Dispatch Working Group
Internet-Draft
Intended status: Informational
Expires: August 21, 2015

A. Allen, Ed.
Blackberry

February 17, 2015

Indicating that out of dialog requests related to an existing dialog must be routed via an intermediary
draft-allen-dispatch-routing-out-of-dialog-request-01

Abstract

This document discusses the problems for out of dialog requests that are related to another dialog, caused by B2BUA intermediaries that modify SIP parameters or terminate dialogs and proposes some possible solutions.

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 August 21, 2015.

Table of Contents

1. Introduction

In RFC 3265 [1] and RFC 3515 [4] SUBCRIBE requests and REFER requests were allowed to reuse a dialog created by another SIP method (e.g. INVITE). RFC 6665 [3] has deprecated such dialog reuse due to all the problems that dialog reuse caused. However some B2BUA intermediaries change parameters in SIP requests or terminate dialogs and need to receive the SUBSCRIBE and REFER requests that relate to an existing dialog that is record routed via the B2BUA. While draft-ietf-sipcore-refer-explicit-subscription [6] defines a means for the sending of REFER requests to ensure that no subscription is created by the REFER recipient and thus it is safe to send the REFER request on a existing dialog the cases where notifications are needed still require the SUBSCRIBE and REFER request to be sent on a new dialog.

2. Problem statement

SIP sessions often involve intermediaries acting as B2BUAs that in addition to forwarding SIP requests and responses also act as UAs to perform more complex manipulations of the session. Such manipulations include modifying URIs in the Request-URI, Contact address or other header fields and even terminating the dialog for some mid session requests (for example performing a session transfer when receiving a REFER request rather than forwarding the REFER request to the remote UAS).

Typically such functionality has been achieved by sending REFER and SUBSCRIBE requests within the established dialog for the session, with the intermediary then intercepting the REFER or SUBSCRIBE request and then either modifying to conform to the expected view of the remote UAS or processing the REFER or SUBSCRIBE request rather

than forwarding it to the remote UAS. However such dialog reuse has been problematic and RFC 6665 [3] has deprecated dialog reuse (except for legacy interoperability).

However, if REFER and SUBSCRIBE requests are sent outside the related existing dialog then the requests will not be routed by the manipulating B2BUA and thus will either to fail to arrive at the remote UA due to URI manipulations or fail at the remote UA because parameters in the request (e.g Target-Dialog, Replaces, Refer-To URI, etc) do not match the values at the remote UAS. draft-kaplan-dispatch-gruu-problematic-00 [7] further describes some of the problems if a GRUU is used as the Request-URI of a related out of dialog request.

One way B2BUAs have have addressed this problem is by acting as two UAs back-to-back with the Contact URI being overwritten to be the URI of the B2BUA. However this means that GRUU of the UA is overwritten by the B2BUA and the meaning of the Contact header field parameters becomes obscure. Do the Contact header field parameters reflect the capabilities of the Contact address (i.e the B2BUA) or do they reflect the capabilities of the remote UA? If they relect the capabilities of the B2BUA then the identfication of the capabilities of the remote UAS has been lost. If they reflect the capabilities of the remote UA then they falsely identify that the B2BUA contact address has the capabilities of the remote UA. While some have advocated that a B2BUA should only indicate the capabilities that it understands and supports in the Contact, in the opinion of the author this is not desirable behavior because the feature tags may indicate many kinds of capabilities which do not require the support of the intermediary. For an intermediary only to indicate those capabilities that it understands and supports is a big barrier to UAs mutually exchanging feature capabilities. In the opinion of the author the feature capability indicator mechanism defined in RFC 6809 [2] is the appropriate means for an intermediary to indicate the capabilities that it supports and will allow. It also should be recognised that UAs may store Contact addresses especially if they are GRUUs for use later for originating sessions (e.g. stored in the address book) or for filtering incoming sessions (e.g. incoming sessions addressed to temporary GRUUs). So if the Contact address is overwritten then this information is lost or not valid as a contact outside the lifetime of the current dialog. Additionally the mechanism defined in RFC 6665 [3] depends on the UA receiving a GRUU as the remote target in order to avoid dialog reuse, so overwriting the Contact Address breaks this mechanism.

What is needed is a way for intermediaries that need to receive and manipulate or process mid session requests to indicate that mid session out of dialog requests that relate to the dialog of the

session being established, to indicate a URI to be included in the Route header of such out of dialog requests so that the request will route by the intermediary.

3. Potential solutions

3.1. New SIP header field

A new SIP header field (e.g. OOD-Record-Route) could be defined which contains the URI of the intermediary for routing out of dialog requests that relate to another dialog. The intermediary would include the new header field containing the URI that the intermediary requires related out of dialog requests to be routed to in the requests and responses at dialog establishment. The UA would then include a Route header field containing the URI received in the new header field in any related out of dialog requests it sends.

3.2. New rr-param

A new rr-param(e.g. OOD-RR) could be defined which indicates that this is the URI of the intermediary for routing out of dialog requests that relate to another dialog. The intermediary would include the new rr-param when including its URI in the Record-Route header field. The UA would then include a Route header field containing the URI with the associated new rr-param received in the Record-Route header field in any related out of dialog requests it sends.

3.3. New URI parameter

A new URI parameter (e.g. OOD-RR) could be defined which indicates that this is the URI of the intermediary for routing out of dialog requests that relate to another dialog. The intermediary would include the new URI parameter when including its URI in the Record-Route header field. The UA would then include a Route header field containing the URI with the new URI parameter received in the Record-Route header field in any related out of dialog requests it sends.

3.4. New Feature Capability Indicator

RFC 6809 [2] defines the Feature-Caps header field for intermediaries to include Feature-Capability indicators indicating the capabilities they support. A new feature-capability indicator (e.g. sip.ood-route) could be defined which contains the URI of the intermediary for routing out of dialog requests that relate to another dialog. The intermediary would include a Feature-Caps header field containing the Feature-Capability indicator with the URI that the intermediary requires related out of dialog requests to be routed to in the

requests and responses at dialog establishment. The UA would then include a Route header field containing the URI received in the Feature-Capability indicator in any related out of dialog requests it sends.

3.5. Embed Route header fields in the contact URI

The Contact URI can contain embedded header fields (see RFC 3261 [5]). The intermediary could embed a Route header field containing its own URI in the Contact URI. One advantage of this approach is that there may be some backward compatibility with this mechanism because RFC 3261 [5] compliant UAs should use the embedded Route header fields when constructing a request addressed to this Contact URI. However it is questionable as to how many implemntations actually will do this in practice. A disadvantage of this approach is if the Contact URI is secured using SMIME or a similar means for detecting man in the middle attaks on the Contact address then tampering with the URI could lead to the UAS believing that the Contact URI has been maliciously tampered with.

3.6. Option Tag

A new SIP option tag will be needed for a UA to indicate that it supports the new extension so that the the intermediary can use the new mechanism instead of other approaches that modify the contact address and force dialog reuse. SIP OPTIONS method could be used by the intermediary to determine whether the UAS supports the extension before forwarding the dialog creating request. Alternatively the intermediary might modify the dialog after discovering in a response whether the UAS supports the new extension or not.

4. Security Considerations

The capability to include a URI in a request or response which will cause a UA to route other requests via the intermediary provides the possibility to create man-in-the-middle attacks. However this is also true of existing SIP header fields like Record-Route. The same considerations apply as those to the use of Record-Route header fields.

5. IANA Considerations

This document does not currently have anything requiring action by IANA.

6. Acknowledgements

The author would like to thank Hadrial Kaplan, Paul Kyzivat, Christer Holmberg and Dale Worley for the inital list discussion on the issues raised by this draft and additional suggestions on the solutions.

7. Informative References

[1] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, June 2002.

[2] Holmberg, C., Sedlacek, I., and H. Kaplan, "Mechanism to Indicate Support of Features and Capabilities in the Session Initiation Protocol (SIP)", RFC 6809, November 2012.

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

[4] Sparks, R., "The Session Initiation Protocol (SIP) Refer Method", RFC 3515, April 2003.

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

[6] Sparks, R., , "Explicit Subscriptions for the REFER Method", Internet Draft draft-ietf-sipcore-refer-explicit-subscription-00, November 2014.

[7] Kaplan, H., , "Problems with the SIP Globally Routable User Agent URIs (GRUUs)", Internet Draft draft-kaplan-dispatch-gruu-problematic-00, October 2010.

Author’s Address

Andrew Allen (editor)
Blackberry
1200 Sawgrass Corporate Parkway
Sunrise, Florida 33323
USA

Email: aallen@blackberry.com

Independent
Internet-Draft
Intended status: Informational
Expires: January 19, 2016

H. Butler
Hobu Inc.
M. Daly
Cadcorp
A. Doyle
MIT
S. Gillies
Mapbox Inc.
T. Schaub
Planet Labs
S. Hagen

July 18, 2015

The GeoJSON Format
draft-butler-geojson-06

Abstract

GeoJSON is a geospatial data interchange format based on JavaScript Object Notation (JSON). It defines several types of JSON objects and the manner in which they are combined to represent data about geographic features, their properties, and their spatial extents. This document recommends a single coordinate reference system based on WGS 84. Other coordinate reference systems are not recommended.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 19, 2016.

Table of Contents

1. Introduction

GeoJSON is a format for encoding data about geographic features using JavaScript Object Notation (JSON) [RFC7159]. The format is concerned with features in the broadest sense; any thing with qualities that are bounded in geographical space may be a feature whether it is a physical structure or not. The concepts in GeoJSON are not new; they are derived from pre-existing open geographic information system standards (for COM, SQL, and XML) and have been streamlined to better suit web application development using JSON.

GeoJSON comprises the seven concrete geometry types defined in the OpenGIS Simple Features Implementation Specification for SQL [SFSQL]: 0-dimensional Point and MultiPoint; 1-dimensional curve LineString and MultiLineString; 2-dimensional surface Polygon and MultiPolygon; and the heterogeneous GeometryCollection. GeoJSON representations of instances of these geometry types are analogous to the well-known binary (WKB) and text (WKT) representations described in that same specification.

GeoJSON also comprises the types Feature and FeatureCollection. Feature objects in GeoJSON contain a geometry object with one of the above geometry types and additional properties. A FeatureCollection object contains an array of feature objects. This structure is analogous to that of the Web Feature Service (WFS) response to GetFeatures requests specified in [WFSv1] or to a KML Folder of Placemarks [KMLv2.2]. Some implementations of the WFS specification also provide GeoJSON formatted responses to GetFeature requests, but there is no particular service model or feature type ontology implied in the GeoJSON format specification.

Since its initial publication in 2008 [GJ2008], the GeoJSON format specification has steadily grown in popularity. It is widely used in JavaScript web mapping libraries, JSON-based document databases, and web APIs.

1.1. Requirements Language

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

1.2. Conventions Used in This Document

The ordering of the members of any JSON object defined in this document MUST be considered irrelevant, as specified by [RFC7159].

Some examples use the combination of a JavaScript single line comment (//) followed by an ellipsis (...) as placeholder notation for content deemed irrelevant by the authors. These placeholders must of course be deleted or otherwise replaced, before attempting to validate the corresponding JSON code example.

Whitespace is used in the examples inside this document to help illustrate the data structures, but is not required. Unquoted whitespace is not significant in JSON.

1.3. Specification of GeoJSON

This document updates the original GeoJSON format specification [GJ2008].

1.4. Definitions

- JavaScript Object Notation (JSON), and the terms object, name, value, array, number, true, false, and null are to be interpreted as defined in [RFC7159].

- Inside this document the term "geometry type" refers to the seven case-sensitive strings: "Point", "MultiPoint", "LineString", "MultiLineString", "Polygon", "MultiPolygon", and "GeometryCollection".

- As another shorthand notation, the term "GeoJSON types" refers to the nine case-sensitive strings "Feature", "FeatureCollection" and the geometry types listed above.

1.5. Example

A GeoJSON feature collection:

```
{
    "type": "FeatureCollection",
    "features": [{
        "type": "Feature",
        "geometry": {
            "type": "Point",
            "coordinates": [102.0, 0.5]
        },
        "properties": {
            "prop0": "value0"
        }
    }, {
        "type": "Feature",
        "geometry": {
            "type": "LineString",
            "coordinates": [
                [102.0, 0.0],
                [103.0, 1.0],
                [104.0, 0.0],
                [105.0, 1.0]
            ]
        },
        "properties": {
            "prop0": "value0",
            "prop1": 0.0
        }
    }, {
        "type": "Feature",
        "geometry": {
            "type": "Polygon",
            "coordinates": [
                [
                    [100.0, 0.0],
                    [101.0, 0.0],
                    [101.0, 1.0],
                    [100.0, 1.0],
                    [100.0, 0.0]
                ]
            ]
        },
        "properties": {
            "prop0": "value0",
            "prop1": {
                "this": "that"
            }
        }
    }]
}
```

2. GeoJSON Object

GeoJSON always consists of a single object. This object (referred to as the GeoJSON object below) represents a geometry, feature, or collection of features.

- The top level of GeoJSON text MUST be a JSON object.
- The GeoJSON object MUST have a member with the name "type". The value of the member MUST be one of the GeoJSON types.
- A GeoJSON object MAY have a "bbox" member, the value of which MUST be a bounding box array (see 4. Bounding Boxes).
- The GeoJSON object MAY have any number of other members. Implementations MUST ignore unrecognized members.

2.1. Geometry Object

A geometry object is a GeoJSON object where the "type" value is one of the geometry types. A GeoJSON geometry object of any type other than "GeometryCollection" MUST have a member with the name "coordinates". The value of the coordinates member is always an array. The structure of the elements in this array is determined by the type of geometry. GeoJSON processors MAY interpret geometry objects with empty coordinates arrays as null objects.

2.1.1. Position

A position is the fundamental geometry construct. The "coordinates" member of a geometry object is composed of either:

- one position (in the case of a Point geometry),
- an array of positions (LineString or MultiPoint geometries),
- an array of arrays of positions (Polygons, MultiLineStrings),
- or a multidimensional array of positions (MultiPolygon).

A position is represented by an array of numbers. There MUST be two or more elements. The first two elements will be longitude and latitude, or easting and northing, precisely in that order and using decimal numbers. Altitude or elevation MAY be included as an optional third element.

Additional position elements MAY be included but MUST follow the three specified above and MAY be ignored by software. Interpretation

and meaning of additional elements is beyond the scope of this specification.

Examples of positions and geometries are provided in "Appendix A. Geometry Examples".

2.1.2. Point

For type "Point", the "coordinates" member MUST be a single position.

2.1.3. MultiPoint

For type "MultiPoint", the "coordinates" member MUST be an array of positions.

2.1.4. LineString

For type "LineString", the "coordinates" member MUST be an array of two or more positions.

2.1.5. MultiLineString

For type "MultiLineString", the "coordinates" member MUST be an array of LineString coordinate arrays.

2.1.6. Polygon

To specify a constraint specific to polygons, it is useful to introduce the concept of a linear ring:

- A linear ring is a closed LineString with 4 or more positions.
- The first and last positions are equivalent (they represent equivalent points).
- A linear ring is the boundary of a surface or the boundary of a hole in a surface.
- A linear ring SHOULD follow right-hand rule with respect to the area it bounds (ie. exterior rings are counter-clockwise, holes are clockwise)

Though a linear ring is not explicitly represented as a GeoJSON geometry type, it leads to a canonical formulation of the Polygon geometry type definition as follows:

- For type "Polygon", the "coordinates" member MUST be an array of linear ring coordinate arrays.

- For Polygons with more than one of these rings, the first MUST be the exterior ring and any others MUST be interior rings. The exterior ring bounds the surface and the interiors rings (if present) bound holes within the surface.

2.1.7. MultiPolygon

For type "MultiPolygon", the "coordinates" member MUST be an array of Polygon coordinate arrays.

2.1.8. Geometry Collection

A GeoJSON object with type "GeometryCollection" is a geometry object which represents a collection of geometry objects. A geometry collection MUST have a member with the name "geometries". The value corresponding to "geometries" is an array. Each element in this array is a GeoJSON geometry object.

2.2. Feature Object

A GeoJSON object with the type "Feature" is a feature object.

- A feature object MUST have a member with the name "geometry". The value of the geometry member SHALL be either a geometry object as defined above or, in the the case that the feature is unlocated, a JSON null value.

- A feature object MUST have a member with the name "properties". The value of the properties member is an object (any JSON object or a JSON null value).

- If a feature has a commonly used identifier, that identifier SHOULD be included as a member of the feature object with the name "id" and the value of this member is either a JSON string or number.

2.3. Feature Collection Object

A GeoJSON object with the type "FeatureCollection" is a feature collection object. An object of type "FeatureCollection" MUST have a member with the name "features". The value corresponding to "features" is an array. Each element in the array is a feature object as defined above.

3. Coordinate Reference System

The default reference system for all GeoJSON coordinates SHALL be a geographic coordinate reference system, using the [WGS84] datum, and with longitude and latitude units of decimal degrees. This coordinate reference system is equivalent to the OGC's "http://www.opengis.net/def/crs/OGC/1.3/CRS84" [OGCURL]. An OPTIONAL third position element SHALL be the height in meters above the WGS 84 reference ellipsoid. For widest interoperability, GeoJSON data SHOULD use this default coordinate reference system.

Other coordinate reference systems, including ones described by CRS objects of the kind defined in [GJ2008] are NOT RECOMMENDED. GeoJSON processing software SHALL NOT be expected to have access to coordinate reference systems databases. Applications requiring CRS other than the default MUST assume all responsibility for reference system and coordinate accuracy. Furthermore, GeoJSON coordinates MUST NOT under any circumstances use latitude, longitude order. See Section 6, Interoperability Considerations, for guidance in processing GeoJSON documents that do contain such a CRS object.

4. Bounding Box

A GeoJSON object MAY have a member named "bbox" to include information on the coordinate range for its geometries, features, or feature collections. The value of the bbox member MUST be an array of length 2*n where n is the number of dimensions represented in the contained geometries, with all axes of the most south-westerly point followed by all axes of the more north-easterly point. The axes order of a bbox follows the axes order of geometries.

Example of a bbox member on a feature:

```
{
    "type": "Feature",
    "bbox": [-180.0, -90.0, 180.0, 90.0],
    "geometry": {
        "type": "Polygon",
        "coordinates": [
            [
                [-180.0, 10.0],
                [20.0, 90.0],
                [180.0, -5.0],
                [-30.0, -90.0]
            ]
        ]
    }
    //...
}
```

Example of a bbox member on a feature collection:

```
{
    "type": "FeatureCollection",
    "bbox": [100.0, 0.0, 105.0, 1.0],
    "features": [
    //...
    ]
}
```

Example of a bbox for line crossing the date-line:

```
{
    "type": "Feature",
    "bbox": [170, 10, -170, 11],
    "geometry": {
        "type": "LineString",
        "coordinates": [
            [-170, 10],
            [170, 11]
        ]
    }
    //...
}
```

5. Security Considerations

GeoJSON shares security issues common to all JSON content types. See [RFC7159] Section 12 for additional information. GeoJSON does not provide executable content.

As with other geographic data formats, e.g., [KMLv2.2], providing details about the locations of sensitive persons, animals, habitats, and facilities can expose them to unauthorized tracking or injury. GeoJSON does not provide privacy or integrity services; if sensitive data requires privacy or integrity protection the service must be provided externally.

6. Interoperability Considerations

6.1. I-JSON

GeoJSON texts SHOULD follow the constraints of I-JSON [RFC7493] for maximum interoperability.

6.2. Coordinate Precision

The size of a GeoJSON text in bytes is a major interoperability consideration and precision of coordinate values has a large impact on the size of texts. A GeoJSON text containing many detailed polygons can be inflated almost by a factor of two by increasing coordinate precision from 6 to 15 decimal places. For geographic coordinates with units of degrees, 6 decimal places (a default common in, e.g., sprintf) amounts to about 10 centimeters, a precision well within that of current GPS systems. Implementations should consider the cost to using a greater precision than necessary.

6.3. Coordinate Order

There are conflicting precedents among geographic data formats over whether latitude or longitude come first in a pair of numbers. Longitude comes first in GeoJSON coordinates as it does in [KMLv2.2].

Some commonly-used CRS definitions specify coordinate ordering that is not longitude then latitude (for a geographic CRS) or easting then northing (for a projected CRS). The CRS historically known as "EPSG:4326" and more accurately named "http://www.opengis.net/def/crs/EPSG/0/4326" is a prime example. Using such a CRS is NOT RECOMMENDED due to the potential disruption of interoperability. When such a CRS is encountered in GeoJSON, the document should be processed with caution. Heuristics may be necessary to interpret the coordinates properly; they may not be in the required longitude, latitude order.

6.4. Coordinate Reference System Identifiers

Earlier versions of the GeoJSON specification recommended use of OGC URNs such as "urn:ogc:def:crs:OGC:1.3:CRS84" to name a CRS. This version deprecates the URNs and recommends a change to HTTP URLs

[Section 3.1]. Widely deployed systems using, e.g. the GDAL and OGR libraries, currently write the deprecated OGC URNs into GeoJSON documents and will do so until replaced by newer versions. GeoJSON processors should be prepared for either form.

6.5. Bounding boxes

In representing features that cross the dateline or the poles, following the ring-orientation best practice (counter-clockwise external rings, clockwise internal rings) and ensuring your bounding boxes use the south-west corner as the first coordinate will improve interoperability. Remain aware that software that represents edges as straight cartesian lines and software that represents edges as great circles will have different interpretations of edges, which vary more the longer the edges are. Try to avoid edges of more than 180 degrees in length as far as possible.

7. IANA Considerations

The MIME media type for GeoJSON text is application/vnd.geo+json.

Type name: application

Subtype name: vnd.geo+json

Required parameters: n/a

Optional parameters: n/a

Encoding considerations: binary

Security considerations: See section 5 above

Interoperability considerations: See section 6 above

Published specification: draft-butler-geojson

Applications that use this media type: various

Additional information:

Magic number(s) : n/a

File extension(s) : .json, .geojson

Macintosh file type code : TEXT

Object Identifiers: n/a

Person to contact for further information:

Sean Gillies

sean.gillies@gmail.com

Intended usage: COMMON

Restrictions on usage: none

8. References

8.1. Normative References

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

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <http://www.rfc-editor.org/info/rfc7159>.

[RFC7493] Bray, T., Ed., "The I-JSON Message Format", RFC 7493, DOI 10.17487/RFC7493, March 2015, <http://www.rfc-editor.org/info/rfc7493>.

8.2. Informative References

[GJ2008] Butler, H., Daly, M., Doyle, A., Gillies, S., Schaub, T., and C. Schmidt, "The GeoJSON Format Specification", June 2008.

[KMLv2.2] Wilson, T., "OGC KML", OGC 07-147r2, April 2008.

[OGCURL] Cox, S., "OGC-NA Name type specification - definitions: Part 1 - basic name", OGC 09-048r3, March 2010.

[SFSQL] OpenGIS Consortium, Inc., "OpenGIS Simple Features Specification For SQL Revision 1.1", OGC 99-049, May 1999.

[WFSv1] Vretanos, P., "Web Feature Service Implementation Specification", OGC 02-058, May 2002.

[WGS84] National Imagery and Mapping Agency, "Department of Defense World Geodetic System 1984, Third Edition", 1984.

Appendix A. Geometry Examples

Each of the examples below represents a valid and complete GeoJSON object.

A.1. Points

Point coordinates are in x, y order (easting, northing for projected coordinates, longitude, latitude for geographic coordinates):

```
{
    "type": "Point",
    "coordinates": [100.0, 0.0]
}
```

A.2. LineStrings

Coordinates of LineString are an array of positions (see "2.1.1. Position"):

```
{
    "type": "LineString",
    "coordinates": [
        [100.0, 0.0],
        [101.0, 1.0]
    ]
}
```

A.3. Polygons

Coordinates of a Polygon are an array of LinearRing (cf. "2.1.6 Polygon") coordinate arrays. The first element in the array represents the exterior ring. Any subsequent elements represent interior rings (or holes).

No holes:

```
{
    "type": "Polygon",
    "coordinates": [
        [
            [100.0, 0.0],
            [101.0, 0.0],
            [101.0, 1.0],
            [100.0, 1.0],
            [100.0, 0.0]
        ]
    ]
}
```

With holes:

```
{
    "type": "Polygon",
    "coordinates": [
        [
            [100.0, 0.0],
            [101.0, 0.0],
            [101.0, 1.0],
            [100.0, 1.0],
            [100.0, 0.0]
        ],
        [
            [100.8, 0.8],
            [100.8, 0.2],
            [100.2, 0.2],
            [100.2, 0.8],
            [100.8, 0.8]
        ]
    ]
}
```

With hole crossing dateline:

```
{
    "type": "Polygon",
    "coordinates": [
        [
            [-170.0, 10.0],
            [170.0, 10.0],
            [170.0, -10.0],
            [-170.0, -10.0],
            [-170.0, 10.0]
        ],
        [
            [175.0, 5.0],
            [-175.0, 5.0],
            [-175.0, -5.0],
            [175.0, -5.0],
            [175.0, 5.0]
        ]
    ]
}
```

A.4. MultiPoints

Coordinates of a MultiPoint are an array of positions::

```
{
    "type": "MultiPoint",
    "coordinates": [
        [100.0, 0.0],
        [101.0, 1.0]
    ]
}
```

A.5. MultiLineStrings

Coordinates of a MultiLineString are an array of LineString coordinate arrays:

```
{
    "type": "MultiLineString",
    "coordinates": [
        [
            [100.0, 0.0],
            [101.0, 1.0]
        ],
        [
            [102.0, 2.0],
            [103.0, 3.0]
        ]
    ]
}
```

A.6. MultiPolygons

Coordinates of a MultiPolygon are an array of Polygon coordinate arrays:

```
{
    "type": "MultiPolygon",
    "coordinates": [
        [
            [
                [102.0, 2.0],
                [103.0, 2.0],
                [103.0, 3.0],
                [102.0, 3.0],
                [102.0, 2.0]
            ]
        ],
        [
            [
                [100.0, 0.0],
                [101.0, 0.0],
                [101.0, 1.0],
                [100.0, 1.0],
                [100.0, 0.0]
            ],
            [
                [100.2, 0.2],
                [100.8, 0.2],
                [100.8, 0.8],
                [100.2, 0.8],
                [100.2, 0.2]
            ]
        ]
    ]
}
```

A.7. GeometryCollections

Each element in the geometries array of a GeometryCollection is one of the geometry objects described above:

```
{
    "type": "GeometryCollection",
    "geometries": [{
        "type": "Point",
        "coordinates": [100.0, 0.0]
    }, {
        "type": "LineString",
        "coordinates": [
            [101.0, 0.0],
            [102.0, 1.0]
        ]
    }]
}
```

Appendix B. Contributors

The GeoJSON format is the product of discussion on the GeoJSON mailing list: http://lists.geojson.org/listinfo.cgi/geojson-geojson.org.

Comments are solicited and should be addressed to the GeoJSON mailing list at geojson@lists.geojson.org or to the GeoJSON issue tracker at https://github.com/geojson/draft-geojson/issues.

Authors' Addresses

H. Butler
Hobu Inc.

M. Daly
Cadcorp

A. Doyle
MIT

S. Gillies
Mapbox Inc.

Email: sean.gillies@gmail.com
URI: http://sgillies.net

T. Schaub
Planet Labs

S. Hagen
Rheinaustr. 62
Bonn 53225
DE

Email: stefan@hagen.link
URI: http://stefan-hagen.website/

SIPCORE Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 10, 2015

C. Holmberg
Ericsson
J. Jiang
China Mobile

June 8, 2015

Via header field parameter to indicate received realm
draft-holmberg-dispatch-received-realm-00.txt

Abstract

This specification defines a new Session Initiation Protocol (SIP) Via header field parameter, "received-realm", which allows a SIP entity acting as an entry point to a transit network to indicate from which adjacent upstream network a SIP request is received, using a network realm value associated with the adjacent network.

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 December 10, 2015.

Copyright Notice

Table of Contents

1. Introduction

1.1. General

When SIP sessions are established between networks belonging to different operators, or between interconnected networks belonging to the same operator (or enterprise), the SIP requests might traverse transit network.

Such transit networks might provide different kind of services. In order to do that, a transit network often needs to know to which operator (or enterprise) the adjacent upstream network, from which the SIP session initiation request is received, belongs.

This specification defines a new Session Initiation Protocol (SIP) Via header field parameter, "received-realm", which allows a SIP entity acting as an entry point to a transit network to indicate from which adjacent upstream network a SIP request is received, using a network realm value associated with the adjacent network.

NOTE: As the adjacent network can be an enterprise network, an Inter Operator Identifier (IOI) cannot be used to identity the network, as IOIs are not defined for enterprise networks.

The following sections describe use-case where the information is needed.

1.2. Use-Case: Transit Network Application Services

The 3rd Generation Partnership Project (3GPP) TS 23.228 specifies how an IP Multimedia Subsystem (IMS) network can be used to provide transit functionality. An operator can use its IMS network to provide transit functionality e.g. to non-IMS customers, to enterprise networks, and to other network operators.

The transit network operator can provide application services to the networks for which it is providing transit functionality. Transit application services are typically not provided per user basis, as the transit network does not have access to the user profiles of the networks for which the application services are provided. Instead, the application services are provided per served network.

When a SIP entity that provides application services (e.g. an Application Server) within a transit network receives a SIP request, in order to apply the correct services it needs to know the adjacent upstream network from which the SIP request is received.

1.3. Use-Case: Transit Network Routing

A transit network operator normally interconnects to many diferent operators, including other transit network operators, and provides transit routing of SIP requests received from one operator network towards the destination. The destination can be within an operator network to which the transit network operator has a direct interconnect, or within an operator network that only can be reached via one or more interconnected transit operators.

For each customer, i.e. interconnected network operator for which, the transit network operator routes SIP requests towards the requested destination a set of transit routing polices are defined. These policies are used to determine how a SIP request shall be routed towards the requested destination to meet the agreement the transit network operator has with its customer.

When a SIP entity that performs the transit routing functionality receives a SIP request, in order to apply the correct set of transit routing policies, it needs to know from which of its customers, i.e. adjacent upstream network, the SIP request is received.

2. Applicability

The mechanism defined in this specification MUST only be used by SIP entities that are able to verify from which adjacent upstream network a SIP request is received.

The mechanism for verifying from which adjacent upstream network a SIP request is received is outside the scope of this specification.

3. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [RFC2119].

4. Definitions

SIP entity: SIP User Agent (UA), or SIP proxy, as defined in RFC 3261.

Adjacent upstream SIP network: The adjacent SIP network in the direction from which a SIP request is received.

Network entry point: A SIP entity on the border of network, which receives SIP requests from adjacent upstream networks.

Inter Operator Identifier (IOI): A globally unique identifier to correlate billing information generated within the IP Multimedia Subsystem (IMS).

5. User Agent and Proxy behavior

5.1. General

This section describes how a SIP entity, acting as an entry point to a network, uses the "received-realm" Via header field parameter.

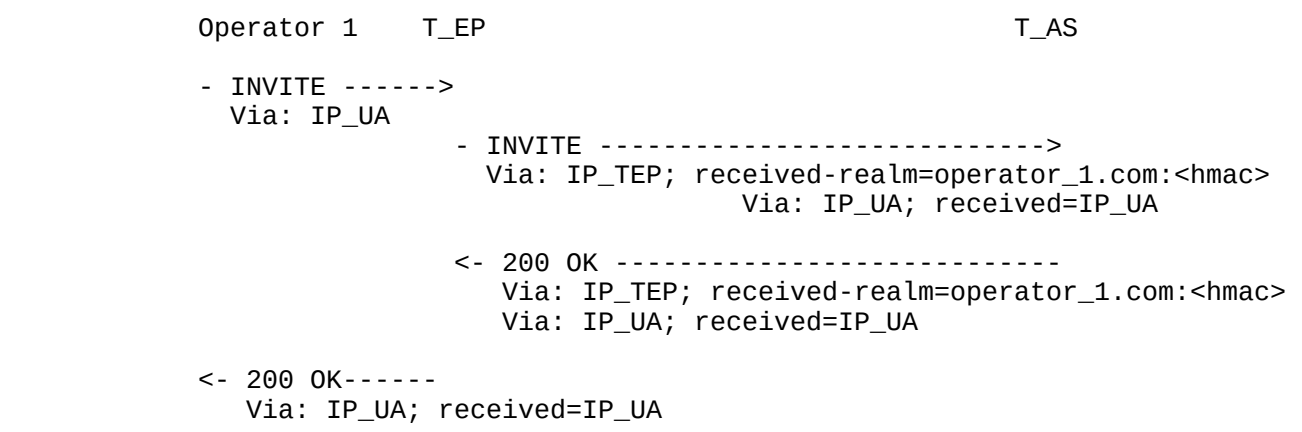
5.2. Behavior of a SIP entity acting as a network entry point

When a SIP entity, acting as a network entry point, forwards a SIP request, or initiates a SIP request on its own (e.g. a PSTN gateway), the SIP entity adds a Via header field to the SIP request, according to the procedures in RFC 3261 [RFC3261]. In addition, if the SIP entity is able to assert the adjacent upstream network, and if the SIP entity is aware of a network realm value defined for that network, the SIP entity can add a "received-realm" Via header field

parameter, conveying the network realm value, to the Via header field added to the SIP request.

When the SIP entity adds a "received-realm" Via header field parameter to a SIP request, it MUST also calculate a Hash-based message authentication code (HMAC) [RFC2104] value from the parameter value, using a secret key which is shared between the SIP entity and any SIP entity which will use the parameter value. The HMAC is then added to the parameter.

6. Example



7. Syntax

7.1. General

This section describes the syntax extensions to the ABNF syntax defined in RFC 3261 [RFC3261], by defining a new Via header field parameter, "received-realm". The ABNF defined in this specification is conformant to RFC 5234 [RFC5234]. "EQUAL", "COLON" and "hostname" are defined in RFC 3261 [RFC3261]. "DIGIT" is defined in RFC 5234 [RFC5234]

7.2. ABNF

```
via-params =/ received-realm

received-realm = "received-realm" EQUAL hostname COLON hmac

hmac = TBD
```

8. IANA Considerations

8.1. 'received-realm' Via header field parameter

This specification defines a new Via header field parameter called received-realm in the "Header Field Parameters and Parameter Values" sub-registry as per the registry created by [RFC3968]. The syntax is defined in Section 7. The required information is:

Header Field	Parameter Name	Predefined Values	Reference
Via	received-realm	No	RFCXXXX

9. Security Considerations

As the received-realm Via header field parameter can be used to trigger applications, it is important to ensure that the parameter has not been added to the SIP message by an unauthorized SIP entity.

The operator MUST change the key on a frequent basis. The operator also needs to take great care in ensuring that the key used to calculate the Hash-based message authentication code (HMAC) value is only known by the network entry point adding the received-realm Via header field parameter to a SIP message and the entities that use the parameter value.

A SIP entity MUST NOT use the parameter value it does not match the associated HMAC value. The SIP entity MUST trigger an alarm, or use a similar mechanism, to inform the operator about the mismatch.

A SIP entity MUST use different key values for each parameter value that it recognizes and use to trigger actions.

10. Acknowledgements

TBD

11. Change Log

[RFC EDITOR NOTE: Please remove this section when publishing]

Changes from draft-holmberg-received-realm-04

- Changed IETF WG from sipcore do dispatch.
- HMAC value added to the parameter.

Changes from draft-holmberg-received-realm-03

- New version due to expiration.

Changes from draft-holmberg-received-realm-02

- New version due to expiration.

Changes from draft-holmberg-received-realm-01

- New version due to expiration.

Changes from draft-holmberg-received-realm-00

- New version due to expiration.

12. References

12.1. Normative References

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

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

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

12.2. Informative References

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[RFC3968] Camarillo, G., "The Internet Assigned Number Authority (IANA) Header Field Parameter Registry for the Session Initiation Protocol (SIP)", BCP 98, RFC 3968, December 2004.

Authors' Addresses

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: christer.holmberg@ericsson.com

Yi Jiang
China Mobile
No.32 Xuanwumen West Street
Beijing Xicheng District 100053
P.R. China

Email: jiangyi@chinamobile.com

SIPCORE Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 6, 2017

C. Holmberg
Ericsson
J. Jiang
China Mobile
December 3, 2016

Session Initiation Protocol (SIP) Via header field parameter to indicate received realm
draft-holmberg-dispatch-received-realm-12.txt

Abstract

This specification defines a new Session Initiation Protocol (SIP) Via header field parameter, "received-realm", which allows a SIP entity acting as an entry point to a transit network to indicate from which adjacent upstream network a SIP request is received, using a network realm value associated with the adjacent network.

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 June 6, 2017.

Copyright Notice

Table of Contents

1. Introduction

1.1. General

When Session Initiation Protocol (SIP) [RFC3261] sessions are established between networks belonging to different operators, or between interconnected networks belonging to the same operator (or enterprise), the SIP requests associated with the session might traverse transit networks.

Such transit networks might provide different kinds of services. In order to provide such services, a transit network often needs to know to which operator (or enterprise) the adjacent upstream network from which the SIP session initiation request is received belongs.

This specification defines a new Session Initiation Protocol (SIP) Via header field parameter, "received-realm", which allows a SIP entity acting as an entry point to a transit network to indicate from which adjacent upstream network a SIP request is received, using a network realm value associated with the adjacent network.

NOTE: As the adjacent network can be an enterprise network, an Inter Operator Identifier (IOI) cannot be used to identity the network, as IOIs are not defined for enterprise networks.

The following sections describe use-cases where the information is needed.

1.2. Use-Case: Transit Network Application Services

The 3rd Generation Partnership Project (3GPP) TS 23.228 [TS.3GPP.23.228] specifies how an IP Multimedia Subsystem (IMS) network can be used to provide transit functionality. An operator can use its IMS network to provide transit functionality e.g., to non-IMS customers, to enterprise networks, and to other network operators.

The transit network operator can provide application services to the networks for which it is providing transit functionality. Transit application services are typically not provided on a per user basis, as the transit network does not have access to the user profiles of the networks for which the application services are provided. Instead, the application services are provided per served network.

When a SIP entity that provides application services (e.g., an Application Server) within a transit network receives a SIP request, in order to apply the correct services, it needs to know the adjacent upstream network from which the SIP request is received.

1.3. Use-Case: Transit Network Routing

A transit network operator normally interconnects to many different operators, including other transit network operators, and provides transit routing of SIP requests received from one operator network towards the destination. The destination can be within an operator network to which the transit network operator has a direct interconnect, or within an operator network that only can be reached via one or more interconnected transit operators.

For each customer, i.e., interconnected network operator, for which the transit network operator routes SIP requests towards the requested destination, a set of transit routing polices are defined. These policies are used to determine how a SIP request shall be routed towards the requested destination to meet the agreement the transit network operator has with its customer.

When a SIP entity that performs the transit routing functionality receives a SIP request, in order to apply the correct set of transit routing policies, it needs to know from which of its customers, i.e., adjacent upstream network, the SIP request is received.

2. Applicability

The mechanism defined in this specification MUST only be used by SIP entities that are able to verify from which adjacent upstream network a SIP request is received.

The mechanism for verifying from which adjacent upstream network a SIP request is received is outside the scope of this specification. Such mechanism might be based for instance on receiving the SIP request on an authenticated Virtual Private Network (VPN), receiving the SIP request on a specific IP address, or on a specific network access.

3. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [RFC2119].

4. Definitions

SIP entity: SIP User Agent (UA), or SIP proxy, as defined in RFC 3261.

Adjacent upstream SIP network: The adjacent SIP network in the direction from which a SIP request is received.

Network entry point: A SIP entity on the border of network, which receives SIP requests from adjacent upstream networks.

Inter Operator Identifier (IOI): A globally unique identifier to correlate billing information generated within the IP Multimedia Subsystem (IMS).

JWS: JSON Web Signature, as defined in [RFC7515].

5. Via ’received-realm’ header field parameter

5.1. General

The Via ’received-realm’ header field parameter value is represented as a combination of an operator identifier, whose value represents the adjacent network, and a serialized JSON Web Signature (JWS) [RFC7515]. The JWS Payload consists of the operator identifier and other SIP information element values.

The procedures for encoding the JWS and calculating the signature are defined in [RFC7515]. As the JWS Payload information is found in other SIP information elements, the JWS Payload is detached from the serialized JWS conveyed in the header field parameter, as described in Appendix F of [RFC7515]. The operator identifier and the serialized JWS are separated using a colon character.

5.2. Operator Identifier

The Operator Identifier is a token value that represents the adjacent operator. The scope of the value is only within the network that inserts the value.

The Operator Identifier value is case-insensitive.

5.3. JWS Header

The following header parameters MUST be included in the JWS.

- The "typ" parameter MUST have a "JWT" value.
- The "alg" parameter MUST have the value of the algorithm used to calculate the JWS.

NOTE: Operators need to agree on the set of supported algorithms for calculating the JWT signature.

NOTE: The "alg" parameter values for specific algorithms are listed in the IANA JSON Web Signature and Encryption Algorithms sub-registry of the JSON Object Signing and Encryption (JOSE) registry. Operators need to use algorithms for which an associated "alg" parameter value has been registered. The procedures for defining new values are defined in [RFC7518].

Example:

```
{
        "typ":"JWT",
        "alg":"HS256"
}
```

5.4. JWS Payload

The following claims MUST be included in the JWS Payload:

- The "sip_from_tag" claim has the value of the From ’tag’ header field parameter of the SIP message.
- The "sip_date" claim has the value of the Date header field in the SIP message, encoded in JSON NumericDate format [RFC7519].
- The "sip_callid" claim has have value of the Call-ID header field in the SIP message.
- The "sip_cseq_num" claim has the numeric value of the CSeq header field in the SIP message.
- the "sip_via_branch" claim has value of the Via branch header field parameter of the Via header field, in the SIP message, to which the received-realm header field parameter is attached.
- the "sip_via_opid" claim has value of the operator identifier part of the Via received-realm header field parameter of the Via header field, in the SIP message, to which the received-realm header field parameter is attached.

Example:

```
{
        "sip_from_tag":"1928301774",
        "sip_date":1472815523,
        "sip_callid":"a84b4c76e66710@pc33.atlanta.com",
        "sip_cseq_num":"314159",
        "sip_via_branch":"z9hG4bK776asdhds",
        "sip_via_opid":"myoperator"
}
```

5.5. JWS Serialization

As the JWS Payload is not carried in the received-realm parameter, in order to make sure that the sender and the receiver construct the JWS Payload object in the same way, the JSON representation of the JWS Payload object it MUST be computed as follows:

- All claims MUST be encoded using lower case characters.
- The claims MUST be in the same order as listed in [Section 5.4].
- All claims except "sip_date" MUST be encoded as StringOrURI JSON string value [RFC7519].
- The "sip_date" claim MUST be encoded as a JSON NumericDate value [RFC7519].
- The JWS Payload MUST follow the rules for the construction of the thumbprint of a JSON Web Key (JWK) as defined in [RFC7638] Section 3 Step 1 only.

Example:

NOTE: Line breaks for display purpose only

```
{"sip_from_tag":"1928301774","sip_date":1472815523,
"sip_callid":"a84b4c76e66710@pc33.atlanta.com","sip_cseq_num":"314159",
"sip_via_branch":"z9hG4bK776asdhds","sip_via_opid":"myoperator"}
```

5.6. Syntax

5.6.1. General

This section describes the syntax extensions to the ABNF syntax defined in [RFC3261], by defining a new Via header field parameter, "received-realm". The ABNF defined in this specification is conformant to RFC 5234 [RFC5234]. "EQUAL", "LDQUOT", "RDQUOT" and "ALPHA" are defined in [RFC3261]. "DIGIT" is defined in [RFC5234].

5.6.2. ABNF

```
via-params     =/ received-realm
received-realm = "received-realm" EQUAL LDQUOT op-id COLON jws RDQUOT
op-id          = token
jws            = header ".." signature
header         = 1*base64-char
signature      = 1*base64-char
base64-char    = ALPHA / DIGIT / "/" / "+"
```

EQUAL, COLON, token, LDQUOT, RDQUOT, ALPHA and DIGIT as defined in [RFC3261].

NOTE: The two adjacent dots in the jws part is due to the detached payload being replaced by an empty string [RFC7515].

5.7. Example

```
Via: SIP/2.0/UDP pc33.example.com;branch=z9hG4bK776;
received-realm="myoperator:eyJ0eXAiOiJKV1QiLA0KICJhbGciOiJIUzI1N..
dBjftJeZ4CVP-mB92K27uhbUJU1p1r_wW1gFWFOEjXk"
```

NOTE: Line breaks for display purpose only

6. User Agent and Proxy behavior

6.1. General

This section describes how a SIP entity, acting as an entry point to a network, uses the "received-realm" Via header field parameter.

6.2. Behavior of a SIP entity acting as a network entry point

When a SIP entity, acting as a network entry point, forwards a SIP request, or initiates a SIP request on its own (e.g., a PSTN gateway), the SIP entity adds a Via header field to the SIP request, according to the procedures in RFC 3261 [RFC3261]. In addition, if the SIP entity is able to assert the adjacent upstream network, and if the SIP entity is aware of a network realm value defined for that network, the SIP entity can add a "received-realm" Via header field parameter, conveying the network realm value as the operator identifier (Section 5.2) part of the header field parameter, to the Via header field added to the SIP request.

In addition the SIP entity MUST also calculate a JWS (Section 5.4) and add the calculated JWS Header and JWS Signature as the jws part of the Via header field parameter.

6.3. Behavior of a SIP entity consuming the received-network value

When a SIP entity receives a Via 'received-network' header field parameter, and intends to perform actions based on the header field parameter value, it MUST first re-calculate the JWS and check whether the result matches the JWS received. If there is not a match, the SIP entity MUST discard the received 'received-network' header field parameter. The SIP entity MAY take also take additional actions (e.g., rejecting the SIP request) based on local policy.

7. Example

```
 Operator 1    T_EP                                  T_AS

 - INVITE ------>
   Via: SIP/2.0/UDP IP_UA
                 -- INVITE ---------------------------->
                    Via: SIP/2.0/UDP IP_TEP;branch=z9hG4bK776;
                     received-realm="myoperator:eyJ0eXAiOiJKV1QiLA0KICJh
                     bGciOiJIUzI1N..dBjftJeZ4CVP-mB92K27uhbUJU1p1r_wW
                     1gFWFOEjXk"
                    Via: SIP/2.0/UDP IP_UA; received=IP_UA

                 <- 200 OK ----------------------------
                    Via: SIP/2.0/UDP IP_TEP;branch=z9hG4bK776;
                     received-realm="myoperator:eyJ0eXAiOiJKV1QiLA0KICJh
                     bGciOiJIUzI1N..dBjftJeZ4CVP-mB92K27uhbUJU1p1r_wW
                     1gFWFOEjXk"
                    Via: SIP/2.0/UDP IP_UA; received=IP_UA

 <- 200 OK------
    Via: SIP/2.0/UDP IP_UA; received=IP_UA
```

8. IANA Considerations

8.1. 'received-realm' Via header field parameter

This specification defines a new Via header field parameter called received-realm in the "Header Field Parameters and Parameter Values" sub-registry as per the registry created by [RFC3968]. The syntax is defined in Section 5.6. The required information is:

Header Field	Parameter Name	Predefined Values	Reference
Via	received-realm	No	RFCXXXX

8.2. JSON Web Token Claims Registration

This specification defines new JSON Web Token claims in the "JSON Web Token Claims" sub-registry as per the registry created by [RFC7519].

Claim Name: "sip_from_tag"

Claim : SIP From tag header field parameter value

Change Controller: IESG

Specification Document(s): RFC XXXX, RFC 3261

Claim Name: "sip_date"

Claim Description: SIP Date header field value

Change Controller: IESG

Specification Document(s): RFC XXXX, RFC 3261

Claim Name: "sip_callid"

Claim Description: SIP Call-Id header field value

Change Controller: IESG

Specification Document(s): RFC XXXX, RFC 3261

Claim Name: "sip_cseq_num"

Claim Description: SIP CSeq numeric header field parameter value

Change Controller: IESG

Specification Document(s): RFC XXXX, RFC 3261

Claim Name: "sip_via_branch"

Claim Description: SIP Via branch header field parameter value

Change Controller: IESG

Specification Document(s): RFC XXXX, RFC 3261

9. Security Considerations

As the received-realm Via header field parameter can be used to trigger applications, it is important to ensure that the parameter has not been added to the SIP message by an unauthorized SIP entity.

The received-realm Via header field parameter is inserted, signed, verified and consumed within an operator network. The operator MUST discard parameters received from another network, and the parameter MUST only be inserted by SIP entities that are able to verify from which adjacent upstream network a SIP request is received.

The operator also needs to take great care in ensuring that the key used to calculate the JWS signature value is only known by the network entities signing and adding the JWS signature to the received-realm Via header field parameter of a SIP message, and to network entities verifying and consuming the parameter value.

The operator MUST use a key management policy that protects agains unauthorised access to the stored keys within nodes where they keys associated with the JWS signature are stored, and that protects against crypto analysis attacks using captured data sent on the wire.

A SIP entity MUST NOT use the adjacent network information if there is a mismatch between the JWS signature received in the SIP header field and the JWS signature calculated by the receiving entity.

Generic security considerations for JWS are defined in [RFC7515].

10. Acknowledgments

Thanks to Adam Roach and Richard Barnes for providing comments and feedback on the document. Francis Dupoint performed the Gen-ART review.

11. Change Log

[RFC EDITOR NOTE: Please remove this section when publishing]

Changes from draft-holmberg-dispatch-received-realm-11

- Editorial change based on IESG review.

Changes from draft-holmberg-dispatch-received-realm-10

- Changes based on IESG review.

- Changes based on SecDIR review.
- Changes based on IANA Service Operator review.

Changes from draft-holmberg-dispatch-received-realm-09

- Reference to RFC 2104 removed.

Changes from draft-holmberg-dispatch-received-realm-08

- Editorial fixed based on Gen-ART review.
- Editorial fixes based on comments from IANA Service Operator review.
- - Quotes removed from sip_date value.

Changes from draft-holmberg-dispatch-received-realm-07

- Editorial changes.

Changes from draft-holmberg-dispatch-received-realm-06

- Changes based on AD review by Ben Campbell:
- - operator-id added to JSON Payload

Changes from draft-holmberg-dispatch-received-realm-05

- Editorial fixes.

Changes from draft-holmberg-dispatch-received-realm-04

- JSON serialization procedure added.

Changes from draft-holmberg-dispatch-received-realm-03

- ABNF correction.

Changes from draft-holmberg-dispatch-received-realm-02

- JWT replaced with JWS.
- Appendix F of RFC 7515 applied.

Changes from draft-holmberg-dispatch-received-realm-01

- Define received-realm parameter value as a JSON Web Token (JWT).

Changes from draft-holmberg-dispatch-received-realm-00

- New version due to expiration of previous version.

Changes from draft-holmberg-received-realm-04

- Changed IETF WG from sipcore do dispatch.
- HMAC value added to the parameter.

Changes from draft-holmberg-received-realm-03

- New version due to expiration.

Changes from draft-holmberg-received-realm-02

- New version due to expiration.

Changes from draft-holmberg-received-realm-01

- New version due to expiration.

Changes from draft-holmberg-received-realm-00

- New version due to expiration.

12. References

12.1. 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>.

[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>.

[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>.

[RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <http://www.rfc-editor.org/info/rfc7515>.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <http://www.rfc-editor.org/info/rfc7519>.

[RFC7638] Jones, M. and N. Sakimura, "JSON Web Key (JWK) Thumbprint", RFC 7638, DOI 10.17487/RFC7638, September 2015, <http://www.rfc-editor.org/info/rfc7638>.

12.2. Informative References

[RFC3968] Camarillo, G., "The Internet Assigned Number Authority (IANA) Header Field Parameter Registry for the Session Initiation Protocol (SIP)", BCP 98, RFC 3968, DOI 10.17487/RFC3968, December 2004, <http://www.rfc-editor.org/info/rfc3968>.

[RFC7518] Jones, M., "JSON Web Algorithms (JWA)", RFC 7518, DOI 10.17487/RFC7518, May 2015, <http://www.rfc-editor.org/info/rfc7518>.

[TS.3GPP.23.228] 3GPP, "IP Multimedia Subsystem (IMS); Stage 2", 3GPP TS 23.228 14.1.0, September 2016.

Authors' Addresses

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: christer.holmberg@ericsson.com

Yi Jiang
China Mobile
No.32 Xuanwumen West Street
Beijing Xicheng District 100053
P.R. China

Email: jiangyi@chinamobile.com

Dispatch
Internet-Draft
Updates: RFC6442 (if approved)
Intended status: Standards Track
Expires: December 3, 2015

J. Winterbottom
Winterb Consulting Services
L. Liess
Deutsche Telekom
B. Chatras
Orange Labs
A. Hutton
Unify

June 1, 2015

Location Source Parameter for the SIP Geolocation Header Field
draft-winterbottom-dispatch-locparam-00.txt

Abstract

There are some circumstances where a geolocation header field may contain more than one location value. Knowing the identity of the node adding the location value allows the recipient more freedom in selecting the value to look at first rather than relying solely on the order of the location values.

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 December 3, 2015.

Copyright Notice

Table of Contents

1. Introduction

The SIP geolocation specification [RFC6442] describes a SIP header field that is used to indicate that the SIP message is conveying location information. The specification suggests that only one location value should be conveyed. However, some communications architectures, such as 3GPP [TS23-167] and ETSI [M493], prefer to use information provided by edge-proxies or acquired through the use of core-network nodes, before using information provided solely by user equipment (UE). These solutions don't preclude the use of UE provided location but require a means of being able to distinguish the identity of the node adding the location value to the SIP message from that provided by the UE. [RFC6442] stipulates that the order of location values in the geolocation header field aligns with the order in which they were added to the header field. Whilst this order provides guidance to the recipient as to which values were added to the message earlier in the communication chain, it does not provide any indication of which node actually added the location value. Knowing the identity of the entity that added the location to the message allows the recipient to choose which location to consider first rather than relying solely on the order of the location values in the geolocation header field.

This document adds a location-source (loc-src) parameter to the location values in [RFC6442] so that the entity adding the location value to geolocation header field can identify itself using its hostname. How the entity adding the location value to the header field obtains the location information is out of scope of this document.

2. Terminology

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

3. Rationale

The primary intent of the parameter defined in this specific is for use in emergency calling. There are various architectures defined for providing emergency calling using SIP-based messaging. Each has it own characteristics with corresponding pros and cons. All of them allow the UE to provide location information, however, many also attach other sources of location information to support veracity checks, provide backup information, or to be used as the primary location. This document makes no attempt to comment on these various architectures or the rationale for them wishing to include multiple location values. It does recognize that these architectures exist

and that there is a need to identify the entity adding the location information.

4. Mechanism

The mechanism employed adds a parameter to the location value defined in [RFC6442] that identifies the hostname of the entity adding the location value to the geolocation header field. The Augmented BNF (ABNF) [RFC5234] for this parameter is shown in Figure 1.

```
location-source = "loc-src=" (host / other-loc-src)
other-loc-src = token
```

Figure 1: Location Source

Any proxy adding a location value to a geolocation header field SHOULD also add its host name using the loc-src parameter so that it is clearly identified as the node adding the location. A UE MUST NOT provide a loc-src parameter value. If a proxy receives a message from an untrusted source with the loc-src parameter set then it MUST remove the loc-src parameter before passing the message into a trusted network.

5. Example

The following example shows a SIP INVITE message containing a geolocation header field with two location values. The first location value points to a PIDF-LO in the SIP body using a content-indirection (cid:) URI per [RFC4483] and this is provided by the UE. The second location value is an https URI the provided by a proxy which identifies itself using the loc-src parameter.

```
INVITE sips:bob@biloxi.example.com SIP/2.0
Via: SIPS/2.0/TLS pc33.atlanta.example.com;branch=z9hG4bK74bf9
Max-Forwards: 70
To: Bob <sips:bob@biloxi.example.com>
From: Alice <sips:alice@atlanta.example.com>;tag=9fxced76sl
Call-ID: 3848276298220188511@atlanta.example.com
Geolocation: <cid:target123@atlanta.example.com>,
     <https://lis.example.com:8222/y77syc7cuecbh>;
               loc-src=edgeproxy.example.com
Geolocation-Routing: yes
Accept: application/sdp, application/pidf+xml
CSeq: 31862 INVITE
Contact: <sips:alice@atlanta.example.com>
Content-Type: multipart/mixed; boundary=boundary1
Content-Length: ...
```

Figure 2: Example Location Request.

6. Privacy Considerations

This document doesn’t change any of the privacy considerations described in [RFC6442]. While the addition of the loc-src parameter does provide an indicator of the entity that added the location in the signaling path this provides little more exposure than a proxy identity being added to the record-route header field.

7. Security Considerations

This document introduces the ability of a proxy or middle box to insert a host name indicating the that they added the specific location value to the geolocation header field. The intent is for this field to be used by the location recipient in the event that the SIP message contains multiple location values. As a consequence this parameter should only be used by the location recipient in a trusted network.

8. IANA Considerations

8.1. Registration of loc-src Parameter for geolocation header field

This document calls for IANA to register a new SIP header parameter as per the guidelines in [RFC3261], which will be added to header sub-registry under http://www.iana.org/assignments/sip-parameters.

Header Field: geolocation

Parameter Name: loc-src

9. Acknowledgements

NONE

10. References

10.1. Normative References

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

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

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

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

10.2. Informative References

[M493] European Telecommunications Standards Institute, "Functional architecture to support European requirements on emergency caller location determination and transport", ES 203 178, V 1.0.5, December 2014.

[RFC4483] Burger, E., "A Mechanism for Content Indirection in Session Initiation Protocol (SIP) Messages", RFC 4483, May 2006.

[TS23-167] 3rd Generation Partnership Project, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; IP Multimedia Subsystem (IMS) emergency sessions", TS 23.167, V 12.1.0, March 2015.

Authors' Addresses

James Winterbottom
Winterb Consulting Services
Gwynneville, NSW 2500
AU

Phone: +61 448 266004
Email: a.james.winterbottom@gmail.com

Laura Liess
Deutsche Telekom
Heinrich-Hertz Str, 3-7
Darmstadt 64295
Germany

Email: l.liess@telekom.de
URI: www.telekom.de

Bruno Chatras
Orange Labs
38-40 rue du General Leclerc
Issy Moulineaux Cedex 9 F-92794
France

Email: bruno.chatras@orange.com

Andrew Hutton
Unify
Technology Drive
Nottingham NG9 1LA
UK

Email: andrew.hutton@unify.com

Dispatch
Internet-Draft
Updates: RFC6442 (if approved)
Intended status: Standards Track
Expires: July 3, 2017

J. Winterbottom
Winterb Consulting Services
R. Roland
L. Liess
Deutsche Telekom
B. Chatras
Orange Labs
A. Hutton
Unify

December 30, 2016

Location Source Parameter for the SIP Geolocation Header Field
draft-winterbottom-dispatch-locparam-02.txt

Abstract

There are some circumstances where a geolocation header field may contain more than one location value. Knowing the identity of the node adding the location value allows the recipient more freedom in selecting the value to look at first rather than relying solely on the order of the location values.

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 July 3, 2017.

Copyright Notice

Table of Contents

1. Introduction

The SIP geolocation specification [RFC6442] describes a SIP header field that is used to indicate that the SIP message is conveying location information. The specification suggests that only one location value should be conveyed. However, some communications architectures, such as 3GPP [TS23-167] and ETSI [M493], prefer to use information provided by edge-proxies or acquired through the use of core-network nodes, before using information provided solely by user equipment (UE). These solutions don't preclude the use of UE provided location but require a means of being able to distinguish the identity of the node adding the location value to the SIP message from that provided by the UE. [RFC6442] stipulates that the order of location values in the geolocation header field aligns with the order in which they were added to the header field. Whilst this order provides guidance to the recipient as to which values were added to the message earlier in the communication chain, it does not provide any indication of which node actually added the location value. Knowing the identity of the entity that added the location to the message allows the recipient to choose which location to consider first rather than relying solely on the order of the location values in the geolocation header field.

This document adds a location-source (loc-src) parameter to the location values in [RFC6442] so that the entity adding the location value to geolocation header field can identify itself using its hostname. How the entity adding the location value to the header field obtains the location information is out of scope of this document.

2. Terminology

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

3. Rationale

The primary intent of the parameter defined in this specific is for use in emergency calling. There are various architectures defined for providing emergency calling using SIP-based messaging. Each has it own characteristics with corresponding pros and cons. All of them allow the UE to provide location information, however, many also attach other sources of location information to support veracity checks, provide backup information, or to be used as the primary location. This document makes no attempt to comment on these various architectures or the rationale for them wishing to include multiple location values. It does recognize that these architectures exist

and that there is a need to identify the entity adding the location information.

The parameter defined in this specification adds the location source generating the location value to increase the trustworthiness of the location information. Thus it is intended to use this parameter in trust domains where Spec(T) as described in [RFC3325] exists only. The functional architecture described within ETSI [M493] is an example of architecture where this parameter makes sense to be used.

4. Mechanism

The mechanism employed adds a parameter to the location value defined in [RFC6442] that identifies the hostname of the entity adding the location value to the geolocation header field. The Augmented BNF (ABNF) [RFC5234] for this parameter is shown in Figure 1.

```
location-source = "loc-src=" (host / other-loc-src)
other-loc-src = token
```

Figure 1: Location Source

Only a fully qualified host name is valid, an IP address MUST NOT be added by an entity conforming with this specification. If a node conforming to this specification receives a geolocation header field with a loc-src parameter containing an IP address then the parameter MUST be removed.

Any proxy adding a location value to a geolocation header field SHOULD also add its host name using the loc-src parameter so that it is clearly identified as the node adding the location. A UE MUST NOT provide a loc-src parameter value. If a proxy receives a message from an untrusted source with the loc-src parameter set then it MUST remove the loc-src parameter before passing the message into a trusted network.

5. Example

The following example shows a SIP INVITE message containing a geolocation header field with two location values. The first location value points to a PIDF-LO in the SIP body using a content-indirection (cid:) URI per [RFC4483] and this is provided by the UE. The second location value is an https URI the provided by a proxy which identifies itself using the loc-src parameter.

```
INVITE sips:bob@biloxi.example.com SIP/2.0
Via: SIPS/2.0/TLS pc33.atlanta.example.com;branch=z9hG4bK74bf9
Max-Forwards: 70
To: Bob <sips:bob@biloxi.example.com>
From: Alice <sips:alice@atlanta.example.com>;tag=9fxced76sl
Call-ID: 3848276298220188511@atlanta.example.com
Geolocation: <cid:target123@atlanta.example.com>,
     <https://lis.example.com:8222/y77syc7cuecbh>;
               loc-src=edgeproxy.example.com
Geolocation-Routing: yes
Accept: application/sdp, application/pidf+xml
CSeq: 31862 INVITE
Contact: <sips:alice@atlanta.example.com>
Content-Type: multipart/mixed; boundary=boundary1
Content-Length: ...
```

Figure 2: Example Location Request.

6. Privacy Considerations

This document doesn’t change any of the privacy considerations described in [RFC6442]. While the addition of the loc-src parameter does provide an indicator of the entity that added the location in the signaling path this provides little more exposure than a proxy identity being added to the record-route header field.

7. Security Considerations

This document introduces the ability of a proxy or middle box to insert a host name indicating the that they added the specific location value to the geolocation header field. The intent is for this field to be used by the location recipient in the event that the SIP message contains multiple location values. As a consequence this parameter should only be used by the location recipient in a trusted network.

The use of this parameter is not restricted to a specific architecture but using multiples locations and loc-src may end in compatibility issues. [RFC6442] already addresses the issue of multiples locations. To avoid problems of wrong interpretation of loc-src the value may be discarded when passed to an other domain.

8. IANA Considerations

8.1. Registration of loc-src Parameter for geolocation header field

This document calls for IANA to register a new SIP header parameter as per the guidelines in [RFC3261], which will be added to header sub-registry under http://www.iana.org/assignments/sip-parameters.

Header Field: geolocation

Parameter Name: loc-src

9. Acknowledgements

NONE

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>.

[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>.

[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>.

[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>.

[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>.

10.2. Informative References

[M493] European Telecommunications Standards Institute, "Functional architecture to support European requirements on emergency caller location determination and transport", ES 203 178, V 1.1.1, Februar 2015.

[RFC4483] Burger, E., Ed., "A Mechanism for Content Indirection in Session Initiation Protocol (SIP) Messages", RFC 4483, DOI 10.17487/RFC4483, May 2006, <http://www.rfc-editor.org/info/rfc4483>.

[TS23-167] 3rd Generation Partnership Project, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; IP Multimedia Subsystem (IMS) emergency sessions", TS 23.167, V 12.1.0, March 2015.

Authors' Addresses

James Winterbottom
Winterb Consulting Services
Gwynneville, NSW 2500
AU

Phone: +61 448 266004
Email: a.james.winterbottom@gmail.com

Roland Jesske
Deutsche Telekom
Heinrich-Hertz Str, 3-7
Darmstadt 64295
Germany

Email: r.jesske@telekom.de
URI: www.telekom.de

Laura Liess
Deutsche Telekom
Heinrich-Hertz Str, 3-7
Darmstadt 64295
Germany

Email: l.liess@gmail.com

Bruno Chatras
Orange Labs
38-40 rue du General Leclerc
Issy Moulineaux Cedex 9 F-92794
France

Email: bruno.chatras@orange.com

Andrew Hutton
Unify
Technology Drive
Nottingham NG9 1LA
UK

Email: andrew.hutton@unify.com