An Extension to the Session Initiation Protocol (SIP) for Endpoint Session View
draft-boulton-sip-endpoint-view-00

Status of this Memo

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

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

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on September 27, 2009.

Copyright Notice

Copyright (c) 2009 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 in effect on the date of publication of this document (http://trustee.ietf.org/license-info). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.
Abstract

This document defines a standard mechanism for capturing and providing important session information associated with the Session Initiation Protocol (SIP). Certain properties of a SIP protocol exchange are essential for further independent signalling interactions. In certain environments this information can be lost when traversing entities such as Back-to-Back User Agents (B2BUA). This document defines a new optional SIP header, Endpoint-View, for capturing appropriate information.

Table of Contents

1. Introduction ................................................. 3
   1.1. Background and Overview ................................. 3
   1.2. Conventions and Terminology .............................. 6
2. Requirements .................................................. 7
3. UAC Behavior - INVITE generation ............................. 8
4. UAS Behavior .................................................. 9
5. Reliable Provisional .......................................... 10
6. UAC Behavior - ACK generation ............................... 11
7. B2BUA Behavior ............................................... 12
8. Security and Privacy ......................................... 13
9. The Endpoint-View Header .................................... 14
10. Example ....................................................... 15
11. Security Considerations .................................... 18
12. IANA Considerations ........................................ 19
   12.1. Header Field ........................................ 19
   12.2. SIP Option Tag ...................................... 19
13. Acknowledgments ............................................. 20
14. References .................................................. 21
   14.1. Normative References .................................. 21
   14.2. Informative References ................................ 22
Authors’ Addresses .............................................. 23
1. Introduction

1.1. Background and Overview

The Session Initiation Protocol (SIP) has emerged as one of the primary multimedia establishment protocols for Voice Over IP (VOIP). Since its conception it has seen increasing usage as more legacy systems are replaced by IP based solutions and requirements for general multimedia content increases.

RFC 3261 [RFC3261] defines the concept of a back-to-back user agent (B2BUA). Such entities have emerged as an important component in VOIP architectures as they allow a central focus of control for both protocol signalling and potentially media. B2BUA’s are considered a logical role composed of SIP User agents acting independently from a protocol signalling perspective but yet appear as a single entity externally. Figure 1 illustrates a simple B2BUA logical function.

```
+--------------------+
|        B2BUA       |
SIP    |  +-----+  +-----+  |    SIP
<--------| UA  |  | UA  |<-|--------->
| +-----+  +-----+  |
+--------------------+
```

Figure 1: B2BUA

The logical role played by a B2BUA is represented by the outer box from Figure 1. To the outside world it appears as single entity. Internally the B2BUA consists of two User Agents which represent independent SIP signalling entities with the endpoint clients. Both independent SIP signalling legs have unique dialog identifiers, as defined in RFC 3261 [RFC3261] and are combined using application logic. For example, a SIP INVITE request would arrive at the left hand side of Figure 1 and could be passed out the other side to another SIP entity. The left hand side and right hand side dialog identifiers (SIP Call-ID header, To header tag parameter and From Header tag parameter) would differ.

In general, a B2BUA would need to map and translate certain SIP headers and parameters to maintain valid information on each side of a B2BUA. A good example is demonstrated when using the SIP REFER method in conjunction with an existing INVITE dialog. Some of the signalling has been left out for simplicity.
In this example ‘UA1’ has INVITE dialogs set up with both ‘UA2’ and ‘UA3’. Both INVITE dialogs have traversed a B2BUA before reaching ‘UA2’ and ‘UA3’. (1) and (2) represent the two SIP INVITE dialogs when ‘UA1’ sends an INVITE to UA2 (through B2BUA1). (3) and (4) represents the two INVITE dialogs when UA1 sends an INVITE to UA3 (through B2BUA2). At some later time, ‘UA1’ issues a SIP REFER request to ‘UA2’ (in or out of dialog) with a SIP Replaces [RFC3891] header. The ‘Refer-To’ header and ‘Replaces’ parameter are constructed based on ‘UA1’s knowledge of the call with ‘UA3’ - which is represented by (3). On receiving the REFER request from ‘UA1’, ‘UA2’ generates the appropriate INVITE request based on the ‘Refer-To’ header. As the ‘Refer-To’ header was constructed based on (3) by ‘UA1’, the SIP URI points to ‘B2BUA2’ and the INVITE dialog parameters represent dialog (3). This results in ‘UA2’ generating a SIP INVITE request that is sent to ‘B2BUA2’ replacing (3). This is illustrated by (5).

A number of SIP architectures, for example the IP Multimedia System (IMS), have based application composition when servicing a user on a concept known as sequencing. Sequencing involves an intelligent entity making recursive decisions on behalf of users as to which application a SIP request should be routed to next. A simple example is illustrated in Figure 3.
In Figure 3, firstly a SIP request arrives at the entity carrying out sequencing, as shown by (1). The Sequencing engines makes the decision to route the request to 'App1' as shown by (2). The pushing of two SIP Route headers (one pointing to 'App1' and the other pointing back to itself) is a common way for a sequencing engine to ensure the request is returned. Once 'App1' has finished with the request it is returned back the sequencing engine for further processing (3). (4) The request is then sequenced to 'App2' and then returned back to the sequencing engine (5). This is then repeated for the final application to be sequenced (6)(7). Once all appropriate applications have been traversed, the SIP request is routed onwards (8).

The combination of using sequencing in conjunction with B2BUA causes a number of problems. Some out of dialog SIP requests, headers and parameters are used to directly reference an existing dialog to avoid sharing SIP dialog state (as discussed in RFC 5057 [RFC5057]). The use of GRUU[I-D.ietf-sip-gruu], SIP Join[RFC3911] and Replaces[RFC3891] headers are good examples where specific dialog information is used within a SIP message to reference an existing dialog.

In the simple case, if you have a B2BUA that is aware of translating all appropriate SIP headers then signalling will traverse appropriately. You also need to ensure that the request traverses a B2BUA instance that has appropriate information to achieve the correct outcome. A problem occurs when you are attempting to apply the sequencing model previously discussed. All SIP signalling requests arriving at the sequencing engine have to be treated independently and sequenced appropriately. The applications selected would most probably result in a different application path being taken. In fact, it would be seen as a major constraint for a path to be forced through certain applications to obtain consistent service. For this reason, requests that are targeted using information such as
the SIP Join and Replaces headers need to maintain a consistent, end
to end association. From the earlier example illustrated in
Figure 3, an endpoints view of the call is represented in Figure 4.
It should also be noted that even when such sequencing techniques are
not being used, a SIP request as specified in Figure 2 by (3), is not
guaranteed to be sent to the same B2BUA instance. The mapping
carried out in (4) from Figure 2 would not take place.

```
(1)      +----+   +----+   +----+     (2)
UA1 <----------> |App1|<->|App2|<->|App3|<----------> UA2
+----+   +----+   +----+
```

Figure 4: Endpoint View

Figure 4 illustrates that if App1 and App3 are B2BUAs then the dialog
identifiers will represent the SIP dialog relationship between the
User Agent and the nearest B2BUA (not end to end). This is shown by
(1) and (2) in Figure 4. As a result, if either User Agent wishes to
create a new request that in some way references the existing dialog
(for example the SIP Join/Replaces header) or wants to direct the
request at a specific User Agent instance (by using the contact
address or GRUU), it is only able to populate the new request with
local SIP protocol information (the relationship illustrated by (1)
and (2) between the client and B2BUA). If, for example, one of the
clients wanted to create an out of dialog REFER that included a SIP
Replaces[RFC3891] header parameter in the Refer-To header, it would
only be able to populate based on the local identifiers between the
client and the B2BUA. It is also true that for the SIP R-URI to be
an accurate identifier it also must be populated correctly using the
contact generated by the client and not that by the B2BUA.

This specification proposes a solution to allow an endpoints view of
the call to be provided end-to-end for the purpose of being used in
new, related SIP protocol requests.

1.2. Conventions and Terminology

In this document, BCP 14/RFC 2119 [RFC2119] defines the key words
"MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD",
"SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and
"OPTIONAL". In addition, BCP 15 indicates requirement levels for
compliant implementations.
2. Requirements

- REQ-01 - A SIP User Agent Client is able to request support for a local view of a dialog.

- REQ-02 - A SIP User Agent Server is able to convey support for the local view mechanism.

- REQ-03 - A SIP User Agent Client is able to populate a SIP message appropriately conveying its local view.

- REQ-04 - A SIP User Agent Server is able to populate a SIP message appropriately conveying its local view.

- REQ-05 - Local view information should include the dialog identifiers as specified in [RFC3261].

- REQ-06 - Local view information should include a target for subsequent out of dialog requests. This could either be the local contact address as specified in [RFC3261] or the appropriate GRUU[I-D.ietf-sip-gruu].

- REQ-07 - Appropriate guidelines for support from SIP B2BUAs should be documented to ensure compliance to this extension.

- REQ-08 - The entities receiving the endpoint view information conveyed by this extension must be able to authenticate the entity providing the request.

- REQ-09 - Appropriate security and confidentiality is required for local view mechanism.
3. UAC Behavior - INVITE generation

The procedures defined in this extension SHOULD only be used in a SIP INVITE request as specified in [RFC3261]. Future standardisation mechanisms MAY allow the header to be included in other SIP dialog creating requests. The client MUST generate a standard SIP INVITE request as specified in [RFC3261]. A client wishing to demonstrate support for this extension MUST insert a SIP 'Supported' header as specified in [RFC3261] with the value of 'Endpoint-View'. A client wishing to insist on using this extension MUST insert a SIP 'Require' header as specified in [RFC3261] with the value of 'Endpoint-View'. The client MUST also insert the 'Endpoint-View' SIP header as specified in Section 9. The header will be populated with appropriate information representing the local-tag, call-id and contact address of the User Agent (which could be a GRUU[I-D.ietf-sip-gruu]). At this stage the generating client does not know the value of the remote tag and so it is left out.
4. UAS Behavior

On receiving an INVITE request, the UAS should inspect the message to identify if this extension is supported. This would be reflected by the presence of either a SIP Require or Supported header containing the 'Endpoint-View' token. If the SIP supported header is present then the UAS can optionally decide to include a SIP 'Endpoint-View' header in the corresponding response. If the SIP Require header is present then the UAS MUST support the extension to successfully process the INVITE request. If it does not support this extension it will generate a SIP 420 Bad Extension response as per [RFC3261].

If the UAS decides to indicate support for this draft, either through the presence of a SIP Require/Supported header, a SIP Endpoint-View SIP header MUST inserted as specified in Section 9 in a reliable SIP response. [RFC3261] only defines 2xx responses as reliable while [RFC3262] can also be used to convey this information. The header will be populated with appropriate information representing the local-tag, remote-tag, call-id and contact address of the User Agent (which could be a GRUU). The response is then sent as normal.
5. Reliable Provisional

It should be noted that while the ‘Endpoint-View’ header can be included in reliable provisional responses, the values included in a SIP PRACK/200 OK exchange MUST not contradict the values included in the SIP INVITE request and associated reliable response. In short, the UAC MUST include the ‘Endpoint-View’ header in a PRACK request (when reliable provisional responses [RFC3262] are being used) which will be identical to the values included in INVITE request with the addition of the ‘remote-tag’ header field parameter. This allows the UAS to gain an earlier view of the full dialog details before waiting for the SIP ACK request. The header will be the same as that for the corresponding SIP ACK request. The ‘Endpoint-View’ header MUST NOT be included in the 200 OK to a PRACK as it has no meaning when compared to the value returned in the SIP provisional response.

[Editors Note: Currently we call out that 200 OK to PRACK MUST NOT include this header. It has no meaning and so its exclusion does not break anything. Should we be less explicit?]
6. UAC Behavior - ACK generation

On generating the corresponding SIP ACK request for a successful SIP 200 OK response as per [RFC3261], to complete the INVITE transaction, the SIP UAC will also include the SIP Endpoint-View header representing the local view of the dialog. The header will be populated with appropriate information representing the local-tag, remote-tag, call-id and contact address representing the User Agent (which could be a GRUU). The ACK is then sent as normal.
7. B2BUA Behavior

A B2BUA has no clear definition and is able to manipulate SIP protocol messages in any number of ways. This specification can in no way enforce B2BUA behavior but can provide guidelines. For this extension to work successfully a B2BUA should not remove or manipulate the SIP ‘Endpoint-View’ header when it traverses. For this specification to work the B2BUA is also advised to maintain appropriate SIP Supported and Require headers when they appear in the signalling. If supporting this extension contravenes any form of local policy or security, the B2BUA should act as elegantly as possible (for example, if a SIP Require header is present it should act as a UAS and generate a SIP 420 Bad Extension response).
8. Security and Privacy

[Editors Note: To be included in a later version of the extension. ‘Endpoint-View’ header information needs to be appropriately protected using encryption etc.]
9. The Endpoint-View Header

This specification defines a new SIP protocol header: 'Endpoint-View'. Its formatting as a SIP header is described by the following ABNF[RFC5234] and based on standard SIP syntax.

```
Endpoint-View = "Endpoint-View" HCOLON contact-param *(SEMI
              endpoint-param) ;contact-param from RFC
              3261 and optionally gruu extension
              from RFCXXX
endpoint-param = remote-param / local-param / call-id
              generic-param
remote-param = "remote-tag" EQUAL token
local-param = "local-tag" EQUAL token
call-id = "call-id" EQUAL token
          ;remote-param and local-param from RFC 4538
          ;token and generic-param from RFC 3261
```

Table 1 and 2 are an extension of Tables 2 and 3 in RFC 3261[RFC3261] for the Endpoint-View header field. The column "INF" is for the INFO method [RFC2976], "PRA" is for the PRACK method [RFC3262], "UPD" is for the UPDATE method [RFC3311], "SUB" is for the SUBSCRIBE method [RFC3265], "NOT" is for the NOTIFY method [RFC3265], "MSG" is for the MESSAGE method [RFC3428], "REF" is for the REFER method [RFC3515], and "PUB" is for the PUBLISH method [RFC3903].

<table>
<thead>
<tr>
<th>Header field</th>
<th>where proxy ACK BYE CAN INV OPT REG PUB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endpoint-View</td>
<td>R - o - - o - - - - - - - -</td>
</tr>
<tr>
<td>Endpoint-View</td>
<td>2xx,18x - - - - o - - - - -</td>
</tr>
</tbody>
</table>

Figure 5: Table 1

<table>
<thead>
<tr>
<th>Header field</th>
<th>where proxy PRA UPD SUB NOT INF MSG REF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endpoint-View</td>
<td>R - o - - - - - - - - - - - - - -</td>
</tr>
<tr>
<td>Endpoint-View</td>
<td>2xx,18x - - - - - - - - - - - - - -</td>
</tr>
</tbody>
</table>

Figure 6: Table 2
10. Example

The following is an example of a simple exchange using the 'Endpoint-View' SIP header. Some of the messages have been left out for simplicity.

```
(1) UAC->B2BUA (SIP): INVITE requiring support of the ‘Endpoint-View’ extension.

INVITE sip:uas@example.com SIP/2.0
To: <sip:uas@example.com>
From: <sip:uac@example.com>;tag=8937498
Via: SIP/2.0/UDP uac.example.com;branch=z9hG412345678
CSeq: 1 INVITE
Call-ID: 893jhoeihjr8392@example.com
Require: Endpoint-View
Endpoint-View: <sip:uac@pc1.example.com;gr=kowpeojr893274jksd>
                ;local-tag=8937498
                ;call-id=893jhoeihjr8392@example.com
Contact: <sip:uac@pc1.example.com;gr=kowpeojr893274jksd>
Content-Type: application/sdp
Content-Length: [..]

[SDP NOT INCLUDED]

(2) B2BUA->UAS (SIP): INVITE requiring support of the ‘Endpoint-View’ extension with changed dialog parameters and contact address.
```
INVITE sip:uas@example.com SIP/2.0
To: <sip:uas@example.com>
From: <sip:b2bua@example.com>;tag=438290jdJ239hd
Via: SIP/2.0/UDP b2bua.example.com;branch=z9hG4892374892
CSeq: 1 INVITE
Call-ID: djioHJKd38972yjdf1342@example.com
Require: Endpoint-View
Endpoint-View: <sip:b2bua@example.com;gr=kowpeojr893274jkdsd>
  ;local-tag=8937498
  ;call-id=893jhoeihjr83928example.com
Contact: <sip:b2bua@example.com>
Content-Type: application/sdp
Content-Length: [..]

[SDP NOT INCLUDED]

(3) UAS->B2BUA (SIP): 200 OK

SIP/2.0 200 OK
To: <sip:uas@example.com>;tag=023983774
From: <sip:uac@example.com>;tag=438290jdJ239hd
Via: SIP/2.0/UDP b2bua.example.com;branch=z9hG4892374892
CSeq: 1 INVITE
Call-ID: djioHJKd38972yjdf1342@example.com
Supported: Endpoint-View
Endpoint-View: <sip:uas@pc2.example.com;gr=230plksdajf93824j>
  ;local-tag=023983774
  ;remote-tag=438290jdJ239hd
  ;call-id=djioHJKd38972yjdf1342@example.com
Contact: <uas@pc2.example.com;gr=230plksdajf93824j>
Content-Type: application/sdp
Content-Length: [..]

[SDP NOT INCLUDED]

(4) B2BUA->UAC (SIP): 200 OK
SIP/2.0 200 OK
To: <sip:uas@example.com>;tag=8jc8923jdl
From: <sip:uac@example.com>;tag=8937498
Via: SIP/2.0/UDP b2bua.example.com;branch=z9hG412345678
CSeq: 1 INVITE
Call-ID: 893jhoeihjr8392@example.com
Supported: Endpoint-View
Endpoint-View: <sip:uas@pc2.example.com;gr=230plksdajf93824j>
   ;local-tag=023983774
   ;remote-tag=438290jdJ239hd
   ;call-id=djioHJKd38972yjdf1342@example.com
Contact: <sip:b2bua@example.com;gr=39uadsj8239ujdj0>
Content-Type: application/sdp
Content-Length: [..]

[SDP NOT INCLUDED]

(5) UAC->B2BUA (SIP): ACK

ACK sip:b2bua@example.com;gr=39uadsj8239ujdj0 SIP/2.0
To: <sip:uas@example.com>;tag=8jc8923jdl
From: <sip:uac@example.com>;tag=8937498
Via: SIP/2.0/UDP uac.example.com;branch=z9hG429dj80321je
CSeq: 1 ACK
Call-ID: 893jhoeihjr8392@example.com
Require: Endpoint-View
Endpoint-View: <sip:uac@pc1.example.com;gr=kowpeojr893274jksd>
   ;local-tag=8937498
   ;remote-tag=8jc8923jdl
   ;call-id=893jhoeihjr8392@example.com

(6) B2BUA->UAS (SIP): ACK

ACK sip:uas@pc2.example.com;gr=230plksdajf93824j SIP/2.0
To: <sip:uas@example.com>;tag=023983774
From: <sip:uac@example.com>;tag=438290jdJ239hd
Via: SIP/2.0/UDP b2bua.example.com;branch=z9hG43920dj2io3jd98
CSeq: 1 ACK
Call-ID: djioHJKd38972yjdf1342@example.com
Require: Endpoint-View
Endpoint-View: <sip:uac@pc1.example.com;gr=kowpeojr893274jksd>
   ;local-tag=8937498
   ;remote-tag=8jc8923jdl
   ;call-id=893jhoeihjr8392@example.com
11. Security Considerations

Security Considerations to be included in later versions of this document.
12. IANA Considerations

This specification registers a new SIP header field, a new option tag according to the processes of RFC 3261 [RFC3261].

12.1. Header Field

RFC Number: RFC XXXX

Header Field Name: Endpoint-View

Compact Form: none

12.2. SIP Option Tag

This specification registers a new SIP option tag per the guidelines in Section 27.1 of RFC 3261 [RFC3261].

Name: Endpoint-View

Description: This option tag is used to identify the Endpoint-View header field extension. When used in a Require header field, it implies that the recipient needs to support the Endpoint-View header field. When used in a Supported header field, it implies that the sender of the message supports it.
13. Acknowledgments

The authors would like to thank Gordon Brunson, Harsh Mendiratta and Joel Ezall who contributed significantly to the proposal.
14. References

14.1. Normative References


14.2. Informative References

[I-D.ietf-sip-gruu]


Authors’ Addresses

Chris Boulton
NS-Technologies
Email: chris@ns-technologies.com

Ian Evans
Avaya
Building 3
Wern Fawr Lane
St Mellons
Cardiff, South Wales  CF3 5EA
Email: ievansATavaya.com

Gethin Liddell
Avaya
Building 3
Wern Fawr Lane
St Mellons
Cardiff, South Wales  CF3 5EA
Email: gliddellATavaya.com

David Shutt
Avaya
Building 3
Wern Fawr Lane
St Mellons
Cardiff, South Wales  CF3 5EA
Email: dshuttATavaya.com

Pete Barrett
Avaya
Building 3
Wern Fawr Lane
St Mellons
Cardiff, South Wales  CF3 5EA
Email: pbarrettATavaya.com