ECRIT
Internet-Draft
Intended status: Informational
Expires: August 17, 2014

R. Gellens
Qualcomm Technologies, Inc
B. Rosen
NeuStar, Inc.
H. Tschofenig
(no affiliation)

February 13, 2014

Internet Protocol-based In-Vehicle Emergency Calls
draft-gellens-ecrit-car-crash-02.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).

Profiling and simplifications are possible due to the nature of the functionality that is provided in vehicles with the usage of Global Satellite Navigation System (GNSS).

Status of This Memo

This Internet-Draft will expire on August 17, 2014.

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:

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." 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) and are difficult to extend as new sensors are added.

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 routed to a specialized 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 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.

This document describes how the IETF mechanisms for IP-based emergency calls, including [RFC6443] and [additional-data-draft], 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 [additional-data-draft]).

VEDS is a versatile structure that can accomodate varied needs. However, if additional sets of data are determined to be needed, 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 [additional-data-draft]); 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 (or appending to) a Call-Info header field at the top level of the INVITE. The 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 can be used place emergency calls that are identifiable as ACN calls and that carry one or more standardized crash data objects in an interoperable way.

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 not further discussed in this document but is the subject of a separate document, [eCall-draft]

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).

```
   ///----\\\  proprietary  +------+    911 trunk      +------+
  ||| IVS |||-------------->+ TSP  +------------------>+ PSAP |
   \\\----///  crash data   +------+                   +------+
```

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.

```
                +---+
   ///----\\\   | H |   911/etc voice call via handset   +------+
  ||| IVS |||-->| S +---------------------------------->+ PSAP |
   \\\----///   +---+   location 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.

```
      ///----\\\      911/etc voice call via IVS          +------+
     ||| IVS  |||---------------------------------------->+ PSAP |
      \\\----///     location via text-to-speech          +------+
```

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).

This document does not address pan-European eCall (a mandated and standardized system for emergency calls by in-vehicle systems within Europe and other regions), which is the subject of a separate document, [eCall-draft]

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.

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 3GPP IMS solution with a Request-URI indicating an ACN type of emergency call with vehicle data attached; the MNO only needs to recognize the call as an emergency call and route it to an ESInet; the ESInet recognizes the call as an ACN with vehicle data and routes the call to an NG-ACN capable PSAP; the PSAP interpets the vehicle data sent with the call and makes it available to the call taker.

Because of the need to identify and specially process Next-Generation ACN calls (as discussed above), this document registers new service URN children within the "sos" subservice. These URNs provide the mechanism by which an NG-ACN call is identified, and differentiate between manually and automatically triggered NG-ACN calls (which may be subject to different treatment, depending on policy). The two service URNs are: 'urn:service:sos.vehicle.automatic' and 'urn:service:sos.vehicle.manual'.

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).

```
                proprietary
    ///----\\\  or standard        +------+     standard       +------+
   ||| IVS ||| ------------------->+ TSP  +------------------->+ PSAP |
    \\\----///  crash + other data +------+ crash + other data +------+
```

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.

```
                                 +---+
    ///----\\\     (unclear)     | H |      (unclear)     +------+
   ||| IVS |||------------------>| S +------------------->+ PSAP |
    \\\----///     (unclear)     +---+      (unclear)     +------+
```

Figure 5: Next-Generation Paired Model

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

```
    ///----\\\          NG1-1-2/NG9-1-1 call           +------+
   ||| IVS |||---------------------------------------->+ PSAP |
    \\\----///               crash data                +------+
```

Figure 6: Next-Generation Model

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 . 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). The IVS MAY attach location data to the call INVITE. 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 also 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 versis 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 need to be 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, 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 need to 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), this document proposes an SOS sub-service for ACN/car crash, specifically, "SOS.vehicle". Using a sub-service makes it readily obvious that the call is an ACN; a further child elements is proposed to distinguish 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.vehicle.automatic" and "SOS.vehicle.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 [additional-data-draft]. 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 [additional-data-draft]

   - 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 [additional-data-draft]

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 [additional-data-draft], 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 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 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 specification also 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. For example, "urn:service:test.sos.vehicle.automatic" indicates such a test feature. This functionality is defined in [RFC6881].

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.vehicle.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.

```
                             +-----------------------------------------+
                             |                                         |
             +------------+  |                  +-------+              |
             |            |  |                  | PSAP2 |              |
             |            |  |                  +-------+              |
             | Originating|  |                                         |
             |   Mobile   |  |  +------+     +-------+                 |
   Vehicle-->|  Network   |--+->| ESRP |---->| PSAP1 |---> Call-Taker  |
             |            |  |  +------+     +-------+                 |
             |            |  |                                         |
             +------------+  |                  +-------+              |
                             |                  | PSAP3 |              |
                             |                  +-------+              |
                             |                                         |
                             |                                         |
                             |                                         |
                             |                ESInet                   |
                             +-----------------------------------------+
```

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

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

```
   INVITE urn:service:sos.vehicle.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

   ...eCall VEDS data object goes here

       --boundary1--
```

Figure 8: SIP INVITE indicating an In-Vehicular 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 devices. The reader is referred to [I-D.ietf-ecrit-trustworthy-location] for a discussion of some of these vulnerabilities.

12. IANA Considerations

12.1. Service URN Registration

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

This service identifier reaches a public safety answering point (PSAP), which in turn dispatches aid appropriate to the emergency related to accidents of vehicles. The following two sub-services are registered as well:

urn:service:sos.vehicle.manual

This service URN indicates that an emergency call carrying vehicle sensor ("crash") data has been placed by an in-vehicle system (IVS) based on the manual interaction of the driver or a passenger.

urn:service:sos.vehicle.automatic

This service URN indicates that an emergency call carrying vehicle sensor ("crash") data has been placed by an in-vehicle system (IVS) triggered automatically, for example, due to a crash.

12.2. MIME Content-type Registration for ’application/ EmergencyCall.VEDS+xml’

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

MIME media type name: application

MIME subtype name: 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 [additional-data-draft] 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>

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

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

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 -01 to -02

- Fixed case of 'EmergencyCallData', in accordance with changes to [additional-data-draft]

15.2. Changes from -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

16. References

16.1. Normative References

[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.

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

[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>.

[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.

16.2. Informative references

[I-D.ietf-ecrit-trustworthy-location]
Tschofenig, H., Schulzrinne, H., and B. Aboba, "Trustworthy Location", draft-ietf-ecrit-trustworthy-location-07 (work in progress), July 2013.

[RFC4481] Schulzrinne, H., "Timed Presence Extensions to the Presence Information Data Format (PIDF) to Indicate Status Information for Past and Future Time Intervals", RFC 4481, July 2006.

[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.

[eCall-draft]
Gellens, RG., "Next-Generation Pan-European eCall", 2013.

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
(no affiliation)

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

ECRIT
Internet-Draft
Intended status: Informational
Expires: August 17, 2014

R. Gellens
Qualcomm Technologies, Inc.
H. Tschofenig
(no affiliation)

February 13, 2014

Next-Generation Pan-European eCall
draft-gellens-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 by 2015 in European Union member states, and eCall is 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.

Status of This Memo

This Internet-Draft will expire on August 17, 2014.

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:

3GPP: 3rd Generation Partnership Project
CEN: European Committee for Standardization
EENA: European Emergency Number Association
ESInet: Emergency Services IP network
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). eCall itself is specified by 3GPP and CEN and these specifications include far greater scope than is covered here.

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 2015. eCall is 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 (e.g., a fire) 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). An eCall flag in the call setup marks the call as an eCall, and further indicates 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 optionally 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. 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.

4. eCall Requirements

Overall eCall requirements are specified by by 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 (due to its more compact size). 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.

6. Call Setup

In circuit-switched eCall, the IVS places a special form of a 112 emergency call which carries the 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).

```
    ///----\\\      112 voice call with eCall flag      +------+
   ||| IVS  |||---------------------------------------->+ PSAP |
    \\\----///   vehicle data via eCall in-band modem   +------+
```

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.

```
///----\\\     IMS emergency call with eCall URN    +------+
   IVS    ----------------------------------------->+ PSAP |
\\\----///    vehicle data included in call setup   +------+
```

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.

8. Test Calls

eCall requires the ability to place test calls. These are calls that are recognized and treated as eCalls but are not given emergency call treatment and are not handled by call takers.

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.

9. eCall-Specific Data from PSAP to IVS

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

The same mechanism established in [additional-data-draft], used in this document to carry the MSD from the IVS to the PSAP, can be additionally used to carry a control data block from the PSAP to the IVS. This eCall control block (also 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 the message that contained the MSD or other data (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 creates a framework mechanism by which the PSAP can send eCall-specific data to the IVS and the IVS can respond with data if requested. If control data sent in a response message requests the IVS to send a new MSD or other data block, the IVS can do so 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 (which may be as simple as 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

10. Example

Figure 3 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. (In deployments where there is no ESInet, the originating network routes the call directly 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 to the ESInet. The ESRP routes the call to a PSAP, where it is received by a call taker.

```
            +------------+  +-----------------------------------------+
            |            |  |                                         |
            |            |  |                  +--------+             |
            |            |  |                  | PSAP2  |             |
            |            |  |                  +--------+             |
            |            |  |                                         |
            |            |  |  +------+     +-------+               |
  Vehicle-->|            |--+->| ESRP |---->| PSAP1 |---> Call-Taker  |
            |            |  |  +------+     +-------+               |
            |            |  |                                       |
            |            |  |                  +-------+            |
            |            |  |                  | PSAP3 |            |
            |            |  |                  +-------+            |
            |            |  |                                       |
            | Originating|  |                                       |
            |   Mobile   |  |                                       |
            |  Network   |  |                ESInet                 |
            +------------+  +---------------------------------------+
```

Figure 3: Example of NG-eCall Message Flow

The example, shown in Figure 4, illustrates a SIP eCall INVITE that contains an MSD.

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

      ...eCall MSD data object goes here

       --boundary1--
```

Figure 4: SIP NG-eCall 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 may be determined 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. 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 acknowledgment and optional requests to the vehicle requires authenticity considerations; when the PSAP request is received within an established session initiated by the vehicle as an eCall emergency call, 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 apply (see the PSAP Callback document [I-D.ietf-ecrit-psap-callback]).

12. IANA Considerations

12.1. Service URN Registration

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

This service identifies a type of emergency call (placed by a specialized in-vehicle system and 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].

12.2. MIME Content-type Registration for 'application/emergencyCallData.eCall.MSD+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.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 3023 [RFC3023].

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)

12.3. 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. Contributors

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

14. Acknowledgements

We would like to thank Bob Williams and Ban Al-Bakri for their feedback and suggestions. 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.

15. Changes from Previous Versions

15.1. Changes from -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

15.2. Changes from -02 to -03

- Clarifications and editorial improvements.

15.3. Changes from -01 to -02

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

16. References

16.1. Normative References

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

[I-D.ietf-ecrit-additional-data] Rosen, B., Tschofenig, H., Marshall, R., Randy, R., and J. Winterbottom, "Additional Data related to an Emergency Call", draft-ietf-ecrit-additional-data-15 (work in progress), November 2013.

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

[RFC3023] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", RFC 3023, January 2001.

[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.

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

[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.

[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.

16.2. Informative references

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

[I-D.ietf-ecrit-psap-callback] Schulzrinne, H., Tschofenig, H., Holmberg, C., and M. Patel, "Public Safety Answering Point (PSAP) Callback", draft-ietf-ecrit-psap-callback-13 (work in progress), October 2013.

[I-D.ietf-ecrit-trustworthy-location] Tschofenig, H., Schulzrinne, H., and B. Aboba, "Trustworthy Location", draft-ietf-ecrit-trustworthy-location-07 (work in progress), July 2013.

[RFC4481] Schulzrinne, H., "Timed Presence Extensions to the Presence Information Data Format (PIDF) to Indicate Status Information for Past and Future Time Intervals", RFC 4481, July 2006.

[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.

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

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

Authors' Addresses

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

Email: rg+ietf@qti.qualcomm.com

Hannes Tschofenig
(no affiliation)

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

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

B. Rosen
NeuStar
H. Tschofenig
(no affiliation)
R. Marshall
TeleCommunication Systems, Inc.
R. Gellens
Qualcomm Technologies, Inc.
J. Winterbottom
(no affiliation)

February 12, 2014

Additional Data related to an Emergency Call
draft-ietf-ecrit-additional-data-20.txt

Abstract

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

Status of This Memo

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

Copyright Notice

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 emergency service number, location information, and meta-data about the sources of the data. The device, the access network provider, and any service provider in the call path may have even more information useful for a PSAP. This document extends the basic set of data communicated with an IP-based emergency call, as described in [RFC6443] and [RFC6881], in order to carry additional data which may be useful to an entity or call taker handling the call. This data is "additional" to the basic information found in the emergency call signaling used.

In general, there are three categories of this additional data that may be transmitted with an emergency call:

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

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

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

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

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

This document defines an extensible set of data structures, and mechanisms to transmit this data either by value or by reference, either in the Session Initiation Protocol (SIP) call signaling or in the Presence Information Data Format Location Object (PIDF-LO). The data structures are usable by other communication systems and transports as well. The data structures are defined in Section 3, and the transport mechanisms (using SIP and HTTPS) are defined in Section 4.

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

2. Terminology

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

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

Within each data block definition (see Section 3), the values for the "Use:" label are specified as one of the following:

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

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

’Optional’: means they MAY be present.

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

3. Data Structures

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

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

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

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

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

'Comment': This block provides a way to supply free form human readable text to the PSAP or emergency responders. This simple structure is defined in Section 3.5.

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 "Content-ID" message body header field specified in RFC 2045 [RFC2045] except that the <DataProviderReference> element is not enclosed in brackets (the "<" and ">" symbols are omitted). In other words, the value of an <DataProviderReference> element is syntactically an addr-spec as specified in RFC 822 [RFC0822].

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

3.1. Data Provider Information

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

3.1.1. Data Provider String

Data Element: Data Provider String

Use: Required

XML Element: <DataProviderString>

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

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

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

3.1.2. Data Provider ID

Data Element: Data Provider ID

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

XML Element: <ProviderID>

Description: A jurisdiction-specific code for the access network provider or service provider shown in the <DataProvidedBy> element that created the structure. NOTE: In the US, 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>

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.

3.1.3. Data Provider ID Series

Data Element: Data Provider ID Series

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

XML Element: <ProviderIDSeries>

Description: Identifies the issuer of the ProviderId. The Provider ID Series Registry (see Section 9.1) initially contains the following valid entries:

- NENA

- EENA

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

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

3.1.4. Type of Data Provider

Data Element: Type of Data Provider ID

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

XML Element: <TypeOfProviderID>

Description: Identifies the type of data provider ID being supplied in the ProviderID data element. A registry with an initial set of values is shown in Figure 1 (see also Section 9.1).

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

Figure 1: Type of Data Provider ID Registry.

Reason for Need: Identifies 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.

3.1.5. Data Provider Contact URI

Data Element: Data Provider Contact URI

Use: Required

XML Element: <ContactURI>

Description: When provided by a service provider or an access network provider, this information MUST be a URI to a 24/7 support organization tasked to provide PSAP support for this emergency call. If the call is from a device, this SHOULD be the contact information of the owner of the device. If a telephone number is the contact address then it MUST be a tel URI. If it is provided as a SIP URI then it MUST be in the form of sip:telephonenumber@serviceprovider:user=phone. 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 [I-D.ietf-ecrit-psap-callback]).

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.

3.1.6. Data Provider Languages(s) Supported

Data Element: Data Provider Language(s) supported

Use: Required.

XML Element: <Language>

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

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

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

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

3.1.7. xCard of Data Provider

Data Element: xCard of Data Provider

Use: Optional

XML Element: <DataProviderContact>

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

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

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

3.1.8. Subcontractor Principal

Data Element: Subcontractor Principal

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

XML Element: <SubcontratorPrincipal>

Description: 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.

3.1.9. Subcontractor Priority

Data Element: Subcontractor Priority

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

XML Element: <SubcontractorPriority>

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

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

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

3.1.10. ProviderInfo Example

```
   <?xml version="1.0" encoding="UTF-8"?>
   <ad:EmergencyCallData.ProviderInfo
       xmlns:ad="urn:ietf:params:xml:ns:EmergencyCallData:ProviderInfo"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
           <ad:id>12345</ad:id>
       <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>Service Provider</ad:TypeOfProvider>
       <ad:ContactURI>sip:voip-provider@example.com</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 2: EmergencyCallData.ProviderInfo Example.

3.2. Service Information

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

3.2.1. Service Environment

Data Element: Service Environment

Use: Required

XML Element: <SvcEnvironment>

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

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

How Used by Call Taker: Information may be used to assist in determining equipment and manpower requirements for emergency responders. As the information is not always available, and the registry is not all encompassing, this is at best advisory information, but since it mimics a similar capability in some current emergency calling systems, it is known to be valuable. The service provider uses its best information (such as a rate plan, facilities used to deliver service or service description) to determine the information and is not responsible for determining the actual characteristics of the location where the call originates from.

3.2.2. Service Delivered by Provider to End User

Data Element: Service Delivered by Provider to End User

Use: Required

XML Element: <SvcDelByProvider>

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

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

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

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

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

Reason for Need: Knowing the type of service may assist the PSAP with the handling of the call.

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

3.2.3. Service Mobility Environment

Data Element: Service Mobility Environment

Use: Required

XML Element: <SvcMobility>

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

* Mobile: the device should be able to move at any time

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

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

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

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

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

3.2.4. EmergencyCallData.ServiceInfo Example

```
<?xml version="1.0" encoding="UTF-8"?>
<svc:EmergencyCallData.ServiceInfo
     xmlns:svc="urn:ietf:params:xml:ns:EmergencyCallData.ServiceInfo"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <svc:DataProviderReference>string0987654321@example.org
    </svc:DataProviderReference>
    <svc:id>12345</svc:id>
    <svc:SvcEnvironment>Business</svc:SvcEnvironment>
    <svc:SvcDelByProvider>MLTS</svc:SvcDelByProvider>
    <svc:SvcMobility>Fixed</svc:SvcMobility>
</svc:EmergencyCallData.ServiceInfo>
```

Figure 4: EmergencyCallData.ServiceInfo Example.

3.3. Device Information

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

3.3.1. Device Classification

Data Element: Device Classification

Use: Optional

XML Element: <DeviceClassification>

Description: This data element defines the kind of device making the emergency call. If the device provides the data structure, the device information SHOULD be provided. If the service provider provides the structure and it knows what the device is, the service provider SHOULD provide the device information. Often the carrier does not know what the device is. It is possible to receive two Additional Data Associated with a Call data structures, one created by the device and one created by the service provider. This information describes the device, not how it is being used. This data element defines the kind of device making the emergency call. The registry with the initial set of values is shown in Figure 5.

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

Figure 5: Device Classification Registry.

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

How Used by Call Taker: May assist with location of caller. For example, a cordless handset may be outside or next door. May provide the calltaker some context about the caller, the capabilities of the device used for the call or the environment the device is being used in.

3.3.2. Device Manufacturer

Data Element: Device Manufacturer

Use: Optional

XML Element: <DeviceMfgr>

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

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

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

3.3.3. Device Model Number

Data Element: Device Model Number

Use: Optional

XML Element: <DeviceModelNr>

Description: Model number of the device.

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

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

3.3.4. Unique Device Identifier

Data Element: Unique Device Identifier

Use: Optional

XML Element: <UniqueDeviceID>

XML Attribute: <TypeOfDeviceID>

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

The <TypeOfDeviceID> attribute identifies the type of device identifier. A registry with an initial set of values can be seen in Figure 6.

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

Figure 6: Registry with Device Identifier Types.

Reason for Need: Uniquely identifies the device (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>

3.3.5. 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.

3.3.6. Device/Service Specific Additional Data Structure Type

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

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

XML Element: <DeviceSpecificType>

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

* IEEE 1512

* VEDS

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

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

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

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

3.3.7. Issues with getting new types of data into use

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

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

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

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

For devices that have device or service specific data, there are two choices to carry it. A new block can be defined, or the device/service specific additional data URL the 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 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).

3.3.9. EmergencyCallData.DeviceInfo Example

```
<?xml version="1.0" encoding="UTF-8"?>
<dev:EmergencyCallData.DeviceInfo
    xmlns:dev="urn:ietf:params:xml:ns:EmergencyCallData:DeviceInfo"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <dev:DataProviderReference>string0987654321@example.org
    </dev:DataProviderReference>
    <dev:id>12345</dev:id>
    <dev:DeviceClassification>Fixed phone</dev:DeviceClassification>
    <dev:DeviceMfgr>Nokia</dev:DeviceMfgr>
    <dev:DeviceModelNr>Lumia 800</dev:DeviceModelNr>
    <dev:UniqueDeviceID TypeOfDeviceID="IMEI">35788104
    </dev:UniqueDeviceID>
</dev:EmergencyCallData.DeviceInfo>
```

Figure 7: EmergencyCallData.DeviceInfo Example.

3.4. Owner/Subscriber Information

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

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

3.4.1. Subscriber Data Privacy Indicator

Attribute: privacyRequested, boolean.

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

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

Reason for Need: Some jurisdictions require subscriber privacy to be observed.

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

3.4.2. xCard for Subscriber’s Data

Data Element: xCARD for Subscriber’s Data

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

XML Element: <SubscriberData>

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

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

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

3.4.3. EmergencyCallData.SubscriberInfo Example

```
   <?xml version="1.0" encoding="UTF-8"?>
   <sub:EmergencyCallData.SubscriberInfo
        xmlns:sub="urn:ietf:params:xml:ns:EmergencyCallData:SubscriberInfo"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        privacyRequested="false">
       <sub:DataProviderReference>string0987654321@example.org
       </sub:DataProviderReference>
       <sub:SubscriberData xmlns="urn:ietf:params:xml:ns:vcard-4.0">
           <vcards>
               <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>
           </vcards>
       </sub:SubscriberData>
   </sub:EmergencyCallData.SubscriberInfo>
```

Figure 8: EmergencyCallData.SubscriberInfo Example.

3.5. Comment

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

3.5.1. 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

3.5.2. EmergencyCallData.Comment Example

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

Figure 9: EmergencyCallData.Comment Example.

4. Data Transport Mechanisms

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

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

2. Second, the ability to embed additional data structures in the <provided-by> element of a PIDF-LO [RFC4119] is defined. Besides a service provider in the call path, the access network provider

may also have similar information that may be valuable to the PSAP. The access network provider may provide location in the form of a PIDF-LO from a location server via a location configuration protocol. The data structures described in this document are not specific to the location itself, but rather provides descriptive information having to do with the immediate circumstances about the provision of the location (who the access network is, how to contact that entity, what kind of service the access network provides, subscriber information, etc.). This data is similar in nearly every respect to the data known by service providers in the path of the call. When the access network provider and service provider are separate entities, the access network does not participate in the application layer signaling (and hence cannot add a Call-Info header field to the SIP message), but may provide location information to assist in locating the caller's device. The <provided-by> element of the PIDF-LO is a mechanism for the access network provider to supply the information about the entity or organization that supplied this location information. For this reason, this document describes a namespace per RFC 4119 for inclusion in the <provided-by> element of a PIDF-LO for adding information known to the access network provider.

One or more blocks of data registered in the Emergency Call Additional Data registry, as defined in Section 9.1, may be included or referenced in the SIP signaling (using the Call-Info header field) or in the <provided-by> element of a PIDF-LO. Every block must be one of the types in the registry. Since the data of an emergency call may come from multiple sources, the data itself needs information describing the source. Consequently, each entity adding additional data MUST supply the "Data Provider" block. All other blocks are optional, but each entity SHOULD supply any blocks where it has at least some of the information in the block.

4.1. Transmitting Blocks using the Call-Info Header

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

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

A Call-info header with a purpose value starting with 'EmergencyCallData' MUST only be sent on an emergency call, which can be ascertained by the presence of an emergency service urn in a Route header of a SIP message.

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

The data may also be supplied by value in a SIP message. In this case, Content Indirection (CID) [RFC2392] is used, with the CID URL referencing the MIME body part.

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

4.2. Transmitting Blocks by Reference using the Provided-By Element

The 'EmergencyCallDataReference' element is used to transmit an additional data block by reference within a 'Provided-By' element of a PIDF-LO. The 'EmergencyCallDataReference' element has two attributes: 'ref' to specify the URL, and 'purpose' to indicate the type of data block referenced. The value of 'ref' is an HTTPS URL that resolves to a data structure with information about the call. The value of 'purpose' is the same as used in a 'Call-Info' header field (as specified in Section 4.1).

For example, to reference a block with MIME type 'application/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"/>
```

4.3. Transmitting Blocks by Value using the Provided-By Element

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

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

For example:

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

Example Provided-By by Value.

4.4. The Content-Disposition Parameter

RFC 5621 [RFC5621] discusses the handling of message bodies in SIP. It updates and clarifies handling originally defined in RFC 3261 [RFC3261] based on implementation experience. While RFC 3261 did not mandate support for 'multipart' message bodies, 'multipart/mixed'

MIME bodies are used by many extensions (including this document) today. For example, adding a PIDF-LO, SDP, and additional data in body of a SIP message requires a 'multipart' message body.

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

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

```
Content-Type: application/sdp

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

--boundary1

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

...PIDF-LO goes in here

--boundary1--

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

...Data provider information data goes in here

--boundary1--
```

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

5. Examples

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

```
   [================]   (1)                  [================]
   [   O   +----+   ]   Emergency Call       [                ]
   [  /|\  | UA |------------------------------->             ]
   [   |   +----+   ]   +Device Info         [                ]
   [  / \           ]   +Data Provider Info  [                ]
   [                ]   +Location URI        [                ]
   [ Access Network ]                        [                ]
   [   Provider     ]                        [  VoIP Provider ]
   [                ]                        [   example.org  ]
   [       ^        ]                        [                ]
   [=======.========]                        [============|===]
           .                                              |
           .                                              |
           .                 [================]           |
           .                 [                ]   (2)     |
           .    (3)          [             <--------------+
           ....................>     PSAP     ]  Emergency Call
     Location                [                ]  +Device Info
     +Owner/Subscriber Info  [                ]  +Data Provider Info #2
     +Device Info            [                ]  +Location URI
     +Data Provider Info #3  [================]


     Legend:

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

Figure 11: Additional Data Example Flow

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

```
      INVITE urn:service:sos SIP/2.0
      Via: SIPS/2.0/TLS server.example.com;branch=z9hG4bK74bf9
      Max-Forwards: 70
      To: <urn:service:sos>
      From: Hannes Tschofenig <sips:hannes@example.com>;tag=9fxced76sl
      Call-ID: 3848276298220188511@example.com
      Call-Info: <http://wwww.example.com/hannes/photo.jpg> ;purpose=icon,
        <http://www.example.com/hannes/> ;purpose=info,
        <cid:1234567890@atlanta.example.com>
            ;purpose=EmergencyCallData.ProviderInfo,
        <cide: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"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
       <dev:DataProviderReference>string0987654321@example.org
       </dev:DataProviderReference>
       <dev:DeviceClassification>SoftPhn</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"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
           <pi:id>12345</pi:id>
         <pi:DataProviderReference>string0987654321@example.org
         </pi:DataProviderReference>
         <pi:DataProviderString>Hannes Tschofenig
         </pi:DataProviderString>
       <pi:TypeOfProvider>Other</pi:TypeOfProvider>
       <pi:ContactURI>sip:hannes@example.com</pi:ContactURI>
       <pi:Language>EN</pi:Language>
       <xc:DataProviderContact
           xmlns:xc="urn:ietf:params:xml:ns:vcard-4.0">
                   <vcard>
                       <fn><text>Hannes Tschofenig</text></fn>
                       <n>
                           <surname>Hannes</surname>
                           <given>Tschofenig</given>
                           <additional/>
                           <prefix/>
                           <suffix>Dipl. Ing.</suffix>
                       </n>
                       <bday><date>--0203</date></bday>
                       <anniversary>
                           <date-time>20090808T1430-0500</date-time>
                       </anniversary>
                       <gender><sex>M</sex></gender>
                       <lang>
                           <parameters><pref><integer>1</integer></pref>
                           </parameters>
                           <language-tag>de</language-tag>
                       </lang>
                       <lang>
                           <parameters><pref><integer>2</integer></pref>
                           </parameters>
                           <language-tag>en</language-tag>
                       </lang>
                       <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>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>
    </xc:DataProviderContact>
</pi:EmergencyCallData.ProviderInfo>
   --boundary1--
```

Figure 12: End Device sending SIP INVITE with Additional Data.

In this example, information available to the access network operator 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 operator is not involved in application layer exchanges; the SIP INVITE transits the access network transparently, as illustrated in step #1. No change to the SIP INVITE is applied.

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

```
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"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
       <dev:DataProviderReference>string0987654321@example.org
       </dev:DataProviderReference>
       <dev:DeviceClassification>SoftPhn</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"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
       <pi:DataProviderReference>string0987654321@example.org
       </pi:DataProviderReference>
           <pi:DataProviderString>Hannes Tschofenig
           </pi:DataProviderString>
       <pi:TypeOfProvider>Other</pi:TypeOfProvider>
       <pi:ContactURI>sip:hannes@example.com</pi:ContactURI>
       <pi:Language>EN</pi:Language>
       <xc:DataProviderContact
           xmlns:xc="urn:ietf:params:xml:ns:vcard-4.0">
                   <vcard>
                       <fn><text>Hannes Tschofenig</text></fn>
                       <n>
                           <surname>Hannes</surname>
                           <given>Tschofenig</given>
                           <additional/>
                           <prefix/>
                           <suffix>Dipl. Ing.</suffix>
                       </n>
                       <bday><date>--0203</date></bday>
                       <anniversary>
```

```
    <date-time>20090808T1430-0500</date-time>
</anniversary>
<gender><sex>M</sex></gender>
<lang>
    <parameters><pref><integer>1</integer></pref>
    </parameters>
    <language-tag>de</language-tag>
</lang>
<lang>
    <parameters><pref><integer>2</integer></pref>
    </parameters>
    <language-tag>en</language-tag>
</lang>
<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>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>
       </xc:DataProviderContact>
   </pi:EmergencyCallData.ProviderInfo>

      --boundary1--

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

      --boundary1--

      Content-Type: application/EmergencyCallData.ProviderInfo+xml
      Content-ID: <aaabbb@atlanta.example.com>
      Content-Disposition: by-reference;handling=optional
   <?xml version="1.0" encoding="UTF-8"?>
   <pi:EmergencyCallData.ProviderInfo
       xmlns:pi="urn:ietf:params:xml:ns:EmergencyCallData:ProviderInfo"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
       <pi:DataProviderReference>string0987654321@example.org
       </pi:DataProviderReference>
         <pi:DataProviderString>Example VoIP Provider
         </pi:DataProviderString>
       <pi:ProviderID>urn:nena:companyid:ID123</pi:ProviderID>
       <pi:ProviderIDSeries>NENA</pi:ProviderIDSeries>
       <pi:TypeOfProvider>Service Provider</pi:TypeOfProvider>
       <pi:ContactURI>sip:voip-provider@example.com</pi:ContactURI>
       <pi:Language>EN</pi:Language>
       <xc: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>Example VoIP Provider</text>
                       </org>
                       <adr>
                           <parameters>
                               <type><text>work</text></type>
                               <label><text>John Doe
                                   Downing Street 10
                                   London, UK</text></label>
                           </parameters>
                           <pobox/>
                           <ext/>
                           <street>Downing Street 10</street>
                           <locality>London</locality>
                           <region/>
                           <code>SW1A 2AA</code>
                           <country>UK</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:51.503396, 0.127640</uri>
                    </geo>
                    <tz><text>Europe/London</text></tz>
                    <url>
                        <parameters><type><text>home</text></type>
                        </parameters>
                        <uri>http://www.example.com/john.doe</uri>
                    </url>
                </vcard>
    </xc:DataProviderContact>
</pi:EmergencyCallData.ProviderInfo>
```

Figure 13: VoIP Provider sending SIP INVITE with Additional Data.

Finally, the PSAP requests location information from the access network operator. The response is shown in Figure 14. Along with the location information additional data is provided in the <Provided-By> element of the PIDF-LO.

```
<?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>AU</country>
          <A1>NSW</A1>
          <A3>Wollongong</A3>
          <A4>North Wollongong</A4>
```

```
             <RD>Flinders</RD>
             <STS>Street</STS>
             <RDBR>Campbell Street</RDBR>
             <LMK>Gilligan's Island</LMK>
             <LOC>Corner</LOC>
             <NAM>Video Rental Store</NAM>
             <PC>2500</PC>
             <ROOM>Westerns and Classics</ROOM>
             <PLC>store</PLC>
             <POBOX>Private Box 15</POBOX>
           </civicAddress>
         </gp:location-info>
         <gp:usage-rules>
           <gbp:retransmission-allowed>true
           </gbp:retransmission-allowed>
           <gbp:retention-expiry>2013-12-10T20:00:00Z
           </gbp:retention-expiry>
         </gp:usage-rules>
         <gp:method>802.11</gp:method>

             <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>string0987654321@example.org
             </DataProviderReference>
               <DataProviderString>University of California, Irvine
               </DataProviderString>
             <ProviderID>urn:nena:companyid:uci</ProviderID>
             <ProviderIDSeries>NENA</ProviderIDSeries>
             <TypeOfProvider>Other</TypeOfProvider>
             <ContactURI>tel:+1 9498245222</ContactURI>
             <Language>EN</Language>
           </EmergencyCallData.ProviderInfo>

           <EmergencyCallData.Comment
               xmlns="urn:ietf:params:xml:ns:EmergencyCallData:Comment">
               <DataProviderReference>string0987654321@example.org
               </DataProviderReference>
               <Comment xml:lang="en">This is an example text.</Comment>
           </EmergencyCallData.Comment>

          </EmergencyCallDataValue>
         </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 14: Access Network Provider returning PIDF-LO with Additional Data.

6. XML Schemas

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

6.1. 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/2001/xml.xsd"/>

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

        <xs:simpleType name="iso3166a2">
          <xs:restriction base="xs:token">
            <xs:pattern value="[A-Z]{2}"/>
          </xs:restriction>
        </xs:simpleType>

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

       <xs:element name="DataProviderReference"
           type="xs:token" minOccurs="1" maxOccurs="1"/>
       <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="id"
                       type="xs:string" 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="0" maxOccurs="1"/>

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

                   <xs:element name="Language" type="pi:iso3166a2"
                       minOccurs="0" maxOccurs="unbounded" />

                   <xs:element name="DataProviderContact"
                         type="xc:vcardType" minOccurs="0"
                         maxOccurs="1"/>

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

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

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

   </xs:schema>
```

Figure 15: EmergencyCallData.ProviderInfo XML Schema.

6.2. 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:ServiceInfoTyp
e"/>

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

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

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

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

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

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

   </xs:schema>
```

Figure 16: EmergencyCallData.ServiceInfo XML Schema.

6.3. 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 17: EmergencyCallData.DeviceInfo XML Schema.

6.4. EmergencyCallData.SubscriberInfo XML Schema

```
   <?xml version="1.0"?>
   <xs:schema
       targetNamespace=
                         "urn:ietf:params:xml:ns:EmergencyCallData:SubscriberInf
o"
       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"/>

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

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

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

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

   </xs:schema>
```

Figure 18: EmergencyCallData.SubscriberInfo XML Schema.

6.5. 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 19: EmergencyCallData.Comment XML Schema.

6.6. Provided-By XML Schema

This section defines the Provided-By schema.

```
   <?xml version="1.0"?>
   <xs:schema
       targetNamespace=
       "urn:ietf:params:xml:ns: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:ProviderIn
fo"/>
       <xs:import namespace="urn:ietf:params:xml:ns:EmergencyCallData:ServiceInf
o"/>
       <xs:import namespace="urn:ietf:params:xml:ns:EmergencyCallData:DeviceInfo
"/>
       <xs:import namespace="urn:ietf:params:xml:ns:EmergencyCallData:Subscriber
Info"/>
       <xs:import namespace="urn:ietf:params:xml:ns:EmergencyCallData:Comment"/>

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

       <xs:complexType name="provided-by-Type">
           <xs:sequence>
               <xs:element name="DataProviderReference"
                   type="xs:token" minOccurs="1" maxOccurs="1"/>

               <xs:element name="EmergencyCallDataReference"
                   type="ad:ByRefType"
                   minOccurs="0" maxOccurs="unbounded"/>

               <xs:element name="EmergencyCallDataValue"
                   type="ad:EmergencyCallDataValueType"
                   minOccurs="0" maxOccurs="unbounded"/>

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

           </xs:sequence>

       </xs:complexType>

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

       <!-- Additional Data By Value -->

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

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

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

   </xs:schema>
```

Figure 20: Provided-By XML Schema.

7. Security Considerations

The information in this data structure will usually be considered private. HTTPS is specified to require the provider of the information to validate the credentials of the requester. While the creation of a public key infrastructure (PKI) that has global scope may be difficult, the alternatives to creating devices and services that can provide critical information securely are more daunting. The provider may enforce any policy it wishes to use, but PSAPs and responder agencies should deploy a PKI so that providers of additional data can check the certificate of the client and decide the appropriate policy to enforce based on that certificate.

Ideally, the PSAP and emergency responders will be given credentials signed by an authority trusted by the data provider. In most circumstances, nationally recognized credentials would be sufficient, and if the emergency services arranges a PKI, data providers could be provisioned with the root CA public key for a given nation. Some nations are developing a PKI for this, and related, purposes. Since calls could be made from devices where the device and/or the service provider(s) are not local to the emergency authorities, globally recognized credentials are useful. This might be accomplished by extending the notion of the "forest guide" described in [RFC5222] to allow the forest guide to provide the credential of the PKI root for areas that it has coverage information for, but standards for such a mechanism are not yet available. In its absence, the data provider will need to obtain the root CA credentials for any areas it is willing to provide additional data by out of band means. With the credential of the root CA for a national emergency services PKI, the data provider server can validate the credentials of an entity requesting additional data by reference.

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

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

8. Privacy Considerations

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

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

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

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

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

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

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

Disclosure: When the data defined in this document is not properly security (while in transit with traditional communication security techniques, and while at rest using access control mechanisms) there is the risk of disclosure, which is the revelation of information about an individual that affects the way others judge the individual.

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

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

This document defines various data structures that constitutes personal data. Local regulations may govern what data must be provided in emergency calls, but in general, the emergency call system is often aided by the kinds of information described in this document. There is a tradeoff between the privacy considerations and the utility of the data. For adequate protection this specification requires all data exchanges to be secured via communication security techniques (namely TLS) against eavesdropping and inception. Furthermore, security safeguards are required to prevent unauthorized access to data at rest. Various security incidents over the last 10 years have shown data breaches are not not uncommon and are often caused by lack of proper access control frameworks, software bugs (buffer overflows), or missing input parsing (SQL injection attacks). The risks of data breaches is increased with the obligation for emergency services to retain emergency call related data for extended periods, e.g., several years are the norm.

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

9. IANA Considerations

9.1. Registry creation

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

9.1.1. Provider ID Series Registry

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

The content of this registry includes:

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

Source: The full name of the organization issuing the identifiers

URL: A URL to the organization for further information

The initial set of values is listed in Figure 21.

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

Figure 21: Provider ID Series Registry.

9.1.2. 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.

The content of this registry includes:

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

Description: A short description of the token.

The initial set of values is listed in Figure 22.

Token	Description
Business	[[This RFC]]
Residence	[[This RFC]]

Figure 22: Service Environment Registry.

9.1.3. 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:

Name: The value to be used in TypeOfServiceProvider.

Description: A short description of the type of service provider

The initial set of values is defined in Figure 1.

9.1.4. Service Delivered Registry

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

The content of this registry includes:

Name: Enumeration token of the service.

Description: Short description identifying the service.

The initial set of values are defined in Figure 3.

9.1.5. Device Classification Registry

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

The content of this registry includes:

Name: Enumeration token of the device classification.

Description: Short description identifying the device type.

The initial set of values are defined in Figure 5.

9.1.6. Device ID Type Type Registry

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

The content of this registry includes:

Name: Enumeration token of the device id type.

Description: Short description identifying type of device id.

The initial set of values are defined in Figure 6.

9.1.7. Device/Service Data Type Registry

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

The content of this registry includes:

Name: Enumeration token of the data type.

Description: Short description identifying the the data.

Specification: Citation for the specification of the data.

The initial set of values are listed in Figure 23.

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

Figure 23: Device/Service Data Type Registry.

9.1.8. Additional Data Blocks Registry

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

The content of this registry includes:

Name: Element Name of enclosing block.

Reference: The document that describes the block

The initial set of values are listed in Figure 24.

Token	Reference
ProviderInfo	[This RFC]
ServiceInfo	[This RFC]
DeviceInfo	[This RFC]
Subscriber	[This RFC]

Comment	[This RFC]

Figure 24: Additional Data Blocks Registry.

9.2. ’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].

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

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

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

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

9.4. MIME Registrations

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

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

MIME media type name: application

MIME subtype name: EmergencyCallData.ProviderInfo+xml

Mandatory parameters: none

Optional parameters: charset Indicates the character encoding of enclosed XML.

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

Security considerations: This content type is designed to carry the data provider information, which is a sub-category of additional data about an emergency call. Since this data contains personal information appropriate precautions have to be taken to limit unauthorized access, inappropriate disclosure to third parties, and eavesdropping of this information. Please refer to Section 7 and Section 8 for more information.

Interoperability considerations: None

Published specification: [TBD: This specification]

Applications which use this media type: Emergency Services

Additional information: Magic Number: None File Extension: .xml Macintosh file type code: 'TEXT'

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

Intended usage: LIMITED USE

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

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

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

Mandatory parameters: none

Optional parameters: charset Indicates the character encoding of enclosed XML.

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

Security considerations: This content type is designed to carry the service information, which is a sub-category of additional data about an emergency call. Since this data contains personal information appropriate precautions have to be taken to limit unauthorized access, inappropriate disclosure to third parties, and eavesdropping of this information. Please refer to Section 7 and Section 8 for more information.

Interoperability considerations: None

Published specification: [TBD: This specification]

Applications which use this media type: Emergency Services

Additional information: Magic Number: None File Extension: .xml Macintosh file type code: 'TEXT'

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

Intended usage: LIMITED USE

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

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

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

Mandatory parameters: none

Optional parameters: charset Indicates the character encoding of enclosed XML.

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

Security considerations: This content type is designed to carry the device information information, which is a sub-category of additional data about an emergency call. Since this data contains personal information appropriate precautions have to be taken to limit unauthorized access, inappropriate disclosure to third parties, and eavesdropping of this information. Please refer to Section 7 and Section 8 for more information.

Interoperability considerations: None

Published specification: [TBD: This specification]

Applications which use this media type: Emergency Services

Additional information: Magic Number: None File Extension: .xml Macintosh file type code: 'TEXT'

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

Intended usage: LIMITED USE

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

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

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

Mandatory parameters: none

Optional parameters: charset Indicates the character encoding of enclosed XML.

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

Security considerations: This content type is designed to carry owner/subscriber information, which is a sub-category of additional data about an emergency call. Since this data contains personal information appropriate precautions have to be taken to limit unauthorized access, inappropriate disclosure to third parties, and eavesdropping of this information. Please refer to Section 7 and Section 8 for more information.

Interoperability considerations: None

Published specification: [TBD: This specification]

Applications which use this media type: Emergency Services

Additional information: Magic Number: None File Extension: .xml Macintosh file type code: 'TEXT'

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

Intended usage: LIMITED USE

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

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

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

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

MIME media type name: application

MIME subtype name: EmergencyCallData.Comment+xml

Mandatory parameters: none

Optional parameters: charset Indicates the character encoding of enclosed XML.

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

Security considerations: This content type is designed to carry a comment, which is a sub-category of additional data about an emergency call. This data may contain personal information. Appropriate precautions may have to be taken to limit unauthorized access, inappropriate disclosure to third parties, and eavesdropping of this information. Please refer to Section 7 and Section 8 for more information.

Interoperability considerations: None

Published specification: [TBD: This specification]

Applications which use this media type: Emergency Services

Additional information: Magic Number: None File Extension: .xml Macintosh file type code: 'TEXT'

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

Intended usage: LIMITED USE

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

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

9.5. URN Sub-Namespace Registration

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

9.5.2. 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
```

9.5.3. 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
```

9.5.4. 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
```

9.5.5. 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
```

9.5.6. 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
```

9.6. 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 15.

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 16.

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 17.

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 6.4.

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 6.5.

9.7. VCard Parameter Value Registration

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

Value: main

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

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

Example(s): TEL;VALUE=uri;TYPE="main,voice";PREF=1:tel:+1-418-656-90
00

10. Acknowledgments

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

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

11. References

11.1. Normative References

[RFC0822] Crocker, D., "Standard for the format of ARPA Internet text messages", STD 11, RFC 822, August 1982.

[RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

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

[RFC2392] Levinson, E., "Content-ID and Message-ID Uniform Resource Locators", RFC 2392, August 1998.

[RFC3023] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", RFC 3023, January 2001.

[RFC3204] Zimmerer, E., Peterson, J., Vemuri, A., Ong, L., Audet, F., Watson, M., and M. Zonoun, "MIME media types for ISUP and QSIG Objects", RFC 3204, December 2001.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3325] Jennings, C., Peterson, J., and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", RFC 3325, November 2002.

[RFC3459] Burger, E., "Critical Content Multi-purpose Internet Mail Extensions (MIME) Parameter", RFC 3459, January 2003.

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

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

[RFC4288] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", RFC 4288, December 2005.

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

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

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

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

11.2. Informational References

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

Randy, R., "Negotiating Human Language Using SDP", draft-gellens-negotiating-human-language-02 (work in progress), February 2013.

[I-D.ietf-ecrit-psap-callback]
Schulzrinne, H., Tschofenig, H., Holmberg, C., and M. Patel, "Public Safety Answering Point (PSAP) Callback", draft-ietf-ecrit-psap-callback-13 (work in progress), October 2013.

[I-D.ietf-geopriv-relative-location]
Thomson, M., Rosen, B., Stanley, D., Bajko, G., and A. Thomson, "Relative Location Representation", draft-ietf-geopriv-relative-location-08 (work in progress), September 2013.

[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.

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

[RFC5491] Winterbottom, J., Thomson, M., and H. Tschofenig, "GEOPRIV Presence Information Data Format Location Object (PIDF-LO) Usage Clarification, Considerations, and Recommendations", RFC 5491, March 2009.

[RFC5962] Schulzrinne, H., Singh, V., Tschofenig, H., and M. Thomson, "Dynamic Extensions to the Presence Information Data Format Location Object (PIDF-LO)", RFC 5962, September 2010.

[RFC5985] Barnes, M., "HTTP-Enabled Location Delivery (HELD)", RFC 5985, September 2010.

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

[RFC6848] Winterbottom, J., Thomson, M., Barnes, R., Rosen, B., and R. George, "Specifying Civic Address Extensions in the Presence Information Data Format Location Object (PIDF-LO)", RFC 6848, January 2013.

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

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, July 2013.

Appendix A. XML Schema for vCard/xCard

This section contains the vCard/xCard XML schema version of the Relax NG schema defined in RFC 6351 [RFC6351] for simplified use with the XML schemas defined in this document. The schema in RFC 6351 [RFC6351] is the normative source and this section is informative only.

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

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

    4.1
  -->
```

```
     <xs:element name="text" type="xs:string"/>
     <xs:group name="value-text-list">
       <xs:sequence>
         <xs:element maxOccurs="unbounded" ref="ns1:text"/>
       </xs:sequence>
     </xs:group>
     <!-- 4.2 -->
     <xs:element name="uri" type="xs:anyURI"/>
     <!-- 4.3.1 -->
     <xs:element name="date"
     substitutionGroup="ns1:value-date-and-or-time">
       <xs:simpleType>
         <xs:restriction base="xs:string">
           <xs:pattern value="\d{8}|\d{4}-\d\d|
           --\d\d(\d\d)?|---\d\d"/>
         </xs:restriction>
       </xs:simpleType>
     </xs:element>
     <!-- 4.3.2 -->
     <xs:element name="time"
     substitutionGroup="ns1:value-date-and-or-time">
       <xs:simpleType>
         <xs:restriction base="xs:string">
           <xs:pattern value="(\d\d(\d\d(\d\d)?)?|-\d\d(\d\d?)|--\d\d)
           (Z|[+\-]\d\d(\d\d)?)?"/>
         </xs:restriction>
       </xs:simpleType>
     </xs:element>
     <!-- 4.3.3 -->
     <xs:element name="date-time"
     substitutionGroup="ns1:value-date-and-or-time">
       <xs:simpleType>
         <xs:restriction base="xs:string">
           <xs:pattern value=
           "(\d{8}|--\d{4}|---\d\d)T
           \d\d(\d\d(\d\d)?)?(Z|[+\-]\d\d(\d\d)?)?"/>
         </xs:restriction>
       </xs:simpleType>
     </xs:element>
     <!-- 4.3.4 -->
     <xs:element name="value-date-and-or-time" abstract="true"/>
     <!-- 4.3.5 -->
     <xs:complexType name="value-timestamp">
       <xs:sequence>
         <xs:element ref="ns1:timestamp"/>
       </xs:sequence>
     </xs:complexType>
     <xs:element name="timestamp">
```

```
       <xs:simpleType>
         <xs:restriction base="xs:string">
           <xs:pattern value="\d{8}T\d{6}(Z|[+\-]\d\d(\d\d)?)?"/>
         </xs:restriction>
       </xs:simpleType>
     </xs:element>
     <!-- 4.4 -->
     <xs:element name="boolean" type="xs:boolean"/>
     <!-- 4.5 -->
     <xs:element name="integer" type="xs:integer"/>
     <!-- 4.6 -->
     <xs:element name="float" type="xs:float"/>
     <!-- 4.7 -->
     <xs:element name="utc-offset">
       <xs:simpleType>
         <xs:restriction base="xs:string">
           <xs:pattern value="[+\-]\d\d(\d\d)?"/>
         </xs:restriction>
       </xs:simpleType>
     </xs:element>
     <!-- 4.8 -->
     <xs:element name="language-tag">
       <xs:simpleType>
         <xs:restriction base="xs:string">
           <xs:pattern
           value="([a-z]{2,3}((-[a-z]{3}){0,3})?|[a-z]{4,8})
           (-[a-z]{4})?(-([a-z]{2}|\d{3}))?(-([0-9a-z]{5,8}|
           \d[0-9a-z]{3}))*(-[0-9a-wyz](-[0-9a-z]{2,8})+)*
           (-x(-[0-9a-z]{1,8})+)?|x(-[0-9a-z]{1,8})+|[a-z]{1,3}
           (-[0-9a-z]{2,8}){1,2}"/>
         </xs:restriction>
       </xs:simpleType>
     </xs:element>
     <!--

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

```
           <xs:element ref="ns1:language-tag"/>
         </xs:sequence>
       </xs:complexType>
     </xs:element>
     <!-- 5.2 -->
     <xs:group name="param-pref">
       <xs:sequence>
         <xs:element minOccurs="0" ref="ns1:pref"/>
       </xs:sequence>
     </xs:group>
     <xs:element name="pref">
       <xs:complexType>
         <xs:sequence>
           <xs:element name="integer">
             <xs:simpleType>
               <xs:restriction base="xs:integer">
                 <xs:minInclusive value="1"/>
                 <xs:maxInclusive value="100"/>
               </xs:restriction>
             </xs:simpleType>
           </xs:element>
         </xs:sequence>
       </xs:complexType>
     </xs:element>
     <!-- 5.4 -->
     <xs:group name="param-altid">
       <xs:sequence>
         <xs:element minOccurs="0" ref="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>
```

Authors’ Addresses

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

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

Hannes Tschofenig
(no affiliation)

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

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

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

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

Email: rg+ietf@qti.qualcomm.com

James Winterbottom
(no affiliation)
AU

Email: a.james.winterbottom@gmail.com

ECRIT B. Rosen
Internet-Draft NeuStar, Inc.
Intended status: Standards Track H. Schulzrinne
Expires: January 16, 2014 Columbia U.
H. Tschofenig
Nokia Siemens Networks
July 15, 2013

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

Abstract

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

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

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

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

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

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

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

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

2. Terminology

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

3. Architectural Overview

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

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

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

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

2. Emergency alerts targeted to a Service URN used for IP-based emergency calls where the recipient is not known to the originator. In this scenario, the alert may contain only data (e.g., a CAP and a PIDF-LO payload in a SIP MESSAGE).

Figure 1 shows a deployment variant where a sensor, is pre-configured (using techniques outside the scope of this document) to issue an alert to an aggregator that processes these messages and performs whatever steps are necessary to appropriately react on the alert. For example, a security firm may use different sensor inputs to dispatch their security staff to a building they protect or to initiate a third-party emergency call.

```
    +------------+              +------------+
    | Sensor     |              | Aggregator |
    |            |              |            |
    +---+--------+              +------+-----+
        |                              |
     Sensors                           |
     trigger                           |
     emergency                         |
     alert                             |
        |        MESSAGE with CAP      |
        |----------------------------->|
        |                              |
        |                           Aggregator
        |                           processes
        |                           emergency
        |                           alert
        |        200 (OK)              |
        |<-----------------------------|
        |                              |
        |                              |
```

Figure 1: Targeted Emergency Alert Routing

In Figure 2 a scenario is shown whereby the alert is routed using location information and the Service URN. An emergency services routing proxy (ESRP) may use LoST to determine the next hop proxy to route the alert message to. A possible receiver is a PSAP and the recipient of the alert may be call taker. In the generic case, there is very likely no prior relationship between the originator and the receiver, e.g. PSAP. A PSAP, for example, is likely to receive and accept alerts from entities it cannot authorize. This scenario corresponds more to the classical emergency services use case and the description in [RFC6881] is applicable.

```
                           +-----------+                  +----------+
      +--------+           | ESRP      |                  | PSAP     |
      | Sensor |           |           |                  |          |
      +---+----+           +---+-------+                  +---+------+
          |                    |                              |
       Sensors                 |                              |
       trigger                 |                              |
       emergency               |                              |
       alert                   |                              |
          |                    |                              |
          |                    |                              |
          | MESSAGE with CAP   |                              |
          | (including Service URN,                           |
          | such as urn:service:sos)                          |
          |------------------->|                              |
          |                    |                              |
          |              ESRP performs                        |
          |              emergency alert                      |
          |              routing                              |
          |                    |  MESSAGE with CAP            |
          |                    |  (including identity info)   |
          |                    |----------------------------->|
          |                    |                              |
          |                    |                           PSAP
          |                    |                           processes
          |                    |                           emergency
          |                    |                           alert
          |                    |        200 (OK)              |
          |                    |<-----------------------------|
          |                    |                              |
          |  200 (OK)          |                              |
          |<-------------------|                              |
          |                    |                              |
          |                    |                              |
```

Figure 2: Location-Based Emergency Alert Routing

4. Protocol Specification

4.1. CAP Transport

A CAP message may be sent on the initial message of any SIP transaction. However, this document only describes specific behavior when used with a SIP MESSAGE transaction for a one-shot, data-only emergency call. Behavior with other transactions is not defined.

The CAP message included in a SIP message as an additional-data block [I-D.ietf-ecrit-additional-data]. Accordingly, it is introduced to the SIP message with a Call-Info header with a purpose of "emergencyCall.cap". The header may contain a URI that is used by the recipient (or in some cases, an intermediary) to obtain the CAP message. Alternative, the Call-Info header may contain a Content Indirect url [RFC2392] and the CAP message included in the body of the message. In either case, the CAP message is located in a MIME block. The MIME type is set to 'application/emergencyCall.cap+xml'.

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

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

4.2. Profiling of the CAP Document Content

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

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

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

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

Author indication relevant: In case the author is different from the actual originator of the message and this distinction should be preserved then the <sender> element MUST NOT contain the SIP URI of the user agent.

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

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

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

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

4.3. Sending a Data-Only Emergency Call

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

5. Error Handling

This section defines a new error response code and a header field for additional information.

5.1. 425 (Bad Alert Message) Response Code

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

425 (Bad Alert Message)

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

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

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

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

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

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

5.2. The AlertMsg-Error Header Field

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

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

```
   message-header          /= AlertMsg-Error
                              ; (message-header from 3261)
   AlertMsg-Error        = "AlertMsg-Error" HCOLON
                                ErrorValue
   ErrorValue       =  error-code
                               *(SEMI error-params)
   error-code      = 1*3DIGIT
   error-params    = error-code-text
                              / generic-param ; from RFC3261
   error-code-text = "code" EQUAL quoted-string ; from RFC3261
```

HCOLON, SEMI, and EQUAL are defined in RFC3261 [RFC3261]. DIGIT is defined in RFC5234 [RFC5234].

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

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

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

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

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

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

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

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

AlertMsg-Error: 102 ; code="Not enough information to determine the purpose of the alert"

AlertMsg-Error: 103 ; code="Alert Payload was corrupted"

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

6. Updates to the CAP Message

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

7. Example

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

```
MESSAGE sip:aggregator@domain.com SIP/2.0
Via: SIP/2.0/TCP sensor1.domain.com;branch=z9hG4bK776sgdkse
Max-Forwards: 70
From: sip:sensor1@domain.com;tag=49583
To: sip:aggregator@domain.com
Call-ID: asd88asd77a@1.2.3.4
```

```
      Geolocation: <cid:abcdef@domain.com>
        ;routing-allowed=yes
      Supported: geolocation
      Accept: application/pidf+xml, application/emergencyCall.cap+xml
      CSeq: 1 MESSAGE
      Call-Info: cid:abcdef2@domain.com;purpose=emergencyCall.cap
      Content-Type: multipart/mixed; boundary=boundary1
      Content-Length: ...

      --boundary1

      Content-Type: application/emergencyCall.cap
      Content-ID: <abcdef2@domain.com>
      Content-Disposition: by-reference;handling=optional
     <?xml version="1.0" encoding="UTF-8"?>

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

      --boundary1

      Content-Type: application/pidf+xml
      Content-ID: <abcdef2@domain.com>
      Content-Disposition: by-reference;handling=optional
      <?xml version="1.0" encoding="UTF-8"?>
```

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

Figure 3: Example Message conveying an Alert to an Aggregator

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

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

```
      Call-info: cid:abcdef2@domain.com;purpose=emergencyCall.cap
      CSeq: 1 MESSAGE
      Content-Type: multipart/mixed; boundary=boundary1
      Content-Length: ...

      --boundary1

      Content-Type: application/emergencyCall.cap+xml
      Content-ID: <abcdef2@example.com>
     <?xml version="1.0" encoding="UTF-8"?>

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

      --boundary1

      Content-Type: application/pidf+xml
      Content-ID: <abcdef2@domain.com>
      <?xml version="1.0" encoding="UTF-8"?>
          <presence
             xmlns="urn:ietf:params:xml:ns:pidf"
             xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
             xmlns:gbp=
                    "urn:ietf:params:xml:ns:pidf:geopriv10:basicPolicy"
             xmlns:cl="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr"
             xmlns:gml="http://www.opengis.net/gml"
             xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model"
             entity="pres:alice@atlanta.example.com">
           <dm:device id="sensor">
             <gp:geopriv>
               <gp:location-info>
                 <gml:location>
                   <gml:Point srsName="urn:ogc:def:crs:EPSG::4326">
                     <gml:pos>32.86726 -97.16054</gml:pos>
                   </gml:Point>
                </gml:location>
               </gp:location-info>
               <gp:usage-rules>
                 <gbp:retransmission-allowed>false
                 </gbp:retransmission-allowed>
                 <gbp:retention-expiry>2010-11-14T20:00:00Z
                 </gbp:retention-expiry>
               </gp:usage-rules>
               <gp:method>802.11</gp:method>
             </gp:geopriv>
             <dm:timestamp>2010-11-04T20:57:29Z</dm:timestamp>
           </dm:device>
         </presence>
      --boundary1--
```

Figure 4: Example Message conveying an Alert to a PSAP

8. Security Considerations

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

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

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

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

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

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

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

9. IANA Considerations

9.1. Registration of the 'application/emergencyCall.cap+xml' MIME type

To: ietf-types@iana.org

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

MIME media type name: application

MIME subtype name: cap+xml

Required parameters: (none)

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

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

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

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

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

Applications which use this media type: Applications that convey alerts and warnings according to the CAP standard.

Additional information: OASIS has published the Common Alerting Protocol at http://www.oasis-open.org/committees/ documents.php&wg_abbrev=emergency

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

Intended usage: Limited use

Author/Change controller: IETF ECRIT working group

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

9.2. IANA Registration of Additional Data Block

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

9.3. IANA Registration for 425 Response Code

In the SIP Response Codes registry, the following is added

Reference: RFC-XXXX (i.e., this document)

Response code: 425 (recommended number to assign)

Default reason phrase: Bad Alert Message

```
Registry:
  Response Code                                Reference
  -------------------------------------------  ---------
  Request Failure 4xx
    425 Bad Alert Message                     [this doc]
```

This SIP Response code is defined in Section 5.

9.4. IANA Registration of New AlertMsg-Error Header Field

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

1. Update the Header Fields registry with

Registry:

Header Name	compact	Reference
AlertMsg-Error		[this doc]

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

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

9.5. IANA Registration for the SIP AlertMsg-Error Codes

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

Registry Name: AlertMsg-Error Codes

Reference: [this doc]

Registration Procedures: Specification Required

Code	Default Reason Phrase	Reference
100	"Cannot Process the Alert Payload"	[this doc]
101	"Alert Payload was not present or could not be found"	[this doc]

```
   102  "Not enough information to determine
         the purpose of the alert"                        [this doc]

   103  "Alert Payload was corrupted"                      [this doc]
```

Details of these error codes are in Section 5.

10. Acknowledgments

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

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

11. References

11.1. Normative References

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

[cap] Jones, E. and A. Botterell, "Common Alerting Protocol v. 1.1 ", October 2005.

[RFC2392] Levinson, E., "Content-ID and Message-ID Uniform Resource Locators", RFC 2392, August 1998.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3428] Campbell, B., Rosenberg, J., Schulzrinne, H., Huitema, C., and D. Gurle, "Session Initiation Protocol (SIP) Extension for Instant Messaging", RFC 3428, December 2002.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC3903] Niemi, A., "Session Initiation Protocol (SIP) Extension for Event State Publication", RFC 3903, October 2004.

[RFC3023] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", RFC 3023, January 2001.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

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

[RFC6665] Roach, A., "SIP-Specific Event Notification", RFC 6665, July 2012.

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

[I-D.ietf-ecrit-additional-data]
Rosen, B., Tschofenig, H., Marshall, R., Randy, R., and J. Winterbottom, "Additional Data related to an Emergency Call", draft-ietf-ecrit-additional-data-10 (work in progress), July 2013.

[I-D.rosen-ecrit-addldata-subnot]
Rosen, B., "Updating Additional Data related to an Emergency Call using Subscribe/ Notify", draft-rosen-ecrit-addldata-subnot-00 (work in progress), July 2013.

11.2. Informative References

[I-D.ietf-ecrit-trustworthy-location]
Tschofenig, H., Schulzrinne, H., and B. Aboba, "Trustworthy Location", draft-ietf-ecrit-trustworthy-location-06 (work in progress), July 2013.

[RFC4474] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, August 2006.

[RFC3325] Jennings, C., Peterson, J., and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", RFC 3325, November 2002.

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

Authors' Addresses

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

Email: br@brianrosen.net

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

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

Hannes Tschofenig
Nokia Siemens Networks
Linnoitustie 6
Espoo 02600
Finland

Phone: +358 (50) 4871445
Email: Hannes.Tschofenig@gmx.net
URI: http://www.tschofenig.priv.at

ECRIT
Internet-Draft
Intended status: Standards Track
Expires: August 18, 2014

R. Marshall
J. Martin
TCS
B. Rosen
Neustar

February 14, 2014

A LoST extension to support return of complete and similar location info
draft-marshall-ecrit-similar-location-03

Abstract

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

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

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

Whether the input civic location is valid and missing information, or invalid due to missing or wrong information during input, this document provides a mechanism to return a complete set of location information for those valid or invalid cases.

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

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

2. Terminology

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

The following terms are defined in this document:

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

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

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

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

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

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

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

3. Overview of Returned Location Information

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

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

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

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

In valid location responses, when a LoST server returns a response to a findService request that contains a set of CAtype elements considered valid, the location information in the findServiceResponse is extended to include additional location information specific for that location. As an example, the query may contain a HNO (house number), RD (road name) and A3 (city) but may not contain A1, A2, PC (Postal Code) CAtypes. The RD and PC elements may be sufficient to locate the address specified in the request and thus be considered valid. Yet, downstream entities may find it helpful to have the additional A1, A2, and PC location elements that exist, and so the mechanism described here supports their inclusion. Since [RFC5222] currently does not have a way for this additional location information to be returned in the findServiceResponse, this document extends RFC5222 so that it can include a completeLocation element within the findServiceResponse message, representing a "complete" civic location.

input address: 6000 15th Ave NW Seattle

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

When invalid location responses are received, the same mechanism works as follows: when a LoST server returns a response to a findService request that contains a set of CAtype elements with one or more that are tagged as invalid, the location information in the findServiceResponse is extended to include additional location information specific for that location. Differing results in the same data used in the above example, where the RD and PC elements are not sufficient to locate a unique address leads to an "invalid" result. This is the case, despite the fact that the LoST server typically contains additional location elements which could have resulted in a uniquely identifiable location if additional data had been supplied in the query. Since [RFC5222] currently does not have a way for this additional location information to be returned in the findServiceResponse, this document extends RFC5222 so that it can include one or more similarLocation elements within the findServiceResponse message representing "similar" civic locations.

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

input address: 6000 15th Ave Seattle, WA.

Different from the above case, this time we make the assumption that the address is deemed "invalid" by the LoST server because there is no plain "15th Ave" in the city of Seattle with a house number that matches 6000. However there are two addresses within the address dataset that are "similar", when all parts of the address are taken as a whole. These similar addresses that could be suggested to the user are as follows:

similar address #1: 6000 15th Ave NW Seattle, WA 98107 similar address #2: 6000 15th Ave NE Seattle, WA 98105

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

4. Returned Location Information

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

5. Complete Location returned for Valid response

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

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


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

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

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

    </civicAddress>
  </location>

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

</findService>
```

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


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

        <mapping
          expires="NO-CACHE"
          lastUpdated="2006-11-01T01:00:00Z"
          source="authoritative.example"
          sourceId="8799e346000098aa3e">

          <displayName xml:lang="en">Seattle 911</displayName>
          <service>urn:service:sos</service>
          <uri>sip:seattle-911@example.com</uri>
          <serviceNumber>911</serviceNumber>

        </mapping>

        <locationValidation

          <valid>ca:A3 ca:A6 ca:STS ca:POD ca:HNO</valid>
          <invalid></invalid>
          <unchecked></unchecked>

          <rli:completeLocation>  <!-- completed address -->
            <ca:civicAddress>
              <ca:country>US</ca:country>
              <ca:A1>WA</ca:A1>
              <ca:A3>SEATTLE</ca:A3>
              <ca:RD>15TH</ca:RD>
              <ca:STS>AVE</ca:STS>
              <ca:POD>NW</ca:POD>
              <ca:HNO>6000</ca:HNO>
              <ca:PC>98106</ca:PC>
              <ca:PCN>SEATTLE</ca:PCN>
            </ca:civicAddress>

        </rli:completeLocation>

      </locationValidation>

        <path>
          <via source="authoritative.example"/>
        </path>
        <locationUsed id="587cd3880"/>

      </findServiceResponse>


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

6. Similar Location returned for Invalid Response

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

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


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

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

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

          </civicAddress>
        </location>

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

      </findService>

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


      <findServiceResponse>
        xmlns="urn:ietf:params:xml:ns:lost1"
        xmlns:rli="urn:ietf:params:xml:ns:lost-rli1">
        xmlns:ca="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr">
        <mapping
          expires="NO-CACHE"
          lastUpdated="2006-11-01T01:00:00Z"
          source="authoritative.example"
          sourceId="8799e346000098aa3e">

          <displayName xml:lang="en">Seattle 911</displayName>
          <service>urn:service:sos</service>
          <uri>sip:seattle-911@example.com</uri>
          <serviceNumber>911</serviceNumber>

        </mapping>

        <locationValidation

          <valid>ca:country ca:A1 ca:A3</valid>
          <invalid>ca:A6</invalid>
          <unchecked>ca:HNO</unchecked>

          <rli:similarLocation>  <!-- similar location info -->
            <ca:civicAddress>  <!-- similar address #1 -->
              <ca:country>US</ca:country>
              <ca:A1>WA</ca:A1>
              <ca:A3>SEATTLE</ca:A3>
              <ca:RD>15TH</ca:RD>
              <ca:STS>AVE</ca:STS>
              <ca:POD>NW</ca:POD>
              <ca:HNO>6000</ca:HNO>
              <ca:PC>98106</ca:PC>
              <ca:PCN>SEATTLE</ca:PCN>
            </ca:civicAddress>

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

      </locationValidation>

        <path>
```

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

        <locationUsed id="587cd3880"/>

      </findServiceResponse>


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

7. Relax NG schema

This section provides the Relax NG schema of LoST extensions in the compact form. The verbose form is included in a later section [TBA].

```
  namespace a = "http://relaxng.org/ns/compatibility/annotations/1.0"
  default namespace ns1 = "urn:ietf:params:xml:ns:lost-similarLocation1"

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

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



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

  ##
  ##       similarLocation
  ##
  div {
    similarLocation =
      element location {
        attribute id { xsd:token },
        locationInformation
      }+
  }
  ##
  ##       Location Information
  ##
  div {
    locationInformation =
      extensionPoint+,
      attribute profile { xsd:NMTOKEN }?
  }

  ##
  ##       Patterns for inclusion of elements from schemas in
  ##       other namespaces.
  ##
  div {

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

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

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

8. Security Considerations

Whether the input to the LoST server is valid or invalid, the LoST server ultimately determines what it considers to be valid. In the case where the input location is valid, the requester still may not actually understand where that location is. For valid location use cases, this extension returns more location information than the requester may have had which, in turn, may reveal more about the location. While this may be very desirable when, for example, supporting an emergency call, it may not be as desirable for other services. The LoST server implementation should consider the risk of releasing more detail verses the value in doing so. Generally, we do not believe this is a significant problem as the requester must have enough location information to be considered valid, which in most cases is enough to uniquely locate the address. Providing more CAtypes generally doesn’t actually reveal anything more.

9. IANA Considerations

9.1. Relax NG Schema Registration

URI: urn:ietf:params:xml:schema:lost-similarLocation1

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

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

```
default namespace = "urn:ietf:params:xml:ns:lost-similarLocation1
```

and its last line is

```
}
```

9.2. LoST Namespace Registration

URI: urn:ietf:params:xml:ns:lost-similarLocation1

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 Similar Location extension</h1>
  <h2>urn:ietf:params:xml:ns:lost-similarLocation1</h2>
<p>See <a href="http://www.rfc-editor.org/rfc/rfc????.txt">
   RFC????</a>.</p>
</body>
</html>
END
```

10. Acknowledgements

11. References

11.1. Normative References

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

11.2. Informative References

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

Authors’ Addresses

Roger Marshall
TeleCommunication Systems, Inc.
2401 Elliott Avenue
2nd Floor
Seattle, WA 98121
US

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

Jeff Martin
TeleCommunication Systems, Inc.
2401 Elliott Avenue
2nd Floor
Seattle, WA 98121
US

Phone: +1 206 792 2584
Email: jmartin@telecomsys.com
URI: http://www.telecomsys.com

Brian Rosen
Neustar
470 Conrad Dr
Mars, PA 16046
US

Email: br@brianrosen.net

ecrit
Internet-Draft
Intended status: Standards Track
Expires: August 18, 2014

B. Rosen
Neustar

February 14, 2014

Validation of Locations Around a Planned Change
draft-rosen-ecrit-lost-planned-changes-01

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 a TTL element to the response, which informs all queriers the current expected lifetime of the validation.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document describes an update to the LoST protocol [RFC5222] which allows a <findservice> request to add a URI and a date to be used with planned changes to the underlying location information in the server which is used by the validation function. 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. The lt;findserviceResponse> is 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 (typically 00:00 on some date).

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. There are other circumstances where locations that were previously valid become invalid. 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, 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 provided that the location information in the request was valid. 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, the LIS will prepare a new record, to be inserted in its active database, that becomes valid 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 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> when validation is requested.

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.

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 sends <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.

5. 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.

6. Relax NG Schema

The Relax NG schema in [RFC5222] is modified to be:

```
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*
   }
```

7. 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 can 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.

8. IANA Considerations

8.1. 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

```
}
```

8.2. 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
```

9. Normative References

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

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

Author’s Address

Brian Rosen
Neustar
470 Conrad Dr
Mars, PA 16046
US

EMail: br@brianrosen.net