Network Working Group   P. Saint-Andre
Internet-Draft   Cisco Systems, Inc.
Intended status: Standards Track   A. Houri
Expires: January 2, 2014   IBM
                                                       J. Hildebrand
                                                  Cisco Systems, Inc.
                                                        July 1, 2013

        **Interworking between the Session Initiation Protocol (SIP) and the
      Extensible Messaging and Presence Protocol (XMPP): Addresses and Error
                                 Conditions**
                          draft-ietf-stox-core-00

Abstract

   As a foundation for the definition of bidirectional protocol mappings
   between the Session Initiation Protocol (SIP) and the Extensible
   Messaging and Presence Protocol (XMPP), this document specifies the
   architectural assumptions underlying such mappings as well as the
   mapping of addresses and error conditions.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

The IETF has worked on two signalling technologies that can be used
for multimedia session negotiation, messaging, presence, capabilities
discovery, notifications, and other application-level functionality:

o   The Session Initiation Protocol [RFC3261], along with various SIP
    extensions developed within the SIP for Instant Messaging and
    Presence Leveraging Extensions (SIMPLE) Working Group.
o   The Extensible Messaging and Presence Protocol [RFC6120], along
    with various XMPP extensions developed by the IETF as well as by
    the XMPP Standards Foundation.

Because these technologies are widely deployed, it is important to
clearly define mappings between them for the sake of interworking.
This document inaugurates a series of SIP-XMPP interworking
specifications by defining the architectural assumptions underlying
such mappings as well as the mapping of addresses and error
conditions.

The discussion venue for this document is the mailing list of the
STOX WG; visit https://www.ietf.org/mailman/listinfo/stox for
subscription information and discussion archives.

2. Terminology

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].

3. Architectural Assumptions

Protocol translation between SIP and XMPP could occur in a number of
different entities, depending on the architecture of real-time
communication deployments. For example, protocol translation could
occur within a multi-protocol server (which uses application-specific
connection managers to initiate traffic to and accept traffic from
clients or other servers natively using SIP/SIMPLE, XMPP, etc.),
within a multi-protocol client (which enables a user to establish
connections natively with various servers using SIP/SIMPLE, XMPP,
etc.), or within a gateway that acts as a dedicated protocol
translator (which takes one protocol as input and provides another
protocol as output).

This document assumes that the protocol translation will occur within

a gateway. (This assumption not meant to discourage protocol
translation within multi-protocol clients or servers; instead, this
assumption is followed mainly to clarify the discussion and examples
so that the protocol translation principles can be more easily
understood and can be applied by client and server implementors with
appropriate modifications to the examples and terminology.)
Specifically, we assume that the protocol translation will occur
within an "XMPP-to-SIP gateway" that translates XMPP syntax and
semantics on behalf of an XMPP service when communicating with SIP
services and/or within a "SIP-to-XMPP gateway" that translates SIP
syntax and semantics on behalf of a SIP service when communicating
with XMPP services (naturally, these logical functions could occur in
one and the same actual translator).

This document assumes that a gateway will translate directly from one
protocol to the other. For the sake of the examples, we further
assume that protocol translation will occur within a gateway in the
source domain, so that information generated by the user of an XMPP
service will be translated by a gateway within the trust domain of
that XMPP service, and information generated by the user of a SIP
service will be translated by a gateway within the trust domain of
that SIP service. However, nothing in this document ought to be
taken as recommending against protocol translation at the destination
domain.

An architectural diagram for a possible gateway deployment is shown
below, where the entities have the following significance and the "#"
character is used to show the boundary of a trust domain:

o romeo@example.net -- a SIP user.
o example.net -- a SIP service with a gateway ("GW") to XMPP.
o juliet@example.com -- an XMPP user.
o example.com -- an XMPP service with a gateway ("GW") to SIP.

```
   ##@#########################################################
   #                             #                            #
   #    +-------------+----+   #   +----+-------------+    #
   #    | example.net | GW |---#---| GW | example.com |    #
   #    +-------------+----+   #   +----+-------------+    #
   #          |                  #            |            #
   #     romeo@example.net       #       juliet@example.com  #
   #                             #                            #
   ###########################################################
```
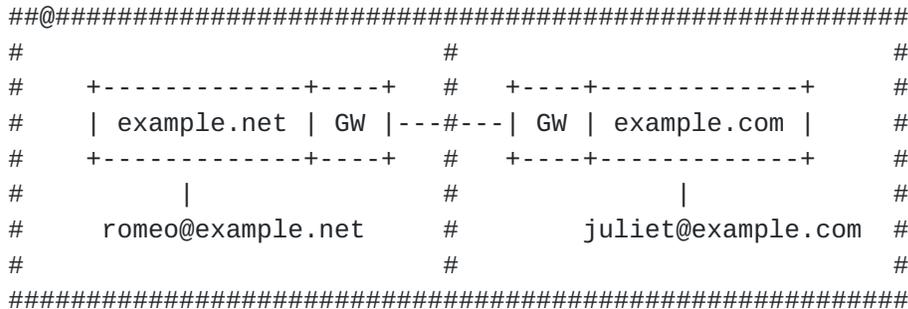
4. Address Mapping

4.1. Overview

The basic SIP address format is a "sip:" or "sips:" URI as specified
in [RFC3261]. When a SIP entity supports extensions for instant
messaging it might be identified by an 'im:' URI as specified in the
Common Profile for Instant Messaging [RFC3860] (see [RFC3428]) and
when a SIP entity spports extensions for presence it might be
identified by a 'pres:' URI as specified in the Common Profile for
Presence [RFC3859] (see [RFC3856]).

The XMPP address format is specified in [RFC6122]; as discussed in
[RFC6121], instant messaging and presence applications of XMPP also
need to support 'im:' and 'pres:' URIs as specified in [RFC3860] and
[RFC3859] respectively, although such support might simply involve
leaving resolution of such addresses up to an XMPP server.

In this document we primarily describe mappings for addresses of the
form <user@domain>; however, we also provide guidelines for mapping
the addresses of specific user agent instances, which take the form
of Globally Routable User Agent URIs (GRUUs) in SIP and
"resourceparts" in XMPP. Mapping of protocol-specific identifiers
(such as telephone numbers) is out of scope for this specification.
In addition, we have ruled the mapping of domain names as out of
scope for now since that is a matter for the Domain Name System;
specifically, the issue for interworking between SIP and XMPP relates
to the translation of fully internationalized domain names (IDNs)
into non-internationalized domain names (IDNs are not allowed in the
SIP address format, but are allowed in the XMPP address via
Internationalized Domain Names in Applications, see [RFC6122] and
[I-D.ietf-xmpp-6122bis]). Therefore, in the following sections we
focus primarily on the local part of an address (these are called
variously "usernames", "instant inboxes", "presentities", and
"localparts" in the protocols at issue), secondarily on the instance-
specific part of an address, and not at all on the domain-name part
of an address.

The sip:/sips:, im:/pres:, and XMPP address schemes allow different
sets of characters (although all three allow alphanumeric characters
and disallow both spaces and control characters). In some cases,
characters allowed in one scheme are disallowed in others; these
characters need to be mapped appropriately in order to ensure
interworking across systems.

4.2. **Local Part Mapping**

   The local part of a sip:/sips: URI inherits from the "userinfo" rule
   in [RFC3986] with several changes; here we discuss the SIP "user"
   rule only:

```
   user             =  1*( unreserved / escaped / user-unreserved )
   user-unreserved  =  "&" / "=" / "+" / "$" / "," / ";" / "?" / "/"
   unreserved       =  alphanum / mark
   mark             =  "-" / "_" / "." / "!" / "~" / "*" / "'"
                       / "(" / ")"
```

   Here we make the simplifying assumption that the local part of an
   im:/pres: URI inherits from the "dot-atom-text" rule in [RFC5322]
   rather than the more complicated "local-part" rule:

```
   dot-atom-text =  1*atext *("." 1*atext)
   atext         =  ALPHA / DIGIT /    ; Any character except
                    "!" / "#" / "$" /  ; controls, SP, and
                    "%" / "&" / "'" /  ; specials. Used for
                    "*" / "+" / "-" /  ; atoms.
                    "/" / "=" / "?" /
                    "^" / "_" / "`" /
                    "{" / "|" / "}" /
                    "~"
```

   The local part of an XMPP address allows any ASCII character except
   space, controls, and the " & ' / : < > @ characters.

   To summarize the foregoing information, the following table lists the
   allowed and disallowed characters in the local part of identifiers
   for each protocol (aside from the alphanumeric, space, and control
   characters), in order by hexadecimal character number (where each "A"
   row shows the allowed characters and each "D" row shows the
   disallowed characters).

Table 1: Allowed and disallowed characters

```
+---+--------------------------------+
| SIP/SIPS CHARACTERS                |
+---+--------------------------------+
| A | !  $ &'()*+,-./ ; = ?      _    ~ |
| D |   "# %             : < > @[\]^ `{|}  |
+---+--------------------------------+
| IM/PRES CHARACTERS                 |
+---+--------------------------------+
| A | ! #$%&'  *+ - /    = ?    ^_`{|}~ |
| D |   "      ()  , . :;< > @[\]       |
+---+--------------------------------+
| XMPP CHARACTERS                    |
+---+--------------------------------+
| A | ! #$%  ()*+,-.  ; = ? [\]^_`{|}~ |
| D |   "   &'        /: < > @         |
+---+--------------------------------+
```

When transforming the local part of an address from one scheme to another, an application SHOULD proceed as follows:

1. Unescape any escaped characters in the source address (e.g., from
    SIP to XMPP unescape "%2F" to "/" and from XMPP to SIP unescape
    "\27" to "'").
2. Leave unmodified any characters that are allowed in the
    destination scheme.
3. Escape any characters that are allowed in the source scheme but
    reserved in the destination scheme, as escaping is defined for
    the destination scheme. In particular:
    *   Where the destination scheme is a URI (i.e., an im:, pres:,
        sip:, or sips: URI), each reserved character MUST be percent-
        encoded to "%hexhex" as specified in Section 2.6 of [RFC4395]
        (e.g., when transforming from XMPP to SIP, encode "/" as
        "%2F").
    *   Where the destination scheme is a native XMPP address, each
        reserved character MUST be encoded to "\hexhex" as specified
        in [XEP-0106] (e.g., when transforming from SIP to XMPP,
        encode "'" as "\27").

4.3. Instance-Specific Mapping

The meaning of a resourcepart in XMPP (i.e., the portion of a JID
after the slash character, such as "foo" in "user@example.com/foo")
matches that of a Globally Routable User Agent URI (GRUU) in SIP
[RFC5627]. In both cases, these constructs identify a particular
device associated with the bare JID ("localpart@domainpart") of an
XMPP entity or with the Address of Record (AOR) of a SIP entity.

Therefore, it is reasonable to map the value of a "gr" URI parameter
to an XMPP resource part, and vice-versa.

Note that the "gr" URI parameter in SIP can contain only characters
from the ASCII range, whereas an XMPP resourcepart can contain nearly
any Unicode character [UNICODE]. Therefore Unicode characters
outside the ASCII range need to be mapped to characters in the ASCII
range, as described below.

4.4. SIP to XMPP

The following is a high-level algorithm for mapping a sip:, sips:,
im:, or pres: URI to an XMPP address:

1. Remove URI scheme.
2. Split at the first '@' character into local part and hostname
    (mapping the latter is out of scope).
3. Translate any percent-encoded strings ("%hexhex") to percent-
    decoded octets.
4. Treat result as a UTF-8 string.
5. Translate "&" to "\26", "'" to "\27", and "/" to "\2f"
    respectively in order to properly handle the characters
    disallowed in XMPP addresses but allowed in sip:/sips: URIs and
    im:/pres: URIs as shown in Table 1 above (this is consistent with
    [XEP-0106]).
6. Apply Nodeprep profile of Stringprep [RFC3454] or its replacement
    (see [RFC6122] and [I-D.ietf-xmpp-6122bis]) for canonicalization
    (OPTIONAL).
7. Recombine local part with mapped hostname to form a bare JID
    ("localpart@domainpart").
8. If the (SIP) address contained a "gr" URI parameter, append a
    slash character "/" and the "gr" value to the bare JID to form a
    full JID ("localpart@domainpart/resourcepart").

4.5. XMPP to SIP

The following is a high-level algorithm for mapping an XMPP address
to a sip:, sips:, im:, or pres: URI:

1. Split XMPP address into localpart (mapping described in remaining
    steps), domainpart (hostname; mapping is out of scope), and
    resourcepart (specifier for particular device or connection, for
    which an OPTIONAL mapping is described below).
2. Apply Nodeprep profile of [RFC3454] or its replacement (see
    [RFC6122] and [I-D.ietf-xmpp-6122bis]) for canonicalization of
    the XMPP localpart (OPTIONAL).

3. Translate "\26" to "&", "\27" to "'", and "\2f" to "/"
    respectively (this is consistent with [XEP-0106]).
4. Determine if the foreign domain supports im: and pres: URIs
    (discovered via [RFC2782] lookup as specified in [RFC6121]), else
    assume that the foreign domain supports sip:/sips: URIs.
5. If converting into im: or pres: URI, for each byte, if the byte
    is in the set (),.;[\] or is a UTF-8 character outside the ASCII
    range then percent-encode that byte to "%hexhex" format. If
    converting into sip: or sips: URI, for each byte, if the byte is
    in the set #%[\]^`{|} or is a UTF-8 character outside the ASCII
    range then percent-encode that byte to "%hexhex" format.
6. Combine resulting local part with mapped hostname to form
    local@domain address.
7. Prepend with 'im:' scheme (for XMPP <message/> stanzas) or
    'pres:' scheme (for XMPP <presence/> stanzas) if foreign domain
    supports these, else prepend with 'sip:' or 'sips:' scheme
    according to local service policy.
8. If the XMPP address included a resourcepart and the destination
    URI scheme is 'sip:' or 'sips:', optionally append the slash
    character '/' and then append the resourcepart (making sure to
    percent-encode any UTF-8 characters outside the ASCII range) as
    the "gr" URI parameter.

5. Error Condition Mapping

   SIP response codes are specified in [RFC3261] and XMPP error
   conditions are specified in [RFC6120]. Because there is no
   equivalent in XMPP for the provisional (1xx) and successful (2xx)
   response codes in SIP, mappings are provided only for the redirection
   (3xx), request failure (4xx), server failure (5xx), and glogal
   failure (6xx) codes.

5.1. XMPP to SIP

   Table 8: Mapping of XMPP error conditions to SIP response codes

```
   +------------------------------+---------------------+
   |  XMPP Error Condition        |  SIP Response Code  |
   +------------------------------+---------------------+
   |  <bad-request/>              |  400                |
   |  <conflict/>                 |  400                |
   |  <feature-not-implemented/>  |  501                |
   |  <forbidden/>                |  403                |
   |  <gone/>                     |  410                |
   |  <internal-server-error/>    |  500                |
   |  <item-not-found/>           |  404                |
   |  <jid-malformed/>            |  484                |
   |  <not-acceptable/>           |  406                |
   |  <not-allowed/>              |  405                |
   |  <not-authorized/>           |  401                |
   |  <recipient-unavailable/>    |  480                |
   |  <redirect/>                 |  300                |
   |  <registration-required/>    |  407                |
   |  <remote-server-not-found/>  |  502                |
   |  <remote-server-timeout/>    |  504                |
   |  <resource-constraint/>      |  500                |
   |  <service-unavailable/>      |  503                |
   |  <subscription-required/>    |  407                |
   |  <undefined-condition/>      |  400                |
   |  <unexpected-request/>       |  491                |
   +------------------------------+---------------------+
```

5.2. SIP to XMPP

   The mapping of SIP response codes to XMPP error conditions SHOULD be
   as follows (note that XMPP does not include 100-series or 200-series
   response codes, only error conditions):

   Table 9: Mapping of SIP response codes to XMPP error conditions

```
+---------------------+------------------------------+
|  SIP Response Code  |  XMPP Error Condition        |
+---------------------+------------------------------+
|  300                |  <redirect/>                 |
|  301                |  <gone/>                     |
|  302                |  <redirect/>                 |
|  305                |  <redirect/>                 |
|  380                |  <not-acceptable/>           |
|  400                |  <bad-request/>              |
|  401                |  <not-authorized/>           |
|  402                |  see note [1]                |
|  403                |  <forbidden/>                |
|  404                |  <item-not-found/>           |
|  405                |  <not-allowed/>              |
|  406                |  <not-acceptable/>           |
|  407                |  <registration-required/>    |
|  408                |  <recipient-unavailable/>    |
|  410                |  <gone/>                     |
|  413                |  <bad-request/>              |
|  414                |  <bad-request/>              |
|  415                |  <bad-request/>              |
|  416                |  <bad-request/>              |
|  420                |  <bad-request/>              |
|  421                |  <bad-request/>              |
|  423                |  <bad-request/>              |
|  480                |  <recipient-unavailable/>    |
|  481                |  <item-not-found/>           |
|  482                |  <not-acceptable/>           |
|  483                |  <not-acceptable/>           |
|  484                |  <jid-malformed/>            |
|  485                |  <item-not-found/>           |
|  486                |  <recipient-unavailable/>    |
|  487                |  <recipient-unavailable/>    |
|  488                |  <not-acceptable/>           |
|  491                |  <unexpected-request/>       |
|  493                |  <bad-request/>              |
|  500                |  <internal-server-error/>    |
|  501                |  <feature-not-implemented/>  |
|  502                |  <remote-server-not-found/>  |
|  503                |  <service-unavailable/>      |
|  504                |  <remote-server-timeout/>    |
|  505                |  <not-acceptable/>           |
|  513                |  <bad-request/>              |
|  600                |  <recipient-unavailable/>    |
|  603                |  <recipient-unavailable/>    |
|  604                |  <item-not-found/>           |
|  606                |  <not-acceptable/>           |
+---------------------+------------------------------+
```

   1. The XMPP <payment-required/> error condition was removed in
       [RFC6120].

6. Security Considerations

   Detailed security considerations for SIP are given in [RFC3261] and
   for XMPP in [RFC6120].

7. IANA Considerations

   This document requests no actions of IANA.

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.

   [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.

   [RFC3986]   Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform
               Resource Identifier (URI): Generic Syntax", STD 66,
               RFC 3986, January 2005.

   [RFC4395]   Hansen, T., Hardie, T., and L. Masinter, "Guidelines and
               Registration Procedures for New URI Schemes", RFC 4395,
               February 2006.

   [RFC5627]   Rosenberg, J., "Obtaining and Using Globally Routable User
               Agent URIs (GRUUs) in the Session Initiation Protocol
               (SIP)", RFC 5627, October 2009.

   [RFC6120]   Saint-Andre, P., "Extensible Messaging and Presence
               Protocol (XMPP): Core", RFC 6120, March 2011.

   [RFC6122]   Saint-Andre, P., "Extensible Messaging and Presence
               Protocol (XMPP): Address Format", RFC 6122, March 2011.

   [UNICODE]   The Unicode Consortium, "The Unicode Standard, Version
               6.2", 2012,
               <http://www.unicode.org/versions/Unicode6.2.0/>.

8.2. Informative References

   [I-D.ietf-xmpp-6122bis]
              Saint-Andre, P., "Extensible Messaging and Presence
              Protocol (XMPP): Address Format",
              draft-ietf-xmpp-6122bis-07 (work in progress), April 2013.

   [RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for
              specifying the location of services (DNS SRV)", RFC 2782,
              February 2000.

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

   [RFC3454] Hoffman, P. and M. Blanchet, "Preparation of
              Internationalized Strings ("STRINGPREP")", RFC 3454,
              December 2002.

   [RFC3856] Rosenberg, J., "A Presence Event Package for the Session
              Initiation Protocol (SIP)", RFC 3856, August 2004.

   [RFC3859] Peterson, J., "Common Profile for Presence (CPP)",
              RFC 3859, August 2004.

   [RFC3860] Peterson, J., "Common Profile for Instant Messaging
              (CPIM)", RFC 3860, August 2004.

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

   [RFC6121] Saint-Andre, P., "Extensible Messaging and Presence
              Protocol (XMPP): Instant Messaging and Presence",
              RFC 6121, March 2011.

   [XEP-0106]
              Saint-Andre, P. and J. Hildebrand, "JID Escaping", XSF
              XEP 0106, May 2005.

Appendix A. Acknowledgements

Authors' Addresses

    Peter Saint-Andre
    Cisco Systems, Inc.
    1899 Wynkoop Street, Suite 600
    Denver, CO 80202
    USA

    Phone: +1-303-308-3282
    Email: psaintan@cisco.com

    Avshalom Houri
    IBM
    Building 18/D, Kiryat Weizmann Science Park
    Rehovot 76123
    Israel

    Email: avshalom@il.ibm.com

    Joe Hildebrand
    Cisco Systems, Inc.
    1899 Wynkoop Street, Suite 600
    Denver, CO 80202
    USA

    Email: jhildebr@cisco.com