' per [RFC6086]. The INFO request message is marked as containing the eCall data or control block by a Call-Info header field containing a CID URL referencing the unique identifier of the body part containing the eCall data or control, and a 'purpose' parameter identifying the block. Because the eCall data or control block is being carried in an INFO request message, the body part also carries a Content-Disposition header field set to "Info-Package".

Per [additional-data-draft], emergency call related additional data MAY be included in any SIP request or response message that may contain a body. Hence, notwithstanding Section 4.3.2. of [RFC6086], INFO response messages MAY contain eCall data or control blocks, provided they are included as described in this document (with a Call-Info header field containing a CID URL referencing the unique

identifier of the body part, and a 'purpose' parameter identifying the block). When eCall data or control blocks are included in an INFO response message, this is done per [additional-data-draft] and this document, and not under [RFC6086]; that is, they are included as emergency call additional data, not as an INFO package associated data.

10. Examples

Figure 7 shows an eCall. The call uses the request URI 'urn:service:sos.ecall.automatic' service URN and is recognized as an eCall, and further as one that was invoked automatically by the IVS due to a crash or other serious incident. In this example, the originating network routes the call to an ESInet (as for any emergency call in an environment with an ESInet). The ESInet routes the call to the appropriate NG-eCall capable PSAP. The emergency call is received by the ESInet's Emergency Services Routing Proxy (ESRP), as the entry point into the ESInet. The ESRP routes the call to a PSAP, where it is received by a call taker. In deployments where there is no ESInet, the originating network routes the call directly to the appropriate NG-eCall capable PSAP.

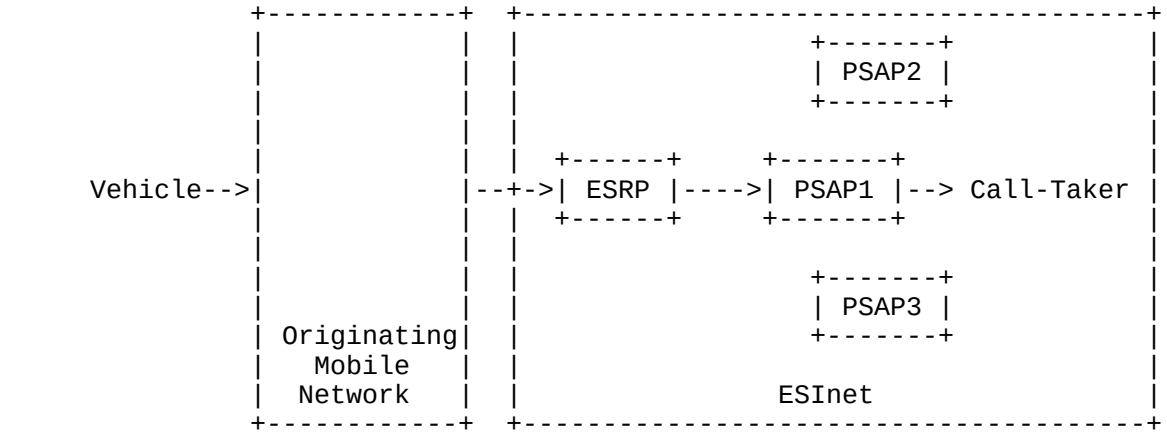


Figure 7: Example of NG-eCall Message Flow

The example, shown in Figure 8, illustrates a SIP eCall INVITE that contains an MSD and an eCall control block with vehicle capabilities. For simplicity, the example does not show all SIP headers, nor does it show the additional data blocks added by the IVS and the originating mobile network.

```
INVITE urn:service:sos.ecall.automatic SIP/2.0
```

```
    To: urn:service:sos.ecall.automatic
    From: <sip:+13145551111@example.com>;tag=9fxced76sl
    Call-ID: 3848276298220188511@atlanta.example.com
    Geolocation: <cid:target123@example.com>
    Geolocation-Routing: no
    Call-Info: cid:1234567890@atlanta.example.com;
               purpose=emergencyCallData.eCall.MSD;
               cid:2345678901@atlanta.example.com;
               purpose=emergencyCallData.eCall.control;
    Accept: application/sdp, application/pidf+xml,
            application/emergencyCallData.eCall.control
    CSeq: 31862 INVITE
    Recv-Info: emergencyCallData.eCall
    Content-Type: multipart/mixed; boundary=boundary1
    Content-Length: ...

    --boundary1

    Content-Type: application/sdp

         ...Session Description Protocol (SDP) goes here...

    --boundary1

    Content-Type: application/emergencyCallData.eCall.MSD+xml
    Content-ID: 1234567890@atlanta.example.com

    <ECallMessage>
       <id>1</id>

       <msd>
          <msdStructure>

             <messageIdentifier>1</messageIdentifier>

             <control>
                <automaticActivation> <true/> </automaticActivation>
                <testCall> <false/> </testCall>
                <positionCanBeTrusted> <true/> </positionCanBeTrusted>
                <vehicleType> <passengerVehicleClassM1/> </vehicleType>
             </control>

             <vehicleIdentificationNumber>
                <isowmi>WMI</isowmi>
                <isovds>VDSVDS</isovds>
                <isovisModelyear>Y</isovisModelyear>
                <isovisSeqPlant>A123456</isovisSeqPlant>
             </vehicleIdentificationNumber>
```

```
            <vehiclePropulsionStorageType>
               <gasolineTankPresent> <true/> </gasolineTankPresent>
               <electricEnergyStorage> <true/> </electricEnergyStorage>
            </vehiclePropulsionStorageType>

            <timestamp>123456789</timestamp>

            <vehicleLocation>
               <positionLatitude>173881200</positionLatitude>
               <positionLongitude>41822520</positionLongitude>
            </vehicleLocation>

            <vehicleDirection>14</vehicleDirection>

            <recentVehicleLocationN1>
               <latitudeDelta>10</latitudeDelta>
               <longitudeDelta>-10</longitudeDelta>
            </recentVehicleLocationN1>

            <recentVehicleLocationN2>
               <latitudeDelta>10</latitudeDelta>
               <longitudeDelta>-20</longitudeDelta>
            </recentVehicleLocationN2>

            <numberOfPassengers>2</numberOfPassengers>

         </msdStructure>

         <optionalAdditionalData>
            <oid>1.2.125</oid>
            <data>30304646</data>
         </optionalAdditionalData>
      </msd>
   </ECallMessage>

   --boundary1

   Content-Type: application/emergencyCallData.eCall.control+xml
   Content-ID: 2345678901@atlanta.example.com

    <?xml version="1.0" encoding="UTF-8"?>
    <EmergencyCallData.eCallControl
        xmlns="urn:ietf:params:xml:ns:EmergencyCallData:eCall-control"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="urn:ietf:params:xml:ns:EmergencyCallData:
            eCall-control">

    <capabilities>
```

```
        <request action="send-data" supported-datatypes="eCall.MSD"/>
        <request action="lamp"
                 supported-lamps="head;interior;fog-front;fog-rear;
                 brake;position-front;position-rear;turn-left;
                 turn-right;hazard"/>
        <request action="msg-static" msgid="3"/>
        <request action="msg-dynamic"/>
        <request action="honk"/>
        <request action="enable-camera"
                 supported-cameras="backup; interior"/>
    </capabilities>

    </EmergencyCallData.eCallControl>

    --boundary1--
```

Figure 8: SIP NG-eCall INVITE

Continuing the example, Figure 9 illustrates a SIP 200 OK response to the INVITE of Figure 8, containing an eCall control block acknowledging successful receipt of the eCall MSD. (For simplicity, the example does not show all SIP headers.)

```
SIP/2.0 200 OK
To: <sip:+13145551111@example.com>;tag=9fxced76sl
From: Exemplar PSAP <urn:service:sos.ecall.automatic>
Call-ID: 3848276298220188511@atlanta.example.com
Call-Info: cid:2345678901@atlanta.example.com;
           purpose=emergencyCallData.eCall.control;
Accept: application/sdp, application/pidf+xml,
        application/emergencyCallData.eCall.control,
        application/emergencyCallData.eCall.MSD
CSeq: 31862 INVITE
Recv-Info: emergencyCallData.eCall
Content-Type: multipart/mixed; boundary=boundaryX
Content-Length: ...

--boundaryX

Content-Type: application/sdp

     ...Session Description Protocol (SDP) goes here...

--boundaryX

<?xml version="1.0" encoding="UTF-8"?>
<EmergencyCallData.eCallControl
    xmlns="urn:ietf:params:xml:ns:EmergencyCallData:eCall-control"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:ietf:params:xml:ns:EmergencyCallData:
        eCall-control">

<ack received="true" ref="1234567890@atlanta.example.com"/>

</EmergencyCallData.eCallControl>

--boundaryX
```

Figure 9: 200 OK response to INVITE

11. Security Considerations

The security considerations described in [RFC5069] apply here.

An eCall will carry two forms of location data: the network-provided location that is inherently part of IMS emergency calls (which might be determined solely by the network, or in cooperation with or possibly entirely by the originating device), and the IVS-supplied location within the MSD. This is likely to be useful to the PSAP, especially when the two locations are independently determined. Even

in situations where the network-supplied location is limited to the cell site, this can be useful as a sanity check on the device-supplied location contained in the MSD.

The document [I-D.ietf-ecrit-trustworthy-location] discusses trust issues regarding location provided by or determined in cooperation with end devices.

The mechanism by which the PSAP sends acknowledgments and requests to the vehicle requires authenticity considerations; when the PSAP request is received within a session initiated by the vehicle as an eCall emergency call placed over a cellular network, there is a higher degree of trust that the source is indeed a PSAP. If the PSAP request is received in other situations, such as a call-back, the trust issues in verifying that a call-back is indeed from a PSAP are more complex (see the PSAP Callback document [RFC7090]). A further safeguard (applicable regardless of which end initiated the call and the means of the call) is for the PSAP or emergency service provider to sign the body part using a certificate issued by a known emergency services certificate authority and for which the IVS can verify the root certificate.

12. XML Schema

This section defines the XML schema of the eCall control block. (The schema for the MSD can be found in EN 15722 [msd].)

```
<?xml version="1.0"?>
<xs:schema
    targetNamespace="urn:ietf:params:xml:ns:EmergencyCallData:eCall-control"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:pi="urn:ietf:params:xml:ns:EmergencyCallData:eCall-control"
    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/2009/01/xml.xsd"/>

    <xs:element name="EmergencyCallData.eCallControl"
                type="pi:eCallControlType"/>


    <xs:complexType name="eCallControlType">
        <xs:complexContent>
            <xs:restriction base="xs:anyType">
                <xs:choice>
```

```
                    <xs:element name="capabilities"
                                type="pi:capabilitiesType"/>
                    <xs:element name="request" type="pi:requestType"/>
                    <xs:element name="ack" type="pi:ackType"/>
                    <xs:any namespace="##other" processContents="lax"
                            minOccurs="0"
                            maxOccurs="unbounded"/>
                </xs:choice>
                <xs:anyAttribute/>
            </xs:restriction>
        </xs:complexContent>
    </xs:complexType>


    <xs:complexType name="ackType">
        <xs:complexContent>
            <xs:restriction base="xs:anyType">
                <xs:sequence minOccurs="1" maxOccurs="unbounded">
                    <xs:element name="actionResult" minOccurs="0">
                        <xs:complexType>
                            <xs:attribute name="action"
                                          type="xs:token"
                                          use="required"/>
                            <xs:attribute name="success"
                                          type="xs:boolean"
                                          use="required"/>
                            <xs:attribute name="reason"
                                          type="xs:token">
                                <xs:annotation>
                                    <xs:documentation>conditionally
                                         mandatory when @success='false"
                                        to indicate reason code for a
                                        failure </xs:documentation>
                                </xs:annotation>
                            </xs:attribute>
                            <xs:attribute name="details"
                                          type="xs:string"/>
                            <xs:anyAttribute processContents="skip"/>
                        </xs:complexType>
                    </xs:element>
                    <xs:any namespace="##other" processContents="lax"
                            minOccurs="0"
                            maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attribute name="ref"
                              type="xs:anyURI"
                              use="required"/>
                <xs:attribute name="received"
```

```
                                type="xs:boolean"/>
                    <xs:anyAttribute/>
                </xs:restriction>
            </xs:complexContent>
        </xs:complexType>


        <xs:complexType name="capabilitiesType">
            <xs:complexContent>
                <xs:restriction base="xs:anyType">
                    <xs:sequence minOccurs="1" maxOccurs="unbounded">
                        <xs:element name="request"
                                    type="pi:requestType"
                                    minOccurs="1"
                            maxOccurs="unbounded"/>
                        <xs:any namespace="##other" processContents="lax"
                                 minOccurs="0"
                            maxOccurs="unbounded"/>
                    </xs:sequence>
                    <xs:anyAttribute/>
                </xs:restriction>
            </xs:complexContent>
        </xs:complexType>


        <xs:complexType name="requestType">
            <xs:complexContent>
                <xs:restriction base="xs:anyType">
                    <xs:choice minOccurs="1" maxOccurs="unbounded">
                        <xs:any namespace="##other" processContents="lax"
                                minOccurs="0"
                                maxOccurs="unbounded"/>
                    </xs:choice>
                    <xs:attribute name="action" type="xs:token" use="required"/>
                    <xs:attribute name="msgid" type="xs:unsignedInt"/>
                    <xs:attribute name="persistence" type="xs:duration"/>
                    <xs:attribute name="datatype" type="xs:token"/>
                    <xs:attribute name="supported-datatypes" type="xs:string"/>
                    <xs:attribute name="lamp-id" type="xs:token"/>
                    <xs:attribute name="lamp-action">
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:pattern value=""/>
                                <xs:pattern value=""/>
                                <xs:enumeration value="on"/>
                                <xs:enumeration value="off"/>
                                <xs:enumeration value="flash"/>
                            </xs:restriction>
```

```
                        </xs:simpleType>
                    </xs:attribute>
                    <xs:attribute name="supported-lamps" type="xs:string"/>
                    <xs:attribute name="camera-id" type="xs:token"/>
                    <xs:attribute name="supported-cameras" type="xs:string"/>
                    <xs:anyAttribute/>
                </xs:restriction>
            </xs:complexContent>
        </xs:complexType>

    </xs:schema>
```

Figure 10: eCall Control Block Schema

13. IANA Considerations

13.1. Service URN Registrations

IANA is requested to register the URN 'urn:service:sos.ecall' under the sub-services 'sos' registry defined in Section 4.2 of [RFC5031].

This service identifies a type of emergency call (placed by a specialized in-vehicle system and a carrying standardized set of data related to the vehicle and crash or incident, and is needed to direct the call to a specialized public safety answering point (PSAP) with technical and operational capabilities to handle such calls. Two sub-services are registered as well, namely

urn:service:sos.ecall.manual

This service URN indicates that an eCall had been triggered based on the manual interaction of the driver or a passenger.

urn:service:sos.ecall.automatic

This service URN indicates that an eCall had been triggered automatically, for example, due to a crash or other serious incident (e.g., fire).

IANA is also requested to register the URN 'urn:service:test.sos.ecall' under the sub-service 'test' registry defined in Setcion 17.2 of [RFC6881].

13.2. MIME Content-type Registration for 'application/ emergencyCallData.eCall.MSD+xml'

IANA is requested to add application/emergencyCallData.eCall.MSD+xml as a MIME content type, with a reference to this document, in accordance to the procedures of RFC 6838 [RFC6838] and guidelines in RFC 7303 [RFC7303].

MIME media type name: application

MIME subtype name: emergencyCallData.eCall.MSD+xml

Mandatory parameters: none

Optional parameters: charset

Indicates the character encoding of the XML content.

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

Security considerations: This content type is designed to carry vehicle and incident-related data during an emergency call. This data contains personal information including vehicle VIN, location, direction, etc. Appropriate precautions need to be taken to limit unauthorized access, inappropriate disclosure to third parties, and eavesdropping of this information. In general, it is permissible for the data to be unprotected while briefly in transit within the Mobile Network Operator (MNO); the MNO is trusted to not permit the data to be accessed by third parties. Sections 7 and Section 8 of [I-D.ietf-ecrit-additional-data] contain more discussion.

Interoperability considerations: None

Published specification: Annex C of EN 15722 [msd]

Applications which use this media type: Pan-European eCall compliant systems

Additional information: None

Magic Number: None

File Extension: .xml

Macintosh file type code: 'TEXT'

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

Intended usage: LIMITED USE

Author: This specification was produced by the European Committee For Standardization (CEN). For contact information, please see <http://www.cen.eu/cen/Pages/contactus.aspx>.

Change controller: The European Committee For Standardization (CEN)

13.3. MIME Content-type Registration for 'application/ emergencyCallData.eCall.control+xml'

IANA is requested to add application/ emergencyCallData.eCall.control+xml as a MIME content type, with a reference to this document, in accordance to the procedures of RFC 6838 [RFC6838] and guidelines in RFC 7303 [RFC7303].

MIME media type name: application

MIME subtype name: emergencyCallData.eCall.control+xml

Mandatory parameters: none

Optional parameters: charset

Indicates the character encoding of the XML content.

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

Security considerations: This content type carries metadata and control information and requests, primarily from a Public Safety Answering Point (PSAP) to an In-Vehicle System (IVS) during an emergency call, and also capabilities from the IVS to the PSAP. Metadata (such as an acknowledgment that data sent by the IVS to the PSAP was successfully received) has limited privacy and security implications. Control information (such as requests from the PSAP that the vehicle perform an action) has some privacy and important security implications. The privacy concern arises from the ability to request the vehicle to transmit a data set, which as described in Section 13.2, may contain personal information. The security implication is the ability to request the vehicle to perform an action. It is important that control information originate only from a PSAP or other emergency services provider,

and not from an impostor. The first safeguard for this is the security of the cellular network over which the emergency call was placed. In particular, when the IVS initiates an eCall over a cellular network, the MNO routes the call to a PSAP. (Calls placed using other means, such as Wi-Fi or over-the-top services, do not carry the same degree of trust.) Calls received by the IVS, such as a call-back from a PSAP, also do not carry the same degree of trust, since the current mechanisms are not ideal for verifying that such a call is indeed from a PSAP in response to an emergency call placed by the IVS. See the discussion in Section 11 and the PSAP Callback document [RFC7090]. A further safeguard, and one applicable regardless of which end initiated the call and the means of the call, is for the PSAP or emergency service provider to sign the body part using a certificate issued by a known emergency services certificate authority and for which the IVS can verify the root certificate. Sections 7 and Section 8 of [I-D.ietf-ecrit-additional-data] contain more discussion.

Interoperability considerations: None

Published specification: Annex C of EN 15722 [msd]

Applications which use this media type: Pan-European eCall compliant systems

Additional information: None

Magic Number: None

File Extension: .xml

Macintosh file type code: 'TEXT'

Person and email address for further information: Randall Gellens, rg+ietf@qti.qualcomm.com

Intended usage: LIMITED USE

Author: The IETF ECRIT WG.

Change controller: The IETF ECRIT WG.

13.4. Registration of the 'eCall.MSD' entry in the Emergency Call Additional Data Blocks registry

This specification requests IANA to add the 'eCall.MSD' entry to the Emergency Call Additional Data Blocks registry (established by [additional-data-draft]), with a reference to this document.

13.5. Registration of the ’eCall.control’ entry in the Emergency Call Additional Data Blocks registry

This specification requests IANA to add the ’eCall.control’ entry to the Emergency Call Additional Data Blocks registry (established by [additional-data-draft]), with a reference to this document.

13.6. Registration of the emergencyCallData.eCall Info Package

IANA is requested to add emergencyCallData.eCall to the Info Packages Registry under "Session Initiation Protocol (SIP) Parameters", with a reference to this document.

13.7. URN Sub-Namespace Registration

13.7.1. Registration for urn:ietf:params:xml:ns:eCall

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

URI: urn:ietf:params:xml:ns:eCall

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 eCall Data</title>
</head>
<body>
    <h1>Namespace for eCall Data</h1>
    <p>See [TBD: This document].</p>
</body>
</html>
END
```

13.7.2. Registration for urn:ietf:params:xml:ns:eCall:control

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

URI: urn:ietf:params:xml:ns:eCall:control

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 eCall Data:
            Control Block</title>
</head>
<body>
     <h1>Namespace for eCall Data</h1>
     <h2>Control Block</h2>
<p>See [TBD: This document].</p>
</body>
</html>
END
```

13.8. Registry creation

This document creates a new registry called ’eCall Control Data’. The following sub-registries are created for this registry.

13.8.1. eCall Control Action Registry

This document creates a new sub-registry called "eCall Control Action Registry". As defined in [RFC5226], this registry operates under "Expert Review" rules. The expert should determine that the proposed action is within the purview of a vehicle, is sufficiently distinguishable from other actions, and the actions is clearly and fully described. In most cases, a published and stable document is referenced for the description of the action.

The content of this registry includes:

Name: The identifier to be used in the 'action' attribute of an eCall control <request> element.

Description: A description of the action. In most cases this will be a reference to a published and stable document. The description MUST specify if any attributes or child elements are optional or mandatory, and describe the action to be taken by the vehicle.

The initial set of values is listed in Table 2.

Name	Description
send-data	Section xxx of this document
msg-static	Section xxx of this document
msg-dynamic	Section xxx of this document
honk	Section xxx of this document
lamp	Section xxx of this document
enable-camera	Section xxx of this document

Table 2: eCall Control Action Registry Initial Values

13.8.2. eCall Static Message Registry

This document creates a new sub-registry called "eCall Static Message Registry". Because all compliant vehicles are expected to support all static messages translated into all languages supported by the vehicle, it is important to limit the number of such messages. As defined in [RFC5226], this registry operates under "Publication Required" rules, which require a stable, public document and imply expert review of the publication. The expert should determine that the document has been published by an appropriate emergency services organization (e.g., NENA, EENA, APCO) and that the proposed message is sufficiently distinguishable from other messages.

The content of this registry includes:

ID: An integer identifier to be used in the 'msgid' attribute of an eCall control <request> element.

Message: The text of the message. Messages are listed in the registry in English; vehicles are expected to implement translations into languages supported by the vehicle.

When new messages are added to the registry, the message text is determined by the registrant; IANA assigns the IDs. Each message is assigned a consecutive integer value as its ID. This allows an IVS to indicate by a single integer value that it supports all messages with that value or lower.

The initial set of values is listed in Table 3.

ID	Message
1	Emergency authorities are aware of your incident and location, but are unable to speak with you right now. We will help you as soon as possible.

Table 3: eCall Static Message Registry

13.8.3. eCall Reason Registry

This document creates a new sub-registry called "eCall Reason Registry" which contains values for the 'reason' attribute of the <actionResult> element. As defined in [RFC5226], this registry operates under "Expert Review" rules. The expert should determine that the proposed reason is sufficiently distinguishable from other reasons and that the proposed description is understandable and correctly worded.

The content of this registry includes:

ID: A short string identifying the reason, for use in the 'reason' attribute of an <actionResult> element.

Description: A description of the reason.

The initial set of values is listed in Table 4.

ID	Description
unsupported	The 'action' is not supported.
unable	The 'action' could not be accomplished.
data-unsupported	The data item referenced in a 'send-data' request is not supported.

Table 4: eCall Reason Registry

13.8.4. eCall Lamp ID Registry

This document creates a new sub-registry called "eCall Lamp ID Registry" to standardize the names of automotive lamps (lights). As defined in [RFC5226], this registry operates under "Expert Review" rules. The expert should determine that the proposed lamp name is clearly understandable and is sufficiently distinguishable from other lamp names.

The content of this registry includes:

Name: The identifier to be used in the 'lamp-ID' attribute of an eCall control <request> element.

Description: A description of the lamp (light).

The initial set of values is listed in Table 5.

Name	Description
head	The main lamps used to light the road ahead
interior	Interior lamp, often at the top center
fog-front	Front fog lamps
fog-rear	Rear fog lamps
brake	Brake indicator lamps
position-front	Front position/parking/standing lamps
position-rear	Rear position/parking/standing lamps
turn-left	Left turn/directional lamps
turn-right	Right turn/directional lamps
hazard	Hazard/four-way lamps

Table 5: eCall Lamp ID Registry Initial Values

13.8.5. eCall Camera ID Registry

This document creates a new sub-registry called "eCall Camera ID Registry" to standardize the names of automotive camera. As defined in [RFC5226], this registry operates under "Expert Review" rules. The expert should determine that the proposed camera name is clearly understandable and is sufficiently distinguishable from other camera names.

The content of this registry includes:

Name: The identifier to be used in the ’camera-ID’ attribute of an eCall control <request> element.

Description: A description of the camera.

The initial set of values is listed in Table 6.

Name	Description
backup	Shows what is behind the vehicle. Also known as rearview, reverse, etc.
interior	Shows the interior (driver)

Table 6: eCall Camera ID Registry Initial Values

14. Contributors

Brian Rosen was a co-author of the original document upon which this document is based.

15. Acknowledgements

We would like to thank Bob Williams and Ban Al-Bakri for their feedback and suggestions, and Keith Drage for his review comments. We would like to thank Michael Montag, Arnoud van Wijk, Gunnar Hellstrom, and Ulrich Dietz for their help with the original document upon which this document is based.

16. Changes from Previous Versions

16.1. Changes from draft-ietf-02 to draft-ietf-03

- Added request to enable cameras
- Improved examples and XML schema
- Clarifications and wording improvements

16.2. Changes from draft-ietf-01 to draft-ietf-02

- Added clarifying text reinforcing that the data exchange is for small blocks of data infrequently transmitted
- Clarified that dynamic media is conveyed using SIP re-INVITE to establish a one-way media stream
- Clarified that the scope is the needs of eCall within the SIP emergency call environment
- Added informative statement that the document may be suitable for reuse by other ACN systems
- Clarified that normative language for the control block applies to both IVS and PSAP
- Removed ’ref’, ’supported-mime’, and <media> elements
- Minor wording improvements and clarifications

16.3. Changes from draft-ietf-00 to draft-ietf-01

- Added further discussion of test calls
- Added further clarification to the document scope
- Mentioned that multi-region vehicles may need to support other crash notification specifications in addition to eCall
- Added details of the eCall metadata and control functionality
- Added IANA registration for the MIME content type for the eCall control object
- Added IANA registries for protocol elements and tokens used in the eCall control object
- Minor wording improvements and clarifications

16.4. Changes from draft-gellens-03 to draft-ietf-00

- Renamed from draft-gellens- to draft-ietf-.
- Added mention of and reference to ETSI TR "Mobile Standards Group (MSG); eCall for VoIP"
- Added text to Introduction regarding migration/co-existence being out of scope
- Added mention in Security Considerations that even if the network-supplied location is just the cell site, this can be useful as a sanity check on the IVS-supplied location
- Minor wording improvements and clarifications

16.5. Changes from draft-gellens-02 to -03

- Clarifications and editorial improvements.

16.6. Changes from draft-gellens-01 to -02

- Minor wording improvements
- Removed ".automatic" and ".manual" from "urn:service:test.sos.ecall" registration and discussion text.

16.7. Changes from draft-gellens-00 to -01

- Now using ’EmergencyCallData’ for purpose parameter values and MIME subtypes, in accordance with changes to [additional-data-draft]
- Added reference to RFC 6443
- Fixed bug that caused Figure captions to not appear

17. References

17.1. Normative References

[EN_16072]
CEN, , "Intelligent transport systems - eSafety - Pan-European eCall operating requirements", December 2011.

[I-D.ietf-ecrit-additional-data]
Gellens, R., Rosen, B., Tschofenig, H., Marshall, R., and J. Winterbottom, "Additional Data Related to an Emergency Call", draft-ietf-ecrit-additional-data-30 (work in progress), June 2015.

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

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.

[RFC5031] Schulzrinne, H., "A Uniform Resource Name (URN) for Emergency and Other Well-Known Services", RFC 5031, January 2008.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC6443] Rosen, B., Schulzrinne, H., Polk, J., and A. Newton, "Framework for Emergency Calling Using Internet Multimedia", RFC 6443, December 2011.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, 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.

[RFC7303] Thompson, H. and C. Lilley, "XML Media Types", RFC 7303, July 2014.

[TS22.101]
3GPP, , "Technical Specification Group Services and System Aspects; Service aspects; Service principles", .

[additional-data-draft]
Rosen, B., Tschofenig, H., Marshall, R., Gellens, R., and J. Winterbottom, "Additional Data related to an Emergency Call", draft-ietf-ecrit-additional-data-11 (work in progress), July 2013.

[msd] CEN, , "Intelligent transport systems -- eSafety -- eCall minimum set of data (MSD), EN 15722", June 2011.

17.2. Informative references

[CEN] "European Committee for Standardization", <http://www.cen.eu>.

[I-D.ietf-ecrit-trustworthy-location]
Tschofenig, H., Schulzrinne, H., and B. Aboba, "Trustworthy Location", draft-ietf-ecrit-trustworthy-location-14 (work in progress), July 2014.

[MSG_TR] ETSI, , "ETSI Mobile Standards Group (MSG); eCall for VoIP", ETSI Technical Report TR 103 140 V1.1.1 (2014-04), April 2014.

[RFC5012] Schulzrinne, H. and R. Marshall, "Requirements for Emergency Context Resolution with Internet Technologies", RFC 5012, January 2008.

[RFC5069] Taylor, T., Tschofenig, H., Schulzrinne, H., and M. Shanmugam, "Security Threats and Requirements for Emergency Call Marking and Mapping", RFC 5069, January 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.

[RFC6086] Holmberg, C., Burger, E., and H. Kaplan, "Session Initiation Protocol (SIP) INFO Method and Package Framework", RFC 6086, January 2011.

[RFC6442] Polk, J., Rosen, B., and J. Peterson, "Location Conveyance for the Session Initiation Protocol", RFC 6442, December 2011.

[RFC7090] Schulzrinne, H., Tschofenig, H., Holmberg, C., and M. Patel, "Public Safety Answering Point (PSAP) Callback", RFC 7090, April 2014.

[SDO-3GPP]
"3d Generation Partnership Project",
<http://www.3gpp.org/>.

[SDO-ETSI]
"European Telecommunications Standards Institute (ETSI)",
<http://www.etsi.org>.

[draft-ietf-ecrit-car-crash]
Gellens, R., Rosen, B., and H. Tschofenig, "Next-Generation Vehicle-Initiated Emergency Calls", draft-ietf-ecrit-car-crash (work in progress), March 2015.

Authors' Addresses

Randall Gellens
Qualcomm Technologies, Inc.
5775 Morehouse Drive
San Diego 92651
US

Email: rg+ietf@qti.qualcomm.com

Hannes Tschofenig

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

ecrit
Internet-Draft
Intended status: Standards Track
Expires: April 21, 2016

B. Rosen
Neustar

October 19, 2015

Validation of Locations Around a Planned Change
draft-rosen-ecrit-lost-planned-changes-03

Abstract

This document defines an extension to LoST (RFC5222) that allows a planned change to the data in the LoST server to occur. Records that previously were valid will become invalid at a date in the future, and new locations will become valid after the date. The extension adds two elements to the <findservice> request: a URI to be used to inform the LIS that previously valid locations will be invalid after the planned change date, and add a date which requests the server to perform validation as of the date specified. It also adds an optional TTL element to the response, which informs all queriers the current expected lifetime of the validation.

Table of Contents

1. Introduction

This document describes an update to the LoST protocol [RFC5222] which allows a <findservice> request to optionally add a URI and a date to be used with planned changes to the underlying location information in the server. The URI is retained by the LoST server, associated with the data record that was validated, and used to notify the LIS (the LoST client) when a location which was previously valid will become invalid. The date is used by the client to ask the server to perform validation as of a future date. In addition to this mechanism, the lt;findserviceResponse> is also extended to provide a TTL for validation, after which the client should revalidate the location.

Validation of civic locations involves dealing with data that changes over time. A typical example is a portion of a county or province that was not part of a municipality is "annexed" to a municipality. Prior to the change, the content of the PIDF A3 element would be blank, or represent some other value and after the change would be the municipality that annexed that part of the county/province. This kind of annexation has an effectivity date and time (typically 00:00 on the first or last day of a month).

Records in a LIS must change around these kinds of events. The old record must be discarded, and a new, validated record must be loaded into the LIS. It is often difficult for the LIS operator to know

that records must be changed around such events. There are other circumstances where locations that were previously valid become invalid, such as a street renaming or renumbering event. As RFC5222 defines validation, the only way for a LIS to discover such changes was to periodically revalidate its entire database. Of course, this would not facilitate timely changes, is not coordinated with the actual change event, and also adds significant load to the LoST server. Even if re-validation is contemplated, the server has no mechanism to control, or even suggest the time period for revalidation

This extension allows the client to provide a stable URI that is retained by the server associated with the location information used in the request. In the event of a planned change, or any other circumstance where the LI becomes invalid, the server sends a notification to the URI informing it of a change. The notification contains the date and time when the LI becomes invalid.

Ideally, following such a notification, the LIS will prepare a new record to be inserted in its active database, that becomes active at the precise planned event date and time, at which point it would also delete the old record. However, the new record has to be valid, and the LIS would like to validate it prior to the planned change event. If it requests validation before the planned event, the server (without this extension) would inform the client that the location was invalid. This extension includes an optional "asOf" date and time in the request that allows the LoST server to provide validation as of the date and time specified, as opposed to the "as of now" implied in the current LoST protocol.

When it is not practical or advisable for the LIS to maintain stable URIs for all of its records, periodic revalidation can be still used to maintain the data in the LIS. However, the server should be able to control the rate of such revalidation. For this purpose, a new TTL element is included in the lt;findserviceResponse> which provides advice from the server to the LIS of when validation is suggested.

2. Conventions used in this document

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

"Server" in this document refers to the LoST server and "Client" is the LoST client, even when the server is performing an operation on the client.

3. <plannedChange> element

This document defines a new element to <findService> called 'plannedChange'. This element contains two attributes: 'uri' and 'asOf'. The 'uri' attribute MUST be a URI with a scheme of https. The URI will be stored by the server against the location in the request for subsequent use with the notification function defined below. To minimize storage requirements of at the server, the length of the URI MUST be less than 256 bytes. Each client of the server may only store one URI against a location, where "location" is defined by policy at the server, since a given unique location may have many combinations of LI elements that resolve to the same location. If the server receives a 'uri' for the same location from the same client, the URI in the request replaces the URI it previously retained. Policy at the server may limit how many uris it retains for a given location. A new warning is defined below to be used to indicate that the URI has not been stored. If the location in the request is invalid, the uri will not be stored and the warning will be returned.

The 'asOf' attribute contains a date and time. The server will validate the location in the request as of the date specified, taking into account planned changes. This allows the client to verify that it can make changes in the LIS commensurate with changes in the LoST server by validating locations in advance of a change.

4. <locationInvalidated> object

When the server needs to invalidate a location where the client provided a URI in <plannedChange>, the server executes an HTTPS POST containing <locationInvalidated> to the URI previously provided. This is the notice from the server to the client that the location may be invalid and should be revalidated. <locationInvalidated> contains an asOf attribute that specifies when the location may become invalid. If the date/time in asOf is earlier than the time the <locationInvalidated> was sent, the location may already be invalid and the LIS should take immediate action. If the POST operation fails, the server MAY retry the operation immediately, and if it fails again, retry the operation at a later time.

5. uri Not Stored Warning

A new warning is added to the exceptionContainer, 'uriNotStored'. This warning MUST NOT be returned unless the plannedChange element was found in the corresponding request. The warning is returned when the server decides not to store the URI found in the plannedChange element. As discussed above, this may occur because, among other reasons, the policy at the server limits how many URIs will be stored

against a specific location, the uri is not well formed or the policy at the server has some other restriction on the feature.

6. TTL in Response

A new 'ttl' element is added to the lt;findserviceResponse>. The ttl element contains a date and time after which the client may wish to revalidate the location at the server. This element MAY be added by the server if validation is requested in the response. The form of the element is the 'expires' pattern, which allows explicit 'No Cache' and 'No Expiration' values to be returned. 'No Cache' has no meaning and MUST NOT be returned in TTL. 'No Expiration' means the server does not have any suggested revalidation period.

Selecting a revalidation interval is a complex balancing of timeliness, server load, stability of the underlying data, and policy of the LoST server. Too short, and load on the server may overwhelm it. Too long and invalid data may persist in the server for too long. The URI mechanism provides timely notice to coordinate changes, but even with it, it is often advisable to revalidate data eventually.

In areas that have little change in data, such as fully built out, stable communities already part of a municipality, it may be reasonable to set revalidation periods of 6 months or longer, especially if the URI mechanism is widely deployed at both the server and the clients. In areas that are quickly growing, 20-30 day revalidation may be more appropriate even though such revalidation would be the majority of the traffic on the LoST server.

When a planned change is made, typically the TTL for the affected records is lowered, so that revalidation is forced soon after the change is implemented. It is not advisable to set the expiration precisely at the planned change time if a large number of records will be changed, since that would cause a large spike in traffic at the change time. Rather, the expiration time should have a random additional time added to it to spread out the load.

7. Relax NG Schema

The Relax NG schema in [RFC5222] is extended to include:

```
namespace a = "http://relaxng.org/ns/compatibility/annotations/1.0"
default namespace ns1 = "urn:ietf:params:xml:ns:lost-plannedChange1"

##
##       Extension to Location-to-Service Translation (LoST) Protocol
##       to support a planned change to location data
```

```
   ##
   ##       plannedChange is used in the extensionPoint of
   ##       commonRequestPattern in a findService request
   ##
   ##       locationInvalidated is used by the LoST server to notify a
   ##       LIS that a previously valid location may be (or will become)
   ##       invalid
   ##
   ##       ttl is used in the extensionPoint of
   ##       commonResponsePattern in a findService response
   ##
   ##       uriNotStored is a new warning to be used in a
   ##       exceptionContainer in the warnings element of a
   ##       findServiceResponse
   ##
   start =
     plannedChange
     | locationInvalidated
     | uriNotStored
   ##
   ##       plannedChange
   ##
   div {
     plannedChange =
       element plannedChange {
         attribute uri {
           xsd:anyURI }?,
         attribute asOf {
           xsd:dateTime }?,
         extensionPoint+
       }
   }

   ##
   ##       locationInvalidated
   ##
   div {
     locationInvalidated =
       element locationInvalidated {
         attribute asOf {
           xsd:dateTime }?,
         extensionPoint+
       }
   }

   ##
   ##       ttl
   ##
```

```
   div {
     ttl =
       element ttl {
         expires,
         extensionPoint+
       }
   }

   ##
   ##       uriNotStored
   ##
   div {
     uriNotStored =
       element uriNotStored { basicException }
   }

   ##
   ##       Patterns for inclusion of elements from schemas in
   ##       other namespaces.
   ##
   div {

     ##
     ##         Any element not in the LoST namespace.
     ##
     notLostChange = 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 = notLostChanged*
   }
```

8. Security Considerations

As an extension to LoST, this document inherits the security issues raised in [RFC5222]. The server could be tricked into storing a malicious URI which, when sent the locationInvalidated object could trigger something untoward. The server MUST NOT accept any data from the client in response to POSTing the locationInvalidated.

The server is subject to abuse by clients because it is being asked to store something and may need to send data to an uncontrolled URI. Clients could request many URIs for the same location for example. The server MUST have policy that limits use of this mechanism by a given client. If the policy is exceeded, the server returns the uriNotStored warning. The server MUST validate that the content of the uri sent is syntactically valid and meets the 256 byte limit. When sending the locationInvalidated object to the uri stored, the server MUST protect itself against common http vulnerabilities.

The mutual authentication between client and server when is RECOMMENDED for both the initial findService operation that requests storing the uri and the sending of the locationInvalidated object. The server should be well known to the client, and its credential should be learned in a reliable way. For example, a public safety system operating the LoST server may have a credential traceable to a well known Certificate Authority known to provide credentials for public safety agencies. Many of the clients will be operated by local ISPs or other service providers where the server operator can reasonably obtain a good credential to use for the URI. Where the server does not recognize the client, its policy MAY limit the use of this feature beyond what it would limit a client it recognized.

9. IANA Considerations

9.1. Relax NG Schema Registration

URI: urn:ietf:params:xml:schema:lost-planedChange1

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

Relax NG Schema: The Relax NG schema to be registered is contained in Section 5. Its first line is

default namespace = "urn:ietf:params:xml:ns:lost-PlannedChange1

and its last line is

}

9.2. LoST Namespace Registration

URI: urn:ietf:params:xml:ns:lost-plannedChange1

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

XML:

```
BEGIN
<?xml version="2.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
  "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="content-type"
        content="text/html;charset=iso-8859-1"/>
  <title>LoST Planned Change Namespace</title>
</head>
<body>
  <h1>Namespace for LoST Planned Change extension</h1>
  <h2>urn:ietf:params:xml:ns:lost-plannedChange1</h2>
<p>See <a href="http://www.rfc-editor.org/rfc/rfc????.txt">
   RFC????</a>.</p>
</body>
</html>
END
```

10. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC5222] Hardie, T., Newton, A., Schulzrinne, H., and H. Tschofenig, "LoST: A Location-to-Service Translation Protocol", RFC 5222, DOI 10.17487/RFC5222, August 2008, <http://www.rfc-editor.org/info/rfc5222>.

Author’s Address

Brian Rosen
Neustar
470 Conrad Dr
Mars, PA 16046
US

EMail: br@brianrosen.net