

ECRIT
Internet-Draft
Intended status: Informational
Expires: April 20, 2016

R. Gellens
Qualcomm Technologies, Inc
B. Rosen
NeuStar, Inc.
H. Tschofenig
(Individual)
October 18, 2015

Next-Generation Vehicle-Initiated Emergency Calls
draft-ietf-ecrit-car-crash-04.txt

Abstract

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

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

This document reuses the technical aspects of next-generation pan-European eCall (a mandated and standardized system for emergency calls by in-vehicle systems within Europe and other regions). However, this document specifies a different set of vehicle (crash) data, specifically, the Vehicle Emergency Data Set (VEDS) rather than the eCall Minimum Set of Data (MSD). This document is an extension of the eCall document, with the differences being that this document makes the MSD data set optional and VEDS mandatory. This document also discusses legacy (circuit-switched) ACN systems and their migration to next-generation emergency calling.

Status of This Memo

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

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

working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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

This Internet-Draft will expire on April 20, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Terminology	3
2. Introduction	3
3. Document Scope	7
4. Overview of Legacy Deployment Models	8
5. Migration to Next-Generation	9
6. Profile	12
7. Call Setup	12
8. Call Routing	15
9. Test Calls	16
10. Example	16
11. Security Considerations	21
12. Privacy Considerations	21
13. IANA Considerations	21
13.1. MIME Content-type Registration for 'application/EmergencyCall.VEDS+xml'	21
13.2. Registration of the 'VEDS' entry in the Emergency Call Additional Data registry	22
14. Contributors	23
15. Acknowledgements	23
16. Changes from Previous Versions	23
16.1. Changes from draft-ietf-03 to draft-ietf-04	23

16.2.	Changes from draft-ietf-02 to draft-ietf-03	23
16.3.	Changes from draft-ietf-01 to draft-ietf-02	23
16.4.	Changes from draft-ietf-00 to draft-ietf-01	23
16.5.	Changes from draft-gellens-02 to draft-ietf-00	23
16.6.	Changes from draft-gellens-01 to -02	24
16.7.	Changes from draft-gellens-00 to -01	24
17.	References	24
17.1.	Normative References	24
17.2.	Informative references	25
	Authors' Addresses	26

1. Terminology

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

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

Additionally, we use the following abbreviations:

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

2. Introduction

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

Drivers often have a poor location awareness, especially outside of major cities, at night and when away from home (especially abroad). In the most crucial cases, the victim(s) might not be able to call because they have been injured or trapped.

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

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

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

The transition to next-generation calling in general, and emergency calling in particular, provides an opportunity to vastly improve the scope, breadth, reliability and usefulness of crash data during an emergency by allowing it to be presented alongside the call, and to be automatically processed by the PSAP and made available to the call taker in an integrated, automated way. In addition, vehicle

manufacturers are provided an opportunity to take advantage of the same standardized mechanisms for data transmission for internal use if they wish (such as telemetry between the vehicle and a service center for both emergency and non-emergency uses, including location-based services, multi-media entertainment systems, and road-side assistance applications).

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

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

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

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

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 might be needed. This information was recently included in the federal guidelines for field triage of injured patients. These guidelines are designed to help responders at the accident scene identify the potential existence of severe internal injuries and to make critical decisions about how and where a patient needs to be transported.

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

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

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

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

A next-generation In-Vehicle System (IVS) transmits crash data by encoding it in a standardized and registered format (such as VEDS)

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

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

3. Document Scope

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

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

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

4. Overview of Legacy Deployment Models

Legacy (circuit-switched) systems for placing emergency calls by in-vehicle systems, including automatic crash notification systems, generally have some 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". 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. Since the TSP knows the location of the vehicle (from on-board GNSS), location-based routing is usually used to route to the appropriate PSAP. In some cases, the TSP is able to transmit location automatically, using similar techniques as for wireless calls. 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 might or might not be conveyed also using text-to-speech in an initial voice greeting. Some such systems use an automated voice prompt menu for the PSAP call taker (e.g., "this is an automatic emergency call from a vehicle; press 1 to open a voice path to the vehicle; press 2 to hear the location read out") to allow the call taker to request location data via text-to-speech.

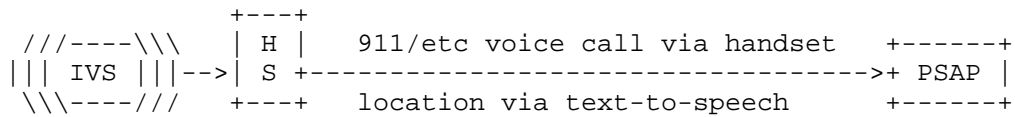


Figure 2: Legacy Paired Model

In the direct model, the IVS directly places an emergency call with the PSAP by dialing a standard emergency number such as 9-1-1. Such systems might communicate location data to the PSAP via text-to-speech; crash data might or might not be conveyed using text-to-speech in an initial voice greeting. Some such systems use an automated voice prompt menu (e.g., "this is an automatic emergency call from a vehicle; press 1 to open a voice path to the vehicle; press 2 to hear the location read out") to allow the call taker to request location data via text-to-speech.

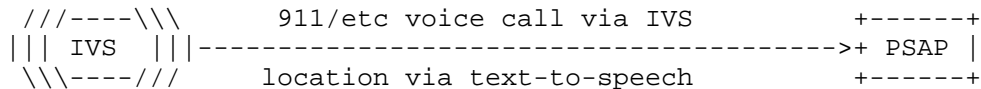


Figure 3: Legacy Direct Model

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, as well as enabling additional communications modalities and enhanced functionality. This allows ACN calls and crash data to be automatically processed by the PSAP and made available to the call taker in an integrated, automated way. Because the crash data is carried in the initial SIP INVITE (per [I-D.ietf-ecrit-additional-data]) the PSAP can present it to the call taker simultaneously with the appearance of the call.

Migration to next-generation (NG) thus provides an opportunity to significantly improve the handling and response to vehicle-initiated emergency calls. Such calls can be recognized as originating from a vehicle, routed to a PSAP equipped both technically and operationally to handle such calls, and the vehicle-determined location and crash data can be made available to the call taker simultaneously with the call appearance.

Vehicle manufacturers using the TSP model can 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 as are used to convey this data to the PSAP.

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

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

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

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

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

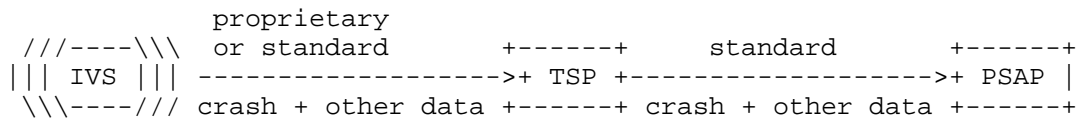


Figure 4: Next-Generation TSP Model

The vehicle manufacturer and the TSP can choose to use the same mechanisms and data objects to transmit crash and location data from the vehicle to the TSP as are described here to transmit such data from to the PSAP.

In the direct model, the IVS communicates crash data to the PSAP directly using the mechanisms and data objects described here.

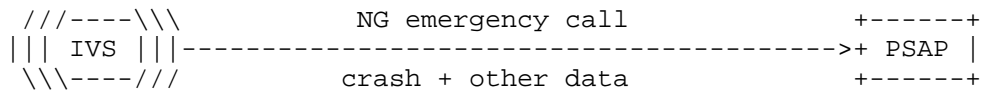


Figure 5: Next-Generation Direct Model

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 undefined what facilities are or will be available for transmitting crash data through the Bluetooth link to the handset for inclusion in an NG emergency call. Hence, manufacturers that use the paired model for legacy calls might choose to adopt either the direct or TSP models for next-generation calls.

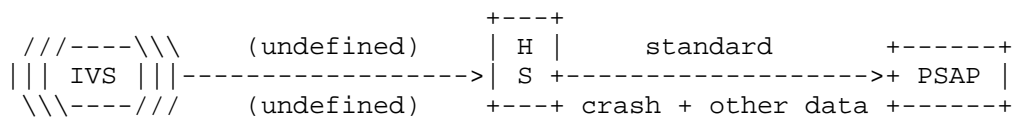


Figure 6: Next-Generation Paired Model

If the call is routed to a PSAP that is not capable of processing the vehicle data, the PSAP ignores (or does not receive) the vehicle data. This is detectable by the IVS or TSP when it receives a 200 OK to the INVITE which lacks an eCall control structure acknowledging receipt of the data [I-D.ietf-ecrit-ecall]. The IVS or TSP then proceeds as it would for a non-NG ACN call (e.g., verbal conveyance of data)

6. Profile

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

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

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

7. Call Setup

It is important that ACN calls be easily identifiable as such at all stages of call handling, and that automatic versus manual triggering be known. ACN calls differ from general emergency calls in several aspects, including the presence of standardized crash data, the fact that the call is known to be placed by an in-vehicle system (which has implications for PSAP operational processes), and, especially for automatic calls, information that can 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 can be considered a specific sub-class of general emergency calls and are optimally handled by a PSAP with the technical and operational capabilities to serve such calls. (This is especially so in environments such as the U.S. where there are many PSAPs and where individual PSAPs have a range of capabilities.) Technical capabilities include the ability to recognize and process standardized crash data. Operational capabilities include training and processes for assessing severe injury likelihood and responding

appropriately (e.g., dispatching trauma-capable medical responders or those trained and equipped to extract occupants from crashed vehicles and handle gasoline or other hazardous materials, transporting victims to a trauma center, alerting the receiving facility, etc.).

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

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

Specifically, the steps required during standardization are:

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

- o The item is registered in the Emergency Call Additional Data registry, as defined in Section 9.1.7 of [I-D.ietf-ecrit-additional-data]
- * For emergency-call-specific formats, the registered name is the root of the MIME Content-Type (not including the 'EmergencyCallData' prefix and any suffix such as '+xml') as described in Section 4.1 of [I-D.ietf-ecrit-additional-data]

When placing an emergency call:

- o The crash data set is created and encoded per its specification
- o The crash data set is attached to the emergency call INVITE as specified in Section 4.1 of [I-D.ietf-ecrit-additional-data], that is, as a MIME body part identified by its MIME Content-Type in the body part's Content-Type header field
- o The body part is assigned a unique identifier label in a Content-ID header field of the body part
- o A Call-Info header field at the top level of the INVITE is added that references the crash data and identifies it by its MIME root (as registered in the Emergency Call Additional Data registry)
 - * The crash data is referenced in the Call-Info header field by a CID URL that contains the unique Content ID assigned to the crash data body part
 - * The crash data is identified in the Call-Info header field by a 'purpose' parameter whose value is 'EmergencyCallData.' concatenated with the specific crash data entry in the Emergency Call Additional Data registry
 - * The Call-Info header field MAY be either solely to reference the crash data (and hence have only the one URL) or can also contain other URLs referencing other data
- o 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.

This document extends [I-D.ietf-ecrit-ecall] by reusing the call set-up and other normative requirements except that in this document, support for the eCall MSD is OPTIONAL and support for VEDS in REQUIRED.

8. Call Routing

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

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

9. Test Calls

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

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

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

10. Example

Figure 7 shows an emergency call placed by a vehicle whereby location information and VEDS crash data are both attached to the SIP INVITE message. The INVITE has a request URI containing the 'urn:service:sos.ecall.automatic' service URN and is thus recognized as an ACN type of emergency call, and is also recognizable as an 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). 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. (In deployments where there is no ESInet, the MNO itself routes the call directly to an appropriate ACN-capable PSAP.)

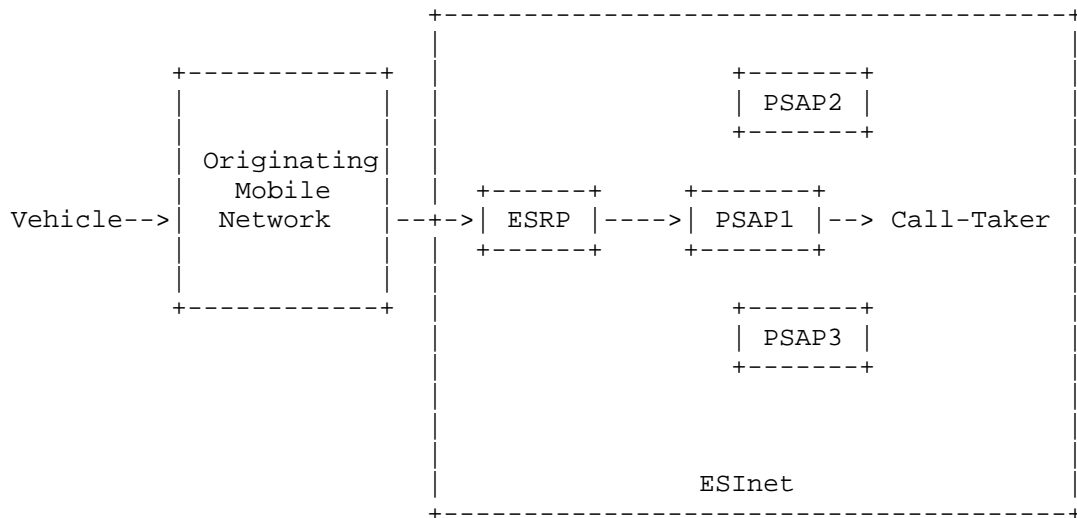


Figure 7: Example of Vehicle-Placed Emergency Call Message Flow

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

The example VEDS data structure shows information about a crashed vehicle. The example communicates that the car is a model year 2015 Saab 9-5 (a car which does not exist). The front airbag deployed as a consequence of the crash. The 'VehicleBodyCategoryCode' indicates that the crashed vehicle is a passenger car (the code is set to '101') and that it is not a convertible (the 'ConvertibleIndicator' value is set to 'false').

The 'VehicleCrashPulse' element provides further information about the crash, namely that the force of impact based on the change in velocity over the duration of the crash pulse was 100 MPH. The principal direction of the force of the impact is set to '12' (which refers to 12 O'Clock, corresponding to a frontal collision). This value is described in the 'CrashPulsePrincipalDirectionOfForceValue' element.

The 'CrashPulseRolloverQuarterTurnsValue' indicates the number of quarter turns in concert with a rollover expressed as a number; in our case 1.

No roll bar was deployed, as indicated in 'VehicleRollbarDeployedIndicator' being set to 'false'.

Next, there is information indicating seatbelt and seat sensor data for individual seat positions in the vehicle. In our example, information from the driver seat is available (value '1' in the 'VehicleSeatLocationCategoryCode' element), that the seatbelt was monitored ('VehicleSeatbeltMonitoredIndicator' element), that the seatbelt was fastened ('VehicleSeatbeltFastenedIndicator' element) and the seat sensor determined that the seat is occupied ('VehicleSeatOccupiedIndicator' element).

Finally, information about the weight of the vehicle, which is 600 kilogram in our example.

In addition to the information about the vehicle, further indications are provided, namely the presence of fuel leakage ('FuelLeakingIndicator' element), an indication whether the vehicle was subjected to multiple impacts ('MultipleImpactsIndicator' element), the orientation of the vehicle at final rest ('VehicleFinalRestOrientationCategoryCode' element) and an indication that there are no parts of the vehicle on fire (the 'VehicleFireIndicator' element).

```
INVITE urn:service:sos.ecall.automatic SIP/2.0
To: urn:service:sos.ecall.automatic
From: <sip:+13145551111@example.com>;tag=9fxced76s1
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
Content-Disposition: by-reference;handling=optional

<?xml version="1.0" encoding="UTF-8"?>
<AutomatedCrashNotification xmlns="http://www.veds.org/acn/1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

<Crash>
  <CrashVehicle>
    <ItemMakeName xmlns="http://niem.gov/niem/niem-core/2.0">
      Saab
    </ItemMakeName>
    <ItemModelName xmlns="http://niem.gov/niem/niem-core/2.0">
      9-5
    </ItemModelName>
    <ItemModelYearDate
      xmlns="http://niem.gov/niem/niem-core/2.0">
      2015
    </ItemModelYearDate>
    <Airbag>
      <AirbagCategoryCode>FRONT</AirbagCategoryCode>
      <AirbagDeployedIndicator>true
    </AirbagDeployedIndicator>
```

```
</Airbag>
<ConvertibleIndicator>false</ConvertibleIndicator>
<PowerSourceCategoryCode>MAIN</PowerSourceCategoryCode>
<VehicleBodyCategoryCode
  xmlns="http://niem.gov/niem/domains/jxdm/4.1">
  101
</VehicleBodyCategoryCode>
<VehicleCrashPulse>
  <CrashPulseChangeInVelocityMeasure>
    <MeasurePointValue
      xmlns="http://niem.gov/niem/niem-core/2.0">
      100
    </MeasurePointValue>
    <MeasureUnitText
      xmlns="http://niem.gov/niem/niem-core/2.0">
      MPH</MeasureUnitText>
    </CrashPulseChangeInVelocityMeasure>
    <CrashPulsePrincipalDirectionOfForceValue>12
    </CrashPulsePrincipalDirectionOfForceValue>
    <CrashPulseRolloverQuarterTurnsValue>1
    </CrashPulseRolloverQuarterTurnsValue>
  </VehicleCrashPulse>
<VehicleRollbarDeployedIndicator>false
</VehicleRollbarDeployedIndicator>
<VehicleSeat>
  <VehicleSeatLocationCategoryCode>1
  </VehicleSeatLocationCategoryCode>
  <VehicleSeatOccupiedIndicator>true
  </VehicleSeatOccupiedIndicator>
  <VehicleSeatbeltFastenedIndicator>true
  </VehicleSeatbeltFastenedIndicator>
  <VehicleSeatbeltMonitoredIndicator>true
  </VehicleSeatbeltMonitoredIndicator>
</VehicleSeat>
<VehicleUnladenWeightMeasure
  xmlns="http://niem.gov/niem/niem-core/2.0">
  <MeasurePointValue
    xmlns="http://niem.gov/niem/niem-core/2.0">
    600
  </MeasurePointValue>
  <MeasureUnitText
    xmlns="http://niem.gov/niem/niem-core/2.0">
    kilogram
  </MeasureUnitText>
</VehicleUnladenWeightMeasure>
</CrashVehicle>
<FuelLeakingIndicator>true</FuelLeakingIndicator>
<MultipleImpactsIndicator>false</MultipleImpactsIndicator>
```

```
<SevereInjuryIndicator>true</SevereInjuryIndicator>
<VehicleFinalRestOrientationCategoryCode>Driver
</VehicleFinalRestOrientationCategoryCode>
<VehicleFireIndicator>false</VehicleFireIndicator>
</Crash>
</AutomatedCrashNotification>
```

--boundary1--

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

11. Security Considerations

This document does not raise security considerations beyond those described in [RFC5069]. As with emergency service systems with end host provided location information there is the possibility that that location is incorrect, either intentionally (in case of an a denial of service attack against the emergency services infrastructure) or due to a malfunctioning device. The reader is referred to [RFC7378] for a discussion of some of these vulnerabilities.

12. Privacy Considerations

Since this document builds on [I-D.ietf-ecrit-ecall], which itself builds on [I-D.ietf-ecrit-additional-data], the data structures specified there, and the corresponding privacy considerations discussed there, apply here as well. The VEDS data structure contains optional elements that can carry identifying and personal information, both about the vehicle and about the owner, as well as location information, and so needs to be protected against unauthorized disclosure. Local regulations may impose additional privacy protection requirements.

13. IANA Considerations

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

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

MIME media type name: application

MIME subtype name: EmergencyCallData.VEDS+xml

Mandatory parameters: none

Optional parameters: charset

Indicates the character encoding of enclosed XML.

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

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

Interoperability considerations: None

Published specification: [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>

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

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

14. Contributors

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

15. Acknowledgements

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

16. Changes from Previous Versions

16.1. Changes from draft-ietf-03 to draft-ietf-04

- o Added example VEDS object
- o Additional clarifications and corrections
- o Removed references from Abstract
- o Moved Document Scope section to follow Introduction

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

- o Additional clarifications and corrections

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

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

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

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

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

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

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

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

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

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

17. References

17.1. Normative References

- [I-D.ietf-ecrit-additional-data]
Gellens, R., Rosen, B., Tschofenig, H., Marshall, R., and J. Winterbottom, "Additional Data Related to an Emergency Call", draft-ietf-ecrit-additional-data-37 (work in progress), October 2015.
- [I-D.ietf-ecrit-ecall]
Gellens, R. and H. Tschofenig, "Next-Generation Pan-European eCall", draft-ietf-ecrit-ecall (work in progress), March 2015.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC3023] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", RFC 3023, DOI 10.17487/RFC3023, January 2001, <<http://www.rfc-editor.org/info/rfc3023>>.
- [RFC4119] Peterson, J., "A Presence-based GEOPRIV Location Object Format", RFC 4119, DOI 10.17487/RFC4119, December 2005, <<http://www.rfc-editor.org/info/rfc4119>>.
- [RFC4288] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", RFC 4288, DOI 10.17487/RFC4288, December 2005, <<http://www.rfc-editor.org/info/rfc4288>>.

- [RFC5031] Schulzrinne, H., "A Uniform Resource Name (URN) for Emergency and Other Well-Known Services", RFC 5031, DOI 10.17487/RFC5031, January 2008, <<http://www.rfc-editor.org/info/rfc5031>>.
- [RFC5491] Winterbottom, J., Thomson, M., and H. Tschofenig, "GEOPRIV Presence Information Data Format Location Object (PIDF-LO) Usage Clarification, Considerations, and Recommendations", RFC 5491, DOI 10.17487/RFC5491, March 2009, <<http://www.rfc-editor.org/info/rfc5491>>.
- [RFC5962] Schulzrinne, H., Singh, V., Tschofenig, H., and M. Thomson, "Dynamic Extensions to the Presence Information Data Format Location Object (PIDF-LO)", RFC 5962, DOI 10.17487/RFC5962, September 2010, <<http://www.rfc-editor.org/info/rfc5962>>.
- [RFC6443] Rosen, B., Schulzrinne, H., Polk, J., and A. Newton, "Framework for Emergency Calling Using Internet Multimedia", RFC 6443, DOI 10.17487/RFC6443, December 2011, <<http://www.rfc-editor.org/info/rfc6443>>.
- [RFC6881] Rosen, B. and J. Polk, "Best Current Practice for Communications Services in Support of Emergency Calling", BCP 181, RFC 6881, DOI 10.17487/RFC6881, March 2013, <<http://www.rfc-editor.org/info/rfc6881>>.
- [VEDS] "Vehicular Emergency Data Set (VEDS) version 3", July 2012, <<https://www.apointnl.org/resources/telematics/aacn-and-veds.html>>.

17.2. Informative references

- [RFC5012] Schulzrinne, H. and R. Marshall, Ed., "Requirements for Emergency Context Resolution with Internet Technologies", RFC 5012, DOI 10.17487/RFC5012, January 2008, <<http://www.rfc-editor.org/info/rfc5012>>.
- [RFC5069] Taylor, T., Ed., Tschofenig, H., Schulzrinne, H., and M. Shanmugam, "Security Threats and Requirements for Emergency Call Marking and Mapping", RFC 5069, DOI 10.17487/RFC5069, January 2008, <<http://www.rfc-editor.org/info/rfc5069>>.
- [RFC7378] Tschofenig, H., Schulzrinne, H., and B. Aboba, Ed., "Trustworthy Location", RFC 7378, DOI 10.17487/RFC7378, December 2014, <<http://www.rfc-editor.org/info/rfc7378>>.

Authors' Addresses

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

Email: rg+ietf@randy.pensive.org

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

Email: br@brianrosen.net

Hannes Tschofenig
(Individual)

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

ECRIT
Internet-Draft
Intended status: Standards Track
Expires: February 12, 2016

B. Rosen
NeuStar, Inc.
H. Schulzrinne
Columbia U.
H. Tschofenig
August 11, 2015

Data-Only Emergency Calls
draft-ietf-ecrit-data-only-ea-10.txt

Abstract

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

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

Status of This Memo

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

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

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

This Internet-Draft will expire on February 12, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

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

1. Introduction

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

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

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

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

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

2. Terminology

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

3. Architectural Overview

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

1. Emergency alerts containing only data are targeted to a intermediary recipient responsible for evaluating the next steps. These steps could include:
 1. Sending a call containing only data toward a Public Safety Answering Point (PSAP);
 2. Establishing a third-party initiated emergency call towards a PSAP that could include audio, video, and data.
2. Emergency alerts may be targeted to a Service URN used for IP-based emergency calls where the recipient is not known to the originator. In this scenario, the alert may contain only data (e.g., a CAP, Geolocation header and one or more Call-Info headers containing Additional Data [I-D.ietf-ecrit-additional-data] in a SIP MESSAGE).

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

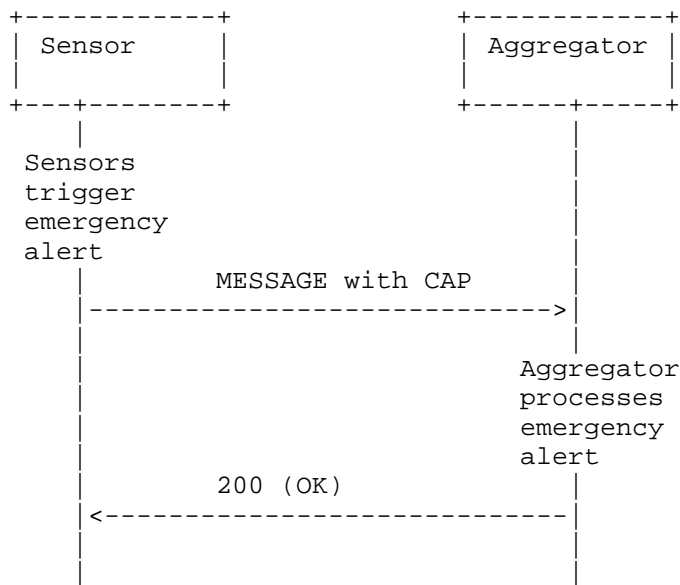


Figure 1: Targeted Emergency Alert Routing

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. In this use case, the only difference between an emergency call, and an emergency data-only call is that the former uses INVITE, creates a session and negotiates one or more media streams, while the latter uses MESSAGE, does not create a session and does not have media.

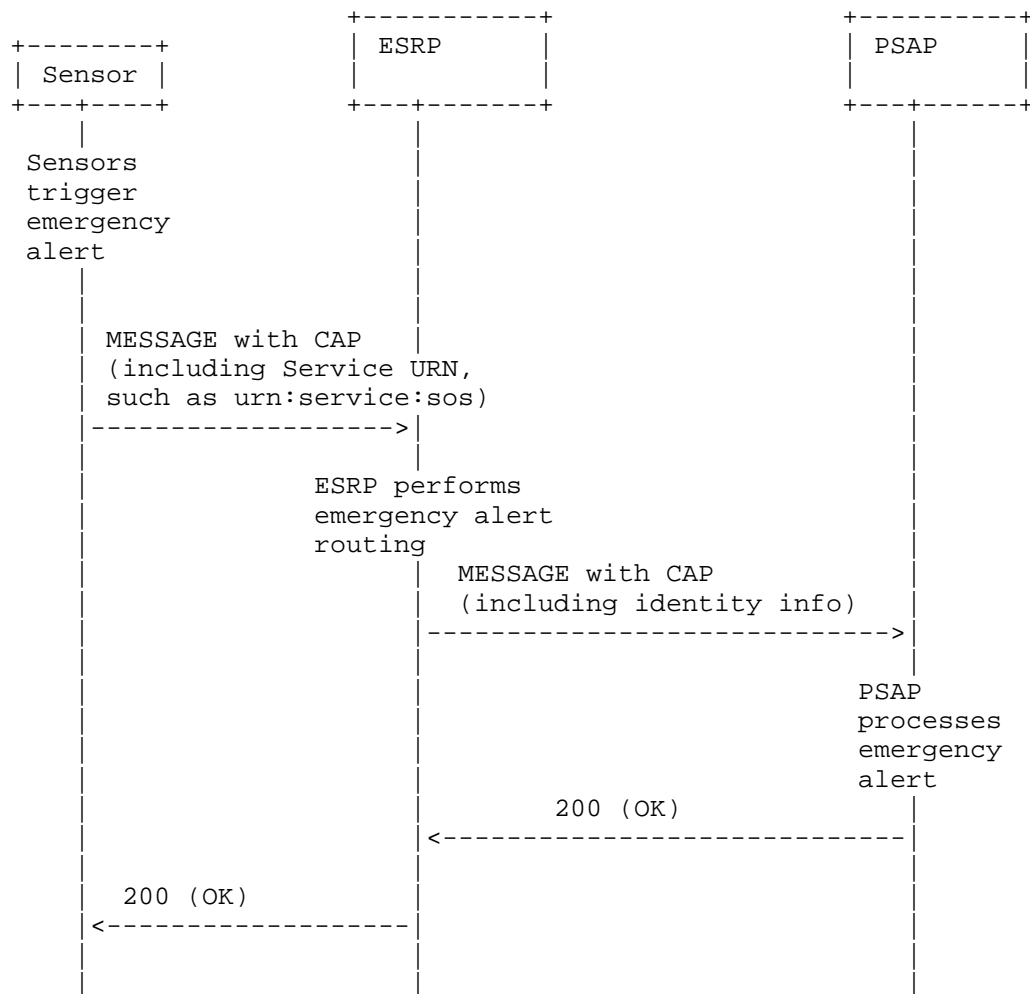


Figure 2: Location-Based Emergency Alert Routing

4. Protocol Specification

4.1. CAP Transport

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

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

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

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

4.2. Profiling of the CAP Document Content

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

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

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

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

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

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

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

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

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

4.3. Sending a Data-Only Emergency Call

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

5. Error Handling

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

5.1. 425 (Bad Alert Message) Response Code

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

425 (Bad Alert Message)

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

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

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

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

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

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

5.2. The AlertMsg-Error Header Field

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

6. Updates to the CAP Message

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

7. Call Backs

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

8. Handling Large Amounts of Data

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

9. Example

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

MESSAGE sip:aggregator@domain.com SIP/2.0

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

--boundary1

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

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

--boundary1

```

Content-Type: application/pidf+xml
Content-ID: <abcdef2@domain.com>
Content-Disposition: by-reference;handling=optional
<?xml version="1.0" encoding="UTF-8"?>
  <presence
    xmlns="urn:ietf:params:xml:ns:pidf"
    xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
    xmlns: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:agggreg.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

10. Security Considerations

This section discusses security considerations when SIP user agents issue emergency alerts utilizing MESSAGE and CAP. Location specific threats are not unique to this document and are discussed in [RFC7378] and [RFC6442].

The ECRIT emergency services architecture [RFC6443] considers classical individual-to-authority emergency calling and the identity of the emergency caller does not play a role at the time of the call establishment itself, i.e., a response to the emergency call will not depend on the identity of the caller. In case of emergency alerts generated by devices, like sensors, the processing may be different in order to reduce the number of falsely generated emergency alerts. Alerts may get triggered based on certain sensor input that may have been caused by other factors than the actual occurrence of an alert relevant event. For example, a sensor may simply be malfunctioning.

For this purpose not all alert messages are directly sent to a PSAP but rather may be pre-processed by a separate entity, potentially under supervision by a human, to filter alerts and potentially correlate received alerts with others to obtain a larger picture of the ongoing situation.

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

1. SIP itself provides security mechanisms that allow the verification of the originator's identity. These mechanisms can be re-used, such as P-Asserted-Identity [RFC3325] or SIP Identity [RFC4474]. The latter provides a cryptographic assurance while the former relies on a chain of trust model.
2. CAP provides additional security mechanisms and the ability to carry additional information about the sender's identity. Section 3.3.2.1 of [cap] specifies the signing algorithms of CAP documents.

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

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

11. IANA Considerations

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

To: ietf-types@iana.org

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

MIME media type name: application

MIME subtype name: cap+xml

Required parameters: (none)

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

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

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

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

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

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

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

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

Intended usage: Limited use

Author/Change controller: IETF ECRIT working group

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

11.2. IANA Registration of Additional Data Block

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

11.3. IANA Registration for 425 Response Code

In the SIP Response Codes registry, the following is added

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

Response code: 425 (recommended number to assign)

Default reason phrase: Bad Alert Message

Registry:

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

This SIP Response code is defined in Section 5.

11.4. IANA Registration of New AlertMsg-Error Header Field

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

1. Update the Header Fields registry with

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]

11.5. IANA Registration for the SIP AlertMsg-Error Codes

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

Registry Name: AlertMsg-Error Codes

Reference: [this doc]

Registration Procedures: Specification Required

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.

12. Acknowledgments

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

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

13. References

13.1. Normative References

- [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, DOI 10.17487/RFC2392, August 1998, <<http://www.rfc-editor.org/info/rfc2392>>.
- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <<http://www.rfc-editor.org/info/rfc3261>>.

- [RFC3428] Campbell, B., Ed., Rosenberg, J., Schulzrinne, H., Huitema, C., and D. Gurle, "Session Initiation Protocol (SIP) Extension for Instant Messaging", RFC 3428, DOI 10.17487/RFC3428, December 2002, <<http://www.rfc-editor.org/info/rfc3428>>.
- [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <<http://www.rfc-editor.org/info/rfc5234>>.
- [RFC3023] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", RFC 3023, DOI 10.17487/RFC3023, January 2001, <<http://www.rfc-editor.org/info/rfc3023>>.
- [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <<http://www.rfc-editor.org/info/rfc3629>>.
- [RFC6442] Polk, J., Rosen, B., and J. Peterson, "Location Conveyance for the Session Initiation Protocol", RFC 6442, DOI 10.17487/RFC6442, December 2011, <<http://www.rfc-editor.org/info/rfc6442>>.
- [RFC6881] Rosen, B. and J. Polk, "Best Current Practice for Communications Services in Support of Emergency Calling", BCP 181, RFC 6881, DOI 10.17487/RFC6881, March 2013, <<http://www.rfc-editor.org/info/rfc6881>>.
- [I-D.ietf-ecrit-additional-data]
Gellens, R., Rosen, B., Tschofenig, H., Marshall, R., and J. Winterbottom, "Additional Data Related to an Emergency Call", draft-ietf-ecrit-additional-data-33 (work in progress), July 2015.
- [I-D.rosen-ecrit-addldata-subnot]
Rosen, B., "Updating Additional Data related to an Emergency Call using Subscribe/ Notify", draft-rosen-ecrit-addldata-subnot-01 (work in progress), November 2013.

13.2. Informative References

- [RFC7378] Tschofenig, H., Schulzrinne, H., and B. Aboba, Ed., "Trustworthy Location", RFC 7378, DOI 10.17487/RFC7378, December 2014, <<http://www.rfc-editor.org/info/rfc7378>>.

- [RFC4474] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, DOI 10.17487/RFC4474, August 2006, <<http://www.rfc-editor.org/info/rfc4474>>.
- [RFC3325] Jennings, C., Peterson, J., and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", RFC 3325, DOI 10.17487/RFC3325, November 2002, <<http://www.rfc-editor.org/info/rfc3325>>.
- [RFC6443] Rosen, B., Schulzrinne, H., Polk, J., and A. Newton, "Framework for Emergency Calling Using Internet Multimedia", RFC 6443, DOI 10.17487/RFC6443, December 2011, <<http://www.rfc-editor.org/info/rfc6443>>.

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
Hall in Tirol 6060
Austria

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

ECRIT
Internet-Draft
Intended status: Informational
Expires: April 20, 2016

R. Gellens
Qualcomm Technologies, Inc.
H. Tschofenig
(Individual)
October 18, 2015

Next-Generation Pan-European eCall
draft-ietf-ecrit-ecall-04.txt

Abstract

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

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

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

Status of This Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 20, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Terminology	3
2. Document Scope	4
3. Introduction	4
4. eCall Requirements	6
5. Vehicle Data	7
6. Call Setup	7
7. Call Routing	9
7.1. ESInets	9
8. Test Calls	10
9. eCall-Specific Control/Metadata	10
9.1. The eCall Control Block	11
9.1.1. The <ack> element	12
9.1.1.1. Attributes of the <ack> element	13
9.1.1.2. Child Elements of the <ack> element	13
9.1.1.3. Ack Examples	14
9.1.2. The <capabilities> element	15
9.1.2.1. Child Elements of the <capabilities> element	15
9.1.2.2. Capabilities Example	16
9.1.3. The <request> element	16
9.1.3.1. Attributes of the <request> element	17
9.1.3.2. Child Elements of the <request> element	19
9.1.3.3. Request Example	19
9.2. The emergencyCallData.eCall INFO package	20
10. Examples	21
11. Security Considerations	25
12. Privacy Considerations	26

13. XML Schema	26
14. IANA Considerations	29
14.1. Service URN Registrations	29
14.2. MIME Content-type Registration for 'application/emergencyCallData.eCall.MSD+xml'	30
14.3. MIME Content-type Registration for 'application/emergencyCallData.eCall.control+xml'	31
14.4. Registration of the 'eCall.MSD' entry in the Emergency Call Additional Data Blocks registry	33
14.5. Registration of the 'eCall.control' entry in the Emergency Call Additional Data Blocks registry	33
14.6. Registration of the emergencyCallData.eCall Info Package	33
14.7. URN Sub-Namespace Registration	33
14.7.1. Registration for urn:ietf:params:xml:ns:eCall	33
14.7.2. Registration for urn:ietf:params:xml:ns:eCall:control	34
14.8. Registry creation	35
14.8.1. eCall Control Action Registry	35
14.8.2. eCall Static Message Registry	36
14.8.3. eCall Reason Registry	37
14.8.4. eCall Lamp ID Registry	37
14.8.5. eCall Camera ID Registry	38
15. Contributors	39
16. Acknowledgements	39
17. Changes from Previous Versions	39
17.1. Changes from draft-ietf-02 to draft-ietf-03	39
17.2. Changes from draft-ietf-01 to draft-ietf-02	39
17.3. Changes from draft-ietf-00 to draft-ietf-01	40
17.4. Changes from draft-gellens-03 to draft-ietf-00	40
17.5. Changes from draft-gellens-02 to -03	40
17.6. Changes from draft-gellens-01 to -02	40
17.7. Changes from draft-gellens-00 to -01	40
18. References	41
18.1. Normative References	41
18.2. Informative references	42
Authors' Addresses	43

1. Terminology

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

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

Additionally, we use the following abbreviations:

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

2. Document Scope

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

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

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

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

3. Introduction

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

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

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

An eCall can be either user-initiated or automatically triggered. Automatically triggered eCalls indicate a car crash or some other serious incident and carry a greater presumption of risk of injury. Manually triggered eCalls might 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 so might be subject to different operational handling by the PSAP.

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

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

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

4. eCall Requirements

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

eCall requires:

- o The call be recognized as an eCall (which is inherently an emergency call)
- o The call setup indicates if the call was manually or automatically triggered
- o A voice channel between the vehicle and the PSAP
- o Carrying the MSD intrinsically with the call (the MSD needs to be available to the same call-taker as the voice)
- o The ability for the PSAP to acknowledge receipt of the MSD
- o The ability for the PSAP to request that the vehicle generate and transmit a new MSD
- o The ability of the PSAP to be able to re-contact the occupants of vehicle after the initial eCall is concluded
- o The ability to perform a test call (which can 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:

- o The ability to carry more data (e.g., an enhanced MSD or an MSD plus additional sets of data)
- o The ability to handle video
- o The ability to handle text
- o 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)
- o The ability for the PSAP to request the vehicle to take actions (e.g., sound the horn, disable the ignition, lock/unlock doors)
- o The ability to avoid audio muting of the voice channel (because the MSD is not transferred using an in-band modem)

5. Vehicle Data

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

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

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

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

6. Call Setup

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

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

```

///----\\      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 interprets 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 can 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 might 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 might need the ability to route an eCall itself (e.g., to an interworking facility with interconnection to a suitable legacy eCall capable PSAP) based on the eCall and manual or automatic indications. The ETSI TR "Mobile Standards Group (MSG); eCall for VoIP" [MSG_TR] discusses transition issues in Clause 7.

8. Test Calls

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

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

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

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

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

9. eCall-Specific Control/Metadata

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

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

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

This mechanism requires

- o An XML definition of the eCall control object
- o An extension mechanism by which new elements can be added to the control object definition (e.g., permitting additional elements to be included by adding their namespace)
- o A MIME type registration for the control object (so it can be carried in SIP messages and responses)
- o 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
- o An Info-Package registration per [RFC6086] permitting the control block within Info messages

9.1. The eCall Control Block

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

- ack Used in a control block sent by either side. The PSAP uses this to acknowledge receipt of data set sent by the IVS. The IVS uses this to acknowledge receipt of a request by the PSAP when that request would not otherwise be acknowledged (if the PSAP requests the vehicle to send data and the vehicle does so, the data serves as a success acknowledgement).
- capabilities: Used in a control block sent from the IVS to the PSAP (e.g., in the initial INVITE) to inform the PSAP of the vehicle capabilities. Child elements contain all actions and data types supported by the vehicle and all available lamps (lights) and cameras.

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

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

- o Transmit data object

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

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

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

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

The <request> element contains attributes to indicate the request and to supply any needed information, and MAY contain a <text> child element to contain the text for a dynamic message. The 'action' attribute is mandatory and indicates the specific action. An IANA registry is created in Section 14.8.1 to contain the allowed values.

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

There is no 'request' action to play dynamic media (such as a pre-recorded audio message). The SIP re-INVITE mechanism can be used to establish a one-way media stream for this purpose.

9.1.1. The <ack> element

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

An <ack> element sent by an IVS references the unique ID of the request being acknowledged, indicates whether the request was successfully performed, and if not, optionally includes an explanation.

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

9.1.1.1. Attributes of the <ack> element

The <ack> element has the following attributes:

Name: ref

Usage: Mandatory

Type: anyURI

Description: References the Content-ID of the body part that contained the data object or control object being acknowledged.

Example: <ack received="yes" ref="1234567890@atlanta.example.com"/>

Name: received

Usage: Conditional: mandatory in an >ack< element sent by a PSAP; not applicable in an >ack< element sent by an IVS

Type: Boolean

Description: Indicates if the referenced object was successfully received or not

Example: <ack received="yes" ref="1234567890@atlanta.example.com"/>

9.1.1.2. Child Elements of the <ack> element

The <ack> element has the following child elements:

Name: actionResult

Usage: Optional

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

Name: action

Usage: Mandatory

Type: token

Description: Contains the value of the 'action' attribute of the <request> element

Name: success

Usage: Mandatory

Type: Boolean

Description: Indicates if the action was successfully accomplished

Name: reason

Usage: Conditional

Type: token

Description: Used when 'success' is "False", this attribute contains a reason code for a failure. A registry for reason codes is defined in Section 14.8.3.

Name: details

Usage: optional

Type: string

Description: Contains further explanation of the circumstances of a success or failure. The contents are implementation-specific and human-readable.

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

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

9.1.1.3. Ack Examples

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

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

</EmergencyCallData.eCallControl>
```

Figure 3: Ack Example from PSAP to IVS

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

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

</EmergencyCallData.eCallControl>
```

Figure 4: Ack Example from IVS to PSAP

9.1.2. The <capabilities> element

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

9.1.2.1. Child Elements of the <capabilities> element

The <capabilities> element has the following child elements:

Name: request

Usage: Mandatory

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

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

All other actions are OPTIONAL.

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

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

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

Examples:

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

9.1.2.2. Capabilities Example

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

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

</EmergencyCallData.eCallControl>
```

Figure 5: Capabilities Example

9.1.3. The <request> element

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

9.1.3.1. Attributes of the <request> element

The <request> element has the following attributes:

Name: action

Usage: Mandatory

Type: token

Description: Identifies the action that the vehicle is requested to perform. An IANA registry is established in Section 14.8.1 to contain the allowed values.

Example: action="send-data"

Name: msgid

Usage: Conditional

Type: int

Description: Mandatory with a "msg-static" action. Indicates the identifier of the static message to be displayed and/or spoken for the vehicle occupants. This document established an IANA registry for messages and their IDs, in Section 14.8.2

Example: msgid="3"

Name: persistence

Usage: Optional

Type: duration

Description: Specifies how long to carry on the specified action, for example, how long to continue honking or flashing. If absent, the default is indefinitely.

Example: persistence="PT1H"

Name: datatype

Usage: Conditional

Type: token

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

Example: datatype="eCall.MSD"

Name: supported-datatypes

Usage: Conditional

Type: string

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

in the 'Emergency Call Data Types' IANA registry established in [additional-data-draft]. Multiple values are separated with a semicolon.

Example: supported-datatypes="eCall.MSD; VEDS; eCall.foo"

Name: lamp-action

Usage: Conditional

Type: token

Description: Used with a 'lamp' action, indicates if the lamp is to be illuminated, turned off, or flashed. Permitted values are 'on', 'off', and 'flash'.

Example: lamp-action="flash"

Name: lamp-ID

Usage: Conditional

Type: token

Description: Used with a 'lamp' action, indicates which lamp the action affects. Permitted values are contained in the registry of lamp-ID tokens created in Section 14.8.4

Example: lamp-ID="hazard"

Name: supported-lamps

Usage: Conditional

Type: string

Description: Used with a 'lamp' action in a <request> element that is a child of a <capability> element, this attribute lists all supported lamps, using values in the registry of lamp-ID tokens created in Section 14.8.4. Multiple values are separated with a semicolon.

Example: supported-lamps="head; interior; fog-front; fog-rear; brake; position-front; position-rear; turn-left; turn-right; hazard"

Name: camera-ID

Usage: Conditional

Type: token

Description: Used with an 'enable-camera' action, indicates which camera to enable. Permitted values are contained in the registry of camera-ID tokens created in Section 14.8.5. When a vehicle camera is enabled, the IVS sends a re-INVITE to negotiate a one-way media stream for the camera.

Example: camera-ID="backup"

Name: supported-cameras

Usage: Conditional

Type: string

Description: Used with an 'enable-camera' action in a <request> element that is a child of a <capability> element, this attribute

lists all cameras that the vehicle supports (can add as a video feed in the current session), using the same identifiers as are used in the 'camera-ID' attribute (contained in the camera ID registry in Section 14.8.5). Multiple values are separated with a semicolon.

Example: supported-cameras="backup; interior"

9.1.3.2. Child Elements of the <request> element

The <request> element has the following child elements:

Name: text

Usage: Conditional

Type: string

Description: Used within a <request action="msg-dynamic"> element to contain the text to be displayed and/or spoken (via text-to-speech) for the vehicle occupants.

Example: <text>Emergency authorities are aware of your incident and location. Due to a multi-vehicle incident in your area, no one is able to speak with you right now. Please remain calm. We will assist you soon.</text>

9.1.3.3. Request Example

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

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

</EmergencyCallData.eCallControl>
```

Figure 6: Request Example

9.2. The emergencyCallData.eCall INFO package

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

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

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

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

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

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

10. Examples

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

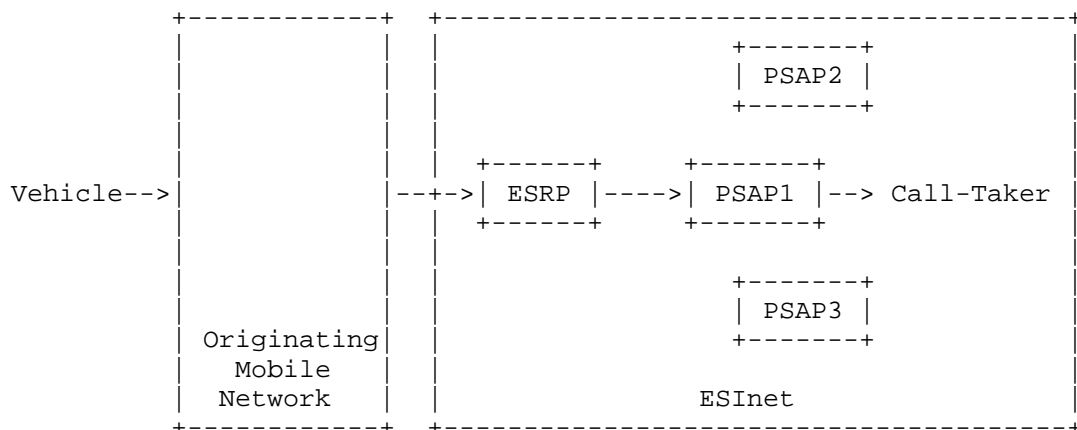


Figure 7: Example of NG-eCall Message Flow

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

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

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

--boundary1

Content-Type: application/sdp

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

--boundary1

Content-Type: application/emergencyCallData.eCall.MSD+xml

Content-ID: 1234567890@atlanta.example.com

Content-Disposition: by-reference;handling=optional

<ECallMessage>

 <id>1</id>

 <msd>

 <msdStructure>

 <messageIdentifier>1</messageIdentifier>

 <control>

 <automaticActivation> <true/> </automaticActivation>

 <testCall> <false/> </testCall>

 <positionCanBeTrusted> <true/> </positionCanBeTrusted>

 <vehicleType> <passengerVehicleClassM1/> </vehicleType>

 </control>

 <vehicleIdentificationNumber>

 <isowmi>WMI</isowmi>

 <isovds>VDSVDS</isovds>

 <isovisModelyear>Y</isovisModelyear>

 <isovisSeqPlant>A123456</isovisSeqPlant>

 </vehicleIdentificationNumber>

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

<timestamp>123456789</timestamp>

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

<vehicleDirection>14</vehicleDirection>

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

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

<numberOfPassengers>2</numberOfPassengers>

</msdStructure>

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

--boundary1
Content-Type: application/emergencyCallData.eCall.control+xml
Content-ID: 2345678901@atlanta.example.com
Content-Disposition: by-reference;handling=optional

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

<capabilities>
```

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

</EmergencyCallData.eCallControl>

--boundary1--
```

Figure 8: SIP NG-eCall INVITE

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


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

--boundaryX
Content-Type: application/sdp

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

--boundaryX
Content-Type: application/EmergencyCallData.eCall-control+xml
Content-ID: 2345678901@atlanta.example.com
Content-Disposition: by-reference;handling=optional

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

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

</EmergencyCallData.eCallControl>

--boundaryX--
```

Figure 9: 200 OK response to INVITE

11. Security Considerations

The security considerations described in [RFC5069] apply here.

In addition to any network-provided location that is inherently permitted for IMS emergency calls (which might be determined solely by the network, or in cooperation with or possibly entirely by the originating device), an eCall carries an IVS-supplied location within

the MSD. This is likely to be useful to the PSAP, especially when the two locations are independently determined. Even in situations where the network-supplied location is limited to the cell site, this can be useful as a sanity check on the device-supplied location contained in the MSD.

The document [RFC7378] discusses trust issues regarding location provided by or determined in cooperation with end devices.

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

12. Privacy Considerations

Since this document builds on [additional-data-draft], the data structures specified there, and the corresponding privacy considerations discussed there, apply here as well. The MSD carries some additional identifying and personal information (mostly about the vehicle and less about the owner), as well as location information, and so needs to be protected against unauthorized disclosure. Local regulations may impose additional privacy protection requirements. The additional functionality enabled by this document, such as access to vehicle camera streams, carries a burden of protection and so implementations need to be careful that access is only provided within the context of an emergency call and to an emergency services provider, for example, within a vehicle-initiated emergency call or by verifying that the request for camera access is signed by a certificate issued by an emergency services registrar.

13. XML Schema

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

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

```
<xs:schema
  targetNamespace="urn:ietf:params:xml:ns:EmergencyCallData:eCall-control"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:pi="urn:ietf:params:xml:ns:EmergencyCallData:eCall-control"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
    schemaLocation="http://www.w3.org/2009/01/xml.xsd"/>

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

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

  <xs:complexType name="ackType">
    <xs:complexContent>
      <xs:restriction base="xs:anyType">
        <xs:sequence minOccurs="1" maxOccurs="unbounded">
          <xs:element name="actionResult" minOccurs="0">
            <xs:complexType>
              <xs:attribute name="action"
                type="xs:token"
                use="required"/>
              <xs:attribute name="success"
                type="xs:boolean"
                use="required"/>
              <xs:attribute name="reason"
                type="xs:token">
                <xs:annotation>
```

```

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

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

<xs:complexType name="requestType">
    <xs:complexContent>
        <xs:restriction base="xs:anyType">
            <xs:choice minOccurs="1" maxOccurs="unbounded">
```

```

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

</xs:schema>

```

Figure 10: eCall Control Block Schema

14. IANA Considerations

14.1. Service URN Registrations

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

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

urn:service:sos.ecall.manual

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

urn:service:sos.ecall.automatic

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

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

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

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

MIME media type name: application

MIME subtype name: emergencyCallData.eCall.MSD+xml

Mandatory parameters: none

Optional parameters: charset

Indicates the character encoding of the XML content.

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

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

Interoperability considerations: None

Published specification: Annex C of EN 15722 [msd]

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

Additional information: None

Magic Number: None

File Extension: .xml

Macintosh file type code: 'TEXT'

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

Intended usage: LIMITED USE

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

Change controller: The European Committee For Standardization (CEN)

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

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

MIME media type name: application

MIME subtype name: emergencyCallData.eCall.control+xml

Mandatory parameters: none

Optional parameters: charset

Indicates the character encoding of the XML content.

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

Security considerations: This content type carries metadata and control information and requests, primarily from a Public Safety Answering Point (PSAP) to an In-Vehicle System (IVS) during an emergency call, and also capabilities from the IVS to the PSAP. Metadata (such as an acknowledgment that data sent by the IVS to the PSAP was successfully received) has limited privacy and security implications. Control information (such as requests from the PSAP that the vehicle perform an action) has some privacy and important security implications. The privacy concern arises from the ability to request the vehicle to transmit a data set, which as described in Section 14.2, can contain personal information. The security implication is the ability to request the vehicle to perform an action. It is important that control information originate only from a PSAP or other emergency services provider, and not from an impostor. The first safeguard for this is the security of the cellular network over which the emergency call was placed. In particular, when the IVS initiates an eCall over a cellular network, the MNO routes the call to a PSAP. (Calls placed using other means, such as Wi-Fi or over-the-top services, do not carry the same degree of trust.) Calls received by the IVS, such as a call-back from a PSAP, also do not carry the same degree of trust, since the current mechanisms are not ideal for verifying that such a call is indeed from a PSAP in response to an emergency call placed by the IVS. See the discussion in Section 11 and the PSAP Callback document [RFC7090]. A further safeguard, and one applicable regardless of which end initiated the call and the means of the call, is for the PSAP or emergency service provider to sign the body part using a certificate issued by a known emergency services certificate authority and for which the IVS can verify the root certificate. Sections 7 and Section 8 of [I-D.ietf-ecrit-additional-data] contain more discussion.

Interoperability considerations: None

Published specification: Annex C of EN 15722 [msd]

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

Additional information: None

Magic Number: None

File Extension: .xml

Macintosh file type code: 'TEXT'

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

Intended usage: LIMITED USE

Author: The IETF ECRIT WG.

Change controller: The IETF ECRIT WG.

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

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

14.5. Registration of the 'eCall.control' entry in the Emergency Call Additional Data Blocks registry

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

14.6. Registration of the emergencyCallData.eCall Info Package

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

14.7. URN Sub-Namespace Registration

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

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

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

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

XML:

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

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

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

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

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

XML:

```
BEGIN
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
    "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="content-type"
        content="text/html; charset=iso-8859-1"/>
    <title>Namespace for eCall Data:
        Control Block</title>
</head>
<body>
    <h1>Namespace for eCall Data</h1>
    <h2>Control Block</h2>
    <p>See [TBD: This document].</p>
</body>
</html>
END
```

14.8. Registry creation

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

14.8.1. eCall Control Action Registry

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

The content of this registry includes:

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

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

The initial set of values is listed in Table 2.

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

Table 2: eCall Control Action Registry Initial Values

14.8.2. eCall Static Message Registry

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

The content of this registry includes:

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

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

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

The initial set of values is listed in Table 3.

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

Table 3: eCall Static Message Registry

14.8.3. eCall Reason Registry

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

The content of this registry includes:

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

Description: A description of the reason.

The initial set of values is listed in Table 4.

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

Table 4: eCall Reason Registry

14.8.4. eCall Lamp ID Registry

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

clearly understandable and is sufficiently distinguishable from other lamp names.

The content of this registry includes:

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

Description: A description of the lamp (light).

The initial set of values is listed in Table 5.

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

Table 5: eCall Lamp ID Registry Initial Values

14.8.5. eCall Camera ID Registry

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

The content of this registry includes:

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

Description: A description of the camera.

The initial set of values is listed in Table 6.

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

Table 6: eCall Camera ID Registry Initial Values

15. Contributors

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

16. Acknowledgements

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

17. Changes from Previous Versions

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

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

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

- o Added clarifying text reinforcing that the data exchange is for small blocks of data infrequently transmitted
- o Clarified that dynamic media is conveyed using SIP re-INVITE to establish a one-way media stream
- o Clarified that the scope is the needs of eCall within the SIP emergency call environment

- o Added informative statement that the document may be suitable for reuse by other ACN systems
- o Clarified that normative language for the control block applies to both IVS and PSAP
- o Removed 'ref', 'supported-mime', and <media> elements
- o Minor wording improvements and clarifications

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

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

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

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

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

- o Clarifications and editorial improvements.

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

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

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

- o Now using 'EmergencyCallData' for purpose parameter values and MIME subtypes, in accordance with changes to [additional-data-draft]
- o Added reference to RFC 6443

- o Fixed bug that caused Figure captions to not appear

18. References

18.1. Normative References

[EN_16072]

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

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

Gellens, R., Rosen, B., Tschofenig, H., Marshall, R., and J. Winterbottom, "Additional Data Related to an Emergency Call", draft-ietf-ecrit-additional-data-37 (work in progress), October 2015.

[RFC2119]

Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.

[RFC3688]

Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<http://www.rfc-editor.org/info/rfc3688>>.

[RFC5031]

Schulzrinne, H., "A Uniform Resource Name (URN) for Emergency and Other Well-Known Services", RFC 5031, DOI 10.17487/RFC5031, January 2008, <<http://www.rfc-editor.org/info/rfc5031>>.

[RFC5226]

Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <<http://www.rfc-editor.org/info/rfc5226>>.

[RFC6443]

Rosen, B., Schulzrinne, H., Polk, J., and A. Newton, "Framework for Emergency Calling Using Internet Multimedia", RFC 6443, DOI 10.17487/RFC6443, December 2011, <<http://www.rfc-editor.org/info/rfc6443>>.

[RFC6838]

Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <<http://www.rfc-editor.org/info/rfc6838>>.

- [RFC6881] Rosen, B. and J. Polk, "Best Current Practice for Communications Services in Support of Emergency Calling", BCP 181, RFC 6881, DOI 10.17487/RFC6881, March 2013, <<http://www.rfc-editor.org/info/rfc6881>>.
- [RFC7303] Thompson, H. and C. Lilley, "XML Media Types", RFC 7303, DOI 10.17487/RFC7303, July 2014, <<http://www.rfc-editor.org/info/rfc7303>>.
- [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.

18.2. Informative references

- [CEN] "European Committee for Standardization", <<http://www.cen.eu>>.
- [MSG_TR] ETSI, , "ETSI Mobile Standards Group (MSG); eCall for VoIP", ETSI Technical Report TR 103 140 V1.1.1 (2014-04), April 2014.
- [RFC5012] Schulzrinne, H. and R. Marshall, Ed., "Requirements for Emergency Context Resolution with Internet Technologies", RFC 5012, DOI 10.17487/RFC5012, January 2008, <<http://www.rfc-editor.org/info/rfc5012>>.
- [RFC5069] Taylor, T., Ed., Tschofenig, H., Schulzrinne, H., and M. Shanmugam, "Security Threats and Requirements for Emergency Call Marking and Mapping", RFC 5069, DOI 10.17487/RFC5069, January 2008, <<http://www.rfc-editor.org/info/rfc5069>>.
- [RFC6086] Holmberg, C., Burger, E., and H. Kaplan, "Session Initiation Protocol (SIP) INFO Method and Package Framework", RFC 6086, DOI 10.17487/RFC6086, January 2011, <<http://www.rfc-editor.org/info/rfc6086>>.

- [RFC7090] Schulzrinne, H., Tschofenig, H., Holmberg, C., and M. Patel, "Public Safety Answering Point (PSAP) Callback", RFC 7090, DOI 10.17487/RFC7090, April 2014, <<http://www.rfc-editor.org/info/rfc7090>>.
- [RFC7378] Tschofenig, H., Schulzrinne, H., and B. Aboba, Ed., "Trustworthy Location", RFC 7378, DOI 10.17487/RFC7378, December 2014, <<http://www.rfc-editor.org/info/rfc7378>>.
- [SDO-3GPP] "3d Generation Partnership Project", <<http://www.3gpp.org/>>.
- [SDO-ETSI] "European Telecommunications Standards Institute (ETSI)", <<http://www.etsi.org>>.
- [draft-ietf-ecrit-car-crash] Gellens, R., Rosen, B., and H. Tschofenig, "Next-Generation Vehicle-Initiated Emergency Calls", draft-ietf-ecrit-car-crash (work in progress), March 2015.

Authors' Addresses

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

Email: rg+ietf@randy.pensive.org

Hannes Tschofenig
(Individual)

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

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

R. Marshall
J. Martin
TCS
B. Rosen
Neustar
October 19, 2015

A LoST extension to return complete and similar location info
draft-ietf-ecrit-similar-location-01

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on April 21, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Terminology	3
3. Overview of Returned Location Information	4
4. Returned Location Information	7
5. Complete Location returned for Valid Location response	7
6. Similar Location returned for Invalid Location response	9
7. Relax NG schema	11
8. Security Considerations	13
9. IANA Considerations	14
9.1. Relax NG Schema Registration	14
9.2. LoST Namespace Registration	14
10. References	15
10.1. Normative References	15
10.2. Informative References	15
Authors' Addresses	16

1. Introduction

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

Whether the input civic location is valid and missing information, or invalid due to missing or wrong information during input, this

document provides a mechanism to return a complete set of civic address elements for those valid or invalid cases.

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

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

2. Terminology

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

The following terms are defined in this document:

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

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

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

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

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

Civic Address Element: The term Civic Address Element is used within this document to apply to an individual CAType data descriptor, for example, as is described in [RFC4776], [RFC5774], and [RFC6848]

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

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

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

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

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

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

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

3. Overview of Returned Location Information

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

When a LoST server is asked to validate a civic location, its goal is to take the set of civic address elements provided as the location information in the LoST request, and find a unique location in its

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

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

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

Clients MAY ignore the location information this extension defines in the response. The information is optional to send, and optional to use. In the case where the location information in the request was valid, this extension does not change the validity. In the case where the location information in the request is invalid, but alternate location information is returned, the original location remains invalid, and the LoST server does not change the mapping response other than optionally including the information defined by this extension.

In a valid location response, a LoST server returns a response to a findService request that contains a set of civic address elements

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

An example showing complete location information supplied:

input address: 6000 15th Ave NW Seattle

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

By contrast, when invalid location is received from the LoST server, with this extension, the same mechanism works as follows: if a LoST server returns a response to a `findService` request that contains a set of civic address elements with one or more labeled as invalid, the location information in the `findServiceResponse` is extended to include additional location information that it suspects might be the location desired.

In the example cited above, policy at the LoST server might deem a missing A3 element as invalid, even if the location information in the request was sufficient to identify a unique address. In that case, the missing element would be listed in the invalid list, and `similarLocation` could be returned in the response showing the missing elements including A3, the same as the above example.

As another example of the use of `similarLocation`, consider the results based on a similar data set as used above, where the HNO, RD, STS, A1, and A3 civic address elements are not sufficient to locate a unique address, which leads to an invalid location result. This is the case, despite the fact that the LoST server typically contains additional civic address elements which could have resulted in a uniquely identifiable location if additional data had been supplied with the query. Since [RFC5222] currently does not have a way for this additional location information to be returned in the

findServiceResponse, this document extends [RFC5222] so that the LoST findServiceResponse message can include one or more similarLocation elements within the findServiceResponse message representing similar civic locations.

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

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

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

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

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

This extension would allow the LoST server 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 implementing this extension MAY include completeLocation or similarLocation in the findService response. completeLocation and similarLocation contain a list of civic address elements identical to the elements used in the location element with the "civic profile".

5. Complete Location returned for Valid Location response

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

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

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

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

      <A1>WA</A1>
      <A3>Seattle</A3>
      <RD>15th</RD>
      <STS>Ave</STS>
      <POD>NW</POD>
      <HNO>6000</HNO>

    </civicAddress>
  </location>

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

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

<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:RD 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 Location response

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

```

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

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

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

```

```
<country>US</country>
<A1>WA</A1>
<A3>Seattle</A3>
<RD>15th</RD>
<STS>Ave</STS>
<POD>N</POD>
<HNO>6000</HNO>

</civicAddress>
</location>

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

</findService>

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

<findServiceResponse>
  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 ca:STS ca:RD</valid>
    <invalid>ca:POD</invalid>
    <unchecked>ca:HNO</unchecked>

    <rli:similarLocation> <!-- similar location info -->
      <ca:civicAddress> <!-- similar address #1 -->
        <ca:country>US</ca:country>
        <ca:A1>WA</ca:A1>
        <ca:A3>SEATTLE</ca:A3>
        <ca:RD>15TH</ca:RD>
```

```

        <ca:STS>AVE</ca:STS>
        <ca: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 [to be supplied in a later version of this draft].

```

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

```

```

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

similar features, such as "did you mean", and address completion, so this capability is not introducing any fundamentally new threat.

9. IANA Considerations

9.1. Relax NG Schema Registration

URI: urn:ietf:params:xml:schema:lost-rl1

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-rl1

and its last line is

}

9.2. LoST Namespace Registration

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

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

XML:

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

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC5222] Hardie, T., Newton, A., Schulzrinne, H., and H. Tschofenig, "LoST: A Location-to-Service Translation Protocol", RFC 5222, DOI 10.17487/RFC5222, August 2008, <<http://www.rfc-editor.org/info/rfc5222>>.

10.2. Informative References

- [RFC4776] Schulzrinne, H., "Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information", RFC 4776, DOI 10.17487/RFC4776, November 2006, <<http://www.rfc-editor.org/info/rfc4776>>.

- [RFC5139] Thomson, M. and J. Winterbottom, "Revised Civic Location Format for Presence Information Data Format Location Object (PIDF-LO)", RFC 5139, DOI 10.17487/RFC5139, February 2008, <<http://www.rfc-editor.org/info/rfc5139>>.
- [RFC5774] Wolf, K. and A. Mayrhofer, "Considerations for Civic Addresses in the Presence Information Data Format Location Object (PIDF-LO): Guidelines and IANA Registry Definition", BCP 154, RFC 5774, DOI 10.17487/RFC5774, March 2010, <<http://www.rfc-editor.org/info/rfc5774>>.
- [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, DOI 10.17487/RFC6848, January 2013, <<http://www.rfc-editor.org/info/rfc6848>>.

Authors' Addresses

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

Email: rmarshall@telecomsys.com
URI: <http://www.telecomsys.com>

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

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: April 21, 2016

B. Rosen
Neustar
October 19, 2015

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

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on April 21, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Conventions used in this document	3
3. <plannedChange> element	4
4. <locationInvalidated> object	4
5. uri Not Stored Warning	4
6. TTL in Response	5
7. Relax NG Schema	5
8. Security Considerations	8
9. IANA Considerations	8
9.1. Relax NG Schema Registration	8
9.2. LoST Namespace Registration	9
10. Normative References	9
Author's Address	9

1. Introduction

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

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

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

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

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

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

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

2. Conventions used in this document

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

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

3. <plannedChange> element

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

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

4. <locationInvalidated> object

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

5. uri Not Stored Warning

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

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

6. TTL in Response

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

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

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

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

7. Relax NG Schema

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

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

```
##
##      Extension to Location-to-Service Translation (LoST) Protocol
##      to support a planned change to location data
```



```
##
##      plannedChange is used in the extensionPoint of
##      commonRequestPattern in a findService request
##
##      locationInvalidated is used by the LoST server to notify a
##      LIS that a previously valid location may be (or will become)
##      invalid
##
##      ttl is used in the extensionPoint of
##      commonResponsePattern in a findService response
##
##      uriNotStored is a new warning to be used in a
##      exceptionContainer in the warnings element of a
##      findServiceResponse
##
start =
  plannedChange
  | locationInvalidated
  | uriNotStored
##
##      plannedChange
##
div {
  plannedChange =
    element plannedChange {
      attribute uri {
        xsd:anyURI }?,
      attribute asOf {
        xsd:dateTime }?,
      extensionPoint+
    }
}

##
##      locationInvalidated
##
div {
  locationInvalidated =
    element locationInvalidated {
      attribute asOf {
        xsd:dateTime }?,
      extensionPoint+
    }
}

##
##      ttl
##
```

```
div {
  ttl =
    element ttl {
      expires,
      extensionPoint+
    }
}

##
##      uriNotStored
##
div {
  uriNotStored =
    element uriNotStored { basicException }
}

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

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

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

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

8. Security Considerations

As an extension to LoST, this document inherits the security issues raised in [RFC5222]. The server could be tricked into storing a malicious URI which, when sent the locationInvalidated object could trigger something untoward. The server **MUST NOT** accept any data from the client in response to POSTing the locationInvalidated.

The server is subject to abuse by clients because it is being asked to store something and may need to send data to an uncontrolled URI. Clients could request many URIs for the same location for example. The server **MUST** have policy that limits use of this mechanism by a given client. If the policy is exceeded, the server returns the uriNotStored warning. The server **MUST** validate that the content of the uri sent is syntactically valid and meets the 256 byte limit. When sending the locationInvalidated object to the uri stored, the server **MUST** protect itself against common http vulnerabilities.

The mutual authentication between client and server when is **RECOMMENDED** for both the initial findService operation that requests storing the uri and the sending of the locationInvalidated object. The server should be well known to the client, and its credential should be learned in a reliable way. For example, a public safety system operating the LoST server may have a credential traceable to a well known Certificate Authority known to provide credentials for public safety agencies. Many of the clients will be operated by local ISPs or other service providers where the server operator can reasonably obtain a good credential to use for the URI. Where the server does not recognize the client, its policy **MAY** limit the use of this feature beyond what it would limit a client it recognized.

9. IANA Considerations

9.1. Relax NG Schema Registration

URI: urn:ietf:params:xml:schema:lost-plannedChange1

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

Relax NG Schema: The Relax NG schema to be registered is contained in Section 5. Its first line is

```
default namespace = "urn:ietf:params:xml:ns:lost-PlannedChange1  
  
and its last line is  
  
}
```

9.2. LoST Namespace Registration

URI: urn:ietf:params:xml:ns:lost-plannedChange1

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

XML:

```
BEGIN
<?xml version="2.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
  "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="content-type"
    content="text/html; charset=iso-8859-1"/>
  <title>LoST Planned Change Namespace</title>
</head>
<body>
  <h1>Namespace for LoST Planned Change extension</h1>
  <h2>urn:ietf:params:xml:ns:lost-plannedChange1</h2>
  <p>See <a href="http://www.rfc-editor.org/rfc/rfc?????.txt">
    RFC?????</a>.</p>
</body>
</html>
END
```

10. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC5222] Hardie, T., Newton, A., Schulzrinne, H., and H. Tschofenig, "LoST: A Location-to-Service Translation Protocol", RFC 5222, DOI 10.17487/RFC5222, August 2008, <<http://www.rfc-editor.org/info/rfc5222>>.

Author's Address

Brian Rosen
Neustar
470 Conrad Dr
Mars, PA 16046
US

EMail: br@brianrosen.net