IETF
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2017

N. Tomkinson
N. Borenstein
Mimecast Ltd

July 8, 2016

Multiple Language Content Type
draft-ietf-slim-multilangcontent-01

Abstract

This document defines an addition to the Multipurpose Internet Mail Extensions (MIME) standard to make it possible to send one message that contains multiple language versions of the same information. The translations would be identified by a language tag and selected by the email client based on a user's language settings.

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 9, 2017.

Copyright Notice

1. Introduction

Since the invention of email and the rapid spread of the Internet, more and more people have been able to communicate in more and more countries and in more and more languages. But during this time of technological evolution, email has remained a single-language communication tool, whether it is English to English, Spanish to Spanish or Japanese to Japanese.

Also during this time, many corporations have established their offices in multi-cultural cities and formed departments and teams that span continents, cultures and languages, so the need to communicate efficiently with little margin for miscommunication has grown exponentially.

The objective of this document is to define an addition to the Multipurpose Internet Mail Extensions (MIME) standard, to make it possible to send a single message to a group of people in such a way that all of the recipients can read the email in their preferred language. The methods of translation of the message content are beyond the scope of this document, but the structure of the email itself is defined herein.

Whilst this document depends on identification of language in message parts for non-real-time communication, there is a companion document that is concerned with a similar problem for real-time communication: [I-D.gellens-slim-negotiating-human-language]

1.1. Requirements Language

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

2. The Content-Type Header Field

The "multipart/multilingual" MIME subtype allows the sending of a message in a number of different languages with the translations embedded in the same message. This MIME subtype helps the receiving email client make sense of the message structure.

The multipart subtype "multipart/multilingual" has similar semantics to "multipart/alternative" (as discussed in RFC 2046 [RFC2046]) in that each of the message parts is an alternative version of the same information. The primary difference between "multipart/multilingual" and "multipart/alternative" is that when using "multipart/ multilingual", the message part to select for rendering is chosen based on the values of the Content-Language field and optionally the

Translation-Type parameter of the Content-Language field instead of the ordering of the parts and the Content-Types.

The syntax for this multipart subtype conforms to the common syntax for subtypes of multipart given in section 5.1.1. of RFC 2046 [RFC2046]. An example "multipart/multilingual" Content-Type header field would look like this:

Content-Type: multipart/multilingual; boundary=01189998819991197253

3. The Message Parts

A multipart/multilingual message will have a number of message parts: exactly one multilingual preface, one or more language message parts and zero or one language independent message part. The details of these are described below.

3.1. The Multilingual Preface

In order for the message to be received and displayed in non-conforming email clients, the message SHOULD contain an explanatory message part which MUST NOT be marked with a Content-Language field and MUST be the first of the message parts. For maximum support in the most basic of non-conforming email clients, it SHOULD have a Content-Type of text/plain. Because non-conforming email clients are expected to treat a message with an unknown multipart type as multipart/mixed (in accordance with sections 5.1.3 and 5.1.7 of RFC 2046 [RFC2046]) they may show all of the message parts sequentially or as attachments. Including and showing this explanatory part will help the message recipient understand the message structure.

This initial message part SHOULD explain briefly to the recipient that the message contains multiple languages and the parts may be rendered sequentially or as attachments. This SHOULD be presented in the same languages that are provided in the subsequent language message parts.

As this explanatory section is likely to contain languages using scripts that require non-US-ASCII characters, it is RECOMMENDED that UTF-8 encoding is used for this message part.

Whilst this section of the message is useful for backward compatibility, it will normally only be shown when rendered by a non-conforming email client, because conforming email clients SHOULD only show the single language message part identified by the user’s preferred language and the language message part’s Content-Language.

For the correct display of the multilingual preface in a non-conforming email client, the sender MAY use the Content-Disposition field with a value of 'inline' in conformance with RFC 2183 [RFC2183] (which defines the Content-Disposition field). If provided, this SHOULD be placed at the multipart/multilingual level and in the multilingual preface. This makes it clear to a non-conforming email client that the multilingual preface should be displayed immediately to the recipient, followed by any subsequent parts marked as 'inline'.

For an example of a multilingual preface, see the examples in Section 8.

3.2. The Language Message Parts

The language message parts are typically translations of the same message content. These message parts SHOULD be ordered so that the first part after the multilingual preface is in the language believed to be the most likely to be recognised by the recipient as this will constitute the default part when language negotiation fails and there is no Language Independent part. All of the language message parts MUST have a Content-Language field and a Content-Type field and MAY have a Translation-Type parameter applied to the Content-Language field.

The Content-Type for each individual language message part SHOULD be message/rfc822 to provide good support with non-conforming email clients. However, an implementation MAY use message/global when support for message/global becomes more commonplace. Each language message part SHOULD have a Subject field in the appropriate language for that language part.

3.3. The Language Independent Message Part

If there is language independent content intended for the recipient to see if they have a preferred language other than one of those specified in the language message parts and the default language message part is unlikely to be understood, another part MAY be provided. This could typically be a language independent graphic. When this part is present, it MUST be the last part, MUST have a Content-Language field with a value of "zxx" (as described in BCP 47/ RFC 5646 [RFC5646]) and SHOULD NOT have a Subject field. The part SHOULD have a Content-Type of message/rfc822 or message/global (to match the language message parts).

4. Message Part Selection

The logic for selecting the message part to render and present to the recipient is summarised in the next few paragraphs.

Firstly, if the email client does not understand multipart/multilingual then it SHOULD treat the message as if it was multipart/mixed and render message parts accordingly.

If the email client does understand multipart/multilingual then it SHOULD ignore the multilingual preface and select the best match for the user's preferred language from the language message parts available. Also, the user may prefer to see the original message content in their second language over a machine translation in their first language. The Translation-Type parameter of the Content-Language field value can be used for further selection based on this preference. The selection of language part may be implemented in a variety of ways, although the matching schemes detailed in RFC 4647 [RFC4647] are RECOMMENDED as a starting point for an implementation. The goal is to render the most appropriate translation for the user.

If there is no match for the user's preferred language (or there is no preferred language information available) the email client SHOULD select the language independent part (if one exists) or the first language part (directly after the multilingual preface) if a language independent part does not exist.

If there is no translation type preference information available, the values of the Translation-Type parameter may be ignored.

Additionally, interactive implementations MAY offer the user a choice from among the available languages.

5. The Content-Language Field

The Content-Language field in the individual language message parts is used to identify the language in which the message part is written. Based on the value of this field, a conforming email client can determine which message part to display (given the user's language settings).

The Content-Language MUST comply with RFC 3282 [RFC3282] (which defines the Content-Language field) and BCP 47/RFC 5646 [RFC5646] (which defines the structure and semantics for the language code values). While RFC 5646 provides a mechanism accommodating increasingly fine-grained distinctions, in the interest of maximum interoperability, each Content-Language value SHOULD be restricted to the largest granularity of language tags; in other words, it is

RECOMMENDED to specify only a Primary-subtag and NOT to include subtags (e.g., for region or dialect) unless the languages might be mutually incomprehensible without them. Examples of this field for English, German and an instruction manual in Spanish and French, could look like the following:

Content-Language: en

Content-Language: de

Content-Language: es, fr

6. The Translation-Type Parameter

The Translation-Type parameter can be applied to the Content-Language field in the individual language message parts and is used to identify the type of translation. Based on the value of this parameter and the user's preferences, a conforming email client can determine which message part to display.

This parameter can have one of three possible values: 'original', 'human' or 'automated' although other values may be added in the future. A value of 'original' is given in the language message part that is in the original language. A value of 'human' is used when a language message part is translated by a human translator or a human has checked and corrected an automated translation. A value of 'automated' is used when a language message part has been translated by an electronic agent without proofreading or subsequent correction.

Examples of this parameter include:

Content-Language: en; translation-type=original

Content-Language: fr; translation-type=human

7. The Subject Field in the Language Message parts

On receipt of the message, conforming email clients will need to render the subject in the correct language for the recipient. To enable this the Subject field SHOULD be provided in each language message part. The value for this field should be a translation of the email subject.

US-ASCII and 'encoded-word' examples of this field include:

Subject: A really simple email subject

```
Subject: =?UTF-8?Q?Un_asunto_de_correo_electr=C3=b3nico_
        realmente_sencillo?=
```

See RFC 2047 [RFC2047] for the specification of 'encoded-word'.

The subject to be presented to the recipient should be selected from the message part identified during the message part selection stage. If no Subject field is found (for example if the language independent part is selected) the top-level Subject header field value should be used.

8. Examples

8.1. An Example of a Simple Multiple language email message

Below is a minimal example of a multiple language email message. It has the multilingual preface and two language message parts.

```
From: Nik
To: Nathaniel
Subject: Example of a message in Spanish and English
Date: Thu, 7 Jul 2016 21:28:00 +0100
MIME-Version: 1.0
Content-type: multipart/multilingual;
        boundary="01189998819991197253"

--01189998819991197253
Content-type: text/plain; charset="UTF-8"
Content-Disposition: inline
Content-transfer-encoding: quoted-printable

This is a message in multiple languages. It says the
same thing in each language. If you can read it in one language,
you can ignore the other translations. The other translations may be
presented as attachments or grouped together.

Este es un mensaje en varios idiomas. Dice lo mismo en
cada idioma. Si puede leerlo en un idioma, puede ignorar las otras
traducciones. Las otras traducciones pueden presentarse como archivos
adjuntos o agrupados.

--01189998819991197253
Content-type: message/rfc822
Content-Language: en; translation-type=original
Content-disposition: inline

Subject: Example of a message in Spanish and English
Content-Type: text/plain; charset="US-ASCII"
```

```
   Content-Transfer-Encoding: 7bit
   MIME-Version: 1.0

   Hello, this message content is provided in your language.

   --01189998819991197253
   Content-type: message/rfc822
   Content-Language: es; translation-type=human
   Content-disposition: inline

   Subject: =?UTF-8?Q?Ejemplo_pr=C3=A1ctico_de_mensaje_?=
    =?UTF-8?Q?en_espa=C3=B1ol_e_ingl=C3=A9s?=
   Content-Type: text/plain; charset="US-ASCII"
   Content-Transfer-Encoding: 7bit
   MIME-Version: 1.0

   Hola, el contenido de este mensaje esta disponible en su idioma.

   --01189998819991197253--
```

8.2. An Example of a Multiple language email message with language independent part

Below is an example of a multiple language email message that has the multilingual preface followed by two language message parts and then a language independent png image.

```
From: Nik
To: Nathaniel
Subject: Example of a message in Spanish and English
Date: Thu, 7 Jul 2016 21:08:00 +0100
MIME-Version: 1.0
Content-type: multipart/multilingual;
        boundary="01189998819991197253"

--01189998819991197253
Content-type: text/plain; charset="UTF-8"
Content-Disposition: inline
Content-transfer-encoding: quoted-printable

This is a message in multiple languages.  It says the
same thing in each language.  If you can read it in one language,
you can ignore the other translations. The other translations may be
presented as attachments or grouped together.

Este es un mensaje en varios idiomas. Dice lo mismo en
cada idioma. Si puede leerlo en un idioma, puede ignorar las otras
traducciones. Las otras traducciones pueden presentarse como archivos
```

adjuntos o agrupados.

```
--01189998819991197253
Content-type: message/rfc822
Content-Language: en; translation-type=original
Content-disposition: inline

Subject: Example of a message in Spanish and English
Content-Type: text/plain; charset="US-ASCII"
Content-Transfer-Encoding: 7bit
MIME-Version: 1.0

Hello, this message content is provided in your language.

--01189998819991197253
Content-type: message/rfc822
Content-Language: es; translation-type=human
Content-disposition: inline

Subject: =?UTF-8?Q?Ejemplo_pr=C3=A1ctico_de_mensaje_?=
 =?UTF-8?Q?en_espa=C3=B1ol_e_ingl=C3=A9s?=
Content-Type: text/plain; charset="US-ASCII"
Content-Transfer-Encoding: 7bit
MIME-Version: 1.0

Hola, el contenido de este mensaje esta disponible en su idioma.

--01189998819991197253
Content-type: message/rfc822; name="Icon"
Content-Language: zxx
Content-disposition: inline

Content-Type: multipart/mixed;
        boundary="99911972530118999881"; charset="US-ASCII"
Content-Transfer-Encoding: 7bit
MIME-Version: 1.0

--99911972530118999881
Content-Type: image/png; name="icon.png"
Content-Disposition: inline
Content-transfer-encoding: base64

iVBORw0KGgoAAAANSUhEUgAAADAAAAAwCAYAAABXAvmHAAAKQ2lDQ1BJQ0MgUHJvZmlsZQAA
SA2dlndUU1... shortened for brevity ...7yxfd1SNsEy+OXr76qr
997zF2hvZYeDEP5ftGV6Xzx2o9MAAAAASUVORK5CYII=

--99911972530118999881--
--01189998819991197253--
```

8.3. An Example of a complex Multiple language email message with language independent part

Below is an example of a more complex multiple language email message. It has the multilingual preface and two language message parts and then a language independent png image. The language message parts have multipart/alternative contents and would therefore require further processing to determine the content to display.

```
From: Nik
To: Nathaniel
Subject: Example of a message in Spanish and English
Date: Thu, 7 Jul 2016 20:55:00 +0100
MIME-Version: 1.0
Content-type: multipart/multilingual;
        boundary="01189998819991197253"

--01189998819991197253
Content-type: text/plain; charset="UTF-8"
Content-Disposition: inline
Content-transfer-encoding: quoted-printable

This is a message in multiple languages. It says the
same thing in each language. If you can read it in one language,
you can ignore the other translations. The other translations may be
presented as attachments or grouped together.

Este es un mensaje en varios idiomas. Dice lo mismo en
cada idioma. Si puede leerlo en un idioma, puede ignorar las otras
traducciones. Las otras traducciones pueden presentarse como archivos
adjuntos o agrupados.

--01189998819991197253
Content-type: message/rfc822
Content-Language: en; translation-type=original
Content-disposition: inline

Subject: Example of a message in Spanish and English
Content-Type: multipart/alternative;
        boundary="72530118999911999881"; charset="US-ASCII"
Content-Transfer-Encoding: 7bit
MIME-Version: 1.0

--72530118999911999881
Content-Type: text/plain; charset="US-ASCII"
Content-Transfer-Encoding: 7bit

Hello, this message content is provided in your language.
```

```
--72530118999911999881
Content-Type: text/html; charset="US-ASCII"
Content-Transfer-Encoding: 7bit

<html><body>Hello, this message content is <b>provided</b> in <i>your</i> langua
ge.</body></html>

--72530118999911999881--
--01189998819991197253
Content-type: message/rfc822
Content-Language: es; translation-type=human
Content-disposition: inline

Subject: =?UTF-8?Q?Ejemplo_pr=C3=A1ctico_de_mensaje_?=
 =?UTF-8?Q?en_espa=C3=B1ol_e_ingl=C3=A9s?=
Content-Type: multipart/alternative;
        boundary="530118999989991197281"; charset="US-ASCII"
Content-Transfer-Encoding: 7bit
MIME-Version: 1.0

--530118999989991197281
Content-Type: text/plain; charset="US-ASCII"
Content-Transfer-Encoding: 7bit

Hola, el contenido de este mensaje esta disponible en su idioma.

--530118999989991197281
Content-Type: text/html; charset="US-ASCII"
Content-Transfer-Encoding: 7bit

<html><body>Hola, el contenido de este <b>mensaje</b> <i>esta</i>
disponible en su idioma.</body></html>

--530118999989991197281--
--01189998819991197253
Content-type: message/rfc822; name="Icon"
Content-Language: zxx
Content-disposition: inline

Content-Type: multipart/mixed;
        boundary="99911972530118999881"; charset="US-ASCII"
Content-Transfer-Encoding: 7bit
MIME-Version: 1.0

--99911972530118999881
Content-Type: image/png; name="icon.png"
Content-Disposition: inline
Content-transfer-encoding: base64
```

```
iVBORw0KGgoAAAANSUhEUgAAADAAAAAwCAYAAABXAvmHAAAKQ2lDQ1BJQ0MgUHJvZmlsZQAA
SA2dlndUU1... shortened for brevity ...7yxfd1SNsEy+OXr76qr
997zF2hvZYeDEP5ftGV6Xzx2o9MAAAAASUVORK5CYII=

--99911972530118999881--
--01189998819991197253--
```

9. Changes from Previous Versions

9.1. Changes from draft-tomkinson-multilangcontent-01 to draft-tomkinson-slim-multilangcontent-00

- File name and version number changed to reflect the proposed WG name SLIM (Selection of Language for Internet Media).
- Replaced the Subject-Translation field in the language message parts with Subject and provided US-ASCII and non-US-ASCII examples.
- Introduced the language-independent message part.
- Many wording improvements and clarifications throughout the document.

9.2. Changes from draft-tomkinson-slim-multilangcontent-00 to draft-tomkinson-slim-multilangcontent-01

- Added Translation-Type in each language message part to identify the source of the translation (original/human/automated).

9.3. Changes from draft-tomkinson-slim-multilangcontent-01 to draft-tomkinson-slim-multilangcontent-02

- Changed Translation-Type to be a parameter for the Content-Language field rather than a new separate field.
- Added a paragraph about using Content-Disposition field to help non-conforming mail clients correctly render the multilingual preface.
- Recommended using a Name parameter on the language part Content-Type to help the recipient identify the translations in non-conforming mail clients.
- Many wording improvements and clarifications throughout the document.

9.4. Changes from draft-tomkinson-slim-multilangcontent-02 to draft-ietf-slim-multilangcontent-00

- Name change to reflect the draft being accepted into SLIM as a working group document.
- Updated examples to use UTF-8 encoding where required.
- Removed references to ’locale’ for identifying language preference.
- Recommended language matching schemes from RFC 4647 [RFC4647].
- Renamed the unmatched part to language independent part to reinforce its intended purpose.
- Added requirement for using Content-Language: zxx in the language independent part.
- Many wording improvements and clarifications throughout the document.

9.5. Changes from draft-ietf-slim-multilangcontent-00 to draft-ietf-slim-multilangcontent-01

- Changed the inner content type to require message/rfc822 or message/global.
- Updated the examples to reflect the new inner content types.
- Added to the security considerations to highlight the risk from insufficient spam filters.

10. Acknowledgements

The authors are grateful for the helpful input received from many people but would especially like to acknowledge the help of Harald Alvestrand, Stephane Bortzmeyer, Eric Burger, Mark Davis, Doug Ewell, Randall Gellens, Gunnar Hellstrom, Sean Leonard, John Levine, Alexey Melnikov, Addison Phillips, Pete Resnick, Fiona Tomkinson, Simon Tyler and Daniel Vargha.

The authors would also like to thank Fernando Alvaro and Luis de Pablo for their work on the Spanish translations.

11. IANA Considerations

The multipart/multilingual MIME type will be registered with IANA.

12. Security Considerations

Whilst it is intended that each language message part is a direct translation of the original message, this may not always be the case and these parts could contain undesirable content. Therefore there is a possible risk that undesirable text or images could be shown to the recipient if the message is passed through a spam filter that does not check all of the message parts. The risk should be minimal due to the fact that an unknown multipart subtype should be treated as multipart/mixed and so each message part should be subsequently scanned.

13. References

13.1. Normative References

[RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, DOI 10.17487/RFC2046, November 1996, <http://www.rfc-editor.org/info/rfc2046>.

[RFC2047] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, DOI 10.17487/RFC2047, November 1996, <http://www.rfc-editor.org/info/rfc2047>.

[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>.

[RFC2183] Troost, R., Dorner, S., and K. Moore, Ed., "Communicating Presentation Information in Internet Messages: The Content-Disposition Header Field", RFC 2183, DOI 10.17487/RFC2183, August 1997, <http://www.rfc-editor.org/info/rfc2183>.

[RFC3282] Alvestrand, H., "Content Language Headers", RFC 3282, DOI 10.17487/RFC3282, May 2002, <http://www.rfc-editor.org/info/rfc3282>.

[RFC4647] Phillips, A. and M. Davis, "Matching of Language Tags", BCP 47, RFC 4647, DOI 10.17487/RFC4647, September 2006, <http://www.rfc-editor.org/info/rfc4647>.

[RFC5646] Phillips, A., Ed. and M. Davis, Ed., "Tags for Identifying Languages", BCP 47, RFC 5646, DOI 10.17487/RFC5646, September 2009, <http://www.rfc-editor.org/info/rfc5646>.

13.2. Informational References

[I-D.gellens-slim-negotiating-human-language] Gellens, R., "Negotiating Human Language in Real-Time Communications", draft-gellens-slim-negotiating-human-language-02 (work in progress), July 2015.

Authors' Addresses

Nik Tomkinson
Mimecast Ltd
CityPoint, One Ropemaker Street
London EC2Y 9AW
United Kingdom

Email: rfc.nik.tomkinson@gmail.com

Nathaniel Borenstein
Mimecast Ltd
480 Pleasant Street
Watertown MA 02472
North America

Email: nsb@mimecast.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 21, 2016

R. Gellens

March 20, 2016

Negotiating Human Language in Real-Time Communications
draft-ietf-slim-negotiating-human-language-01

Abstract

Users have various human (natural) language needs, abilities, and preferences regarding spoken, written, and signed languages. When establishing interactive communication ("calls") there needs to be a way to negotiate (communicate and match) the caller's language and media needs with the capabilities of the called party. This is especially important with emergency calls, where a call can be handled by a call taker capable of communicating with the user, or a translator or relay operator can be bridged into the call during setup, but this applies to non-emergency calls as well (as an example, when calling a company call center).

This document describes the need and a solution using new SDP stream attributes.

Table of Contents

1. Introduction

A mutually comprehensible language is helpful for human communication. This document addresses the real-time, interactive side of the issue. A companion document on language selection in email [draft-tomkinson-multilangcontent] addresses the non-real-time side.

When setting up interactive communication sessions (using SIP or other protocols), human (natural) language and media modality (voice, video, text) negotiation may be needed. Unless the caller and callee know each other or there is contextual or out of band information from which the language(s) and media modalities can be determined, there is a need for spoken, signed, or written languages to be negotiated based on the caller's needs and the callee's capabilities. This need applies to both emergency and non-emergency calls. For various reasons, including the ability to establish multiple streams using different media (e.g., voice, text, video), it makes sense to use a per-stream negotiation mechanism, in this case, SDP.

This approach has a number of benefits, including that it is generic (applies to all interactive communications negotiated using SDP) and not limited to emergency calls. In some cases such a facility isn't needed, because the language is known from the context (such as when a caller places a call to a sign language relay center, to a friend, or colleague). But it is clearly useful in many other cases. For example, someone calling a company call center or a Public Safety Answering Point (PSAP) should be able to indicate if one or more specific signed, written, and/or spoken languages are preferred, the callee should be able to indicate its capabilities in this area, and the call proceed using in-common language(s) and media forms.

Since this is a protocol mechanism, the user equipment (UE client) needs to know the user's preferred languages; a reasonable technique could include a configuration mechanism with a default of the language of the user interface. In some cases, a UE could tie language and media preferences, such as a preference for a video stream using a signed language and/or a text or audio stream using a written/spoken language.

Including the user's human (natural) language preferences in the session establishment negotiation is independent of the use of a relay service and is transparent to a voice service provider. For example, assume a user within the United States who speaks Spanish but not English places a voice call using an IMS device. It doesn't matter if the call is an emergency call or not (e.g., to an airline reservation desk). The language information is transparent to the IMS carrier, but is part of the session negotiation between the UE and the terminating entity. In the case of a call to e.g., an airline, the call can be automatically handled by a Spanish-speaking agent. In the case of an emergency call, the Emergency Services IP network (ESInet) and the PSAP may choose to take the language and media preferences into account when determining how to process the call.

By treating language as another attribute that is negotiated along with other aspects of a media stream, it becomes possible to accommodate a range of users' needs and called party facilities. For example, some users may be able to speak several languages, but have a preference. Some called parties may support some of those languages internally but require the use of a translation service for others, or may have a limited number of call takers able to use certain languages. Another example would be a user who is able to speak but is deaf or hard-of-hearing and requires a voice stream plus a text stream (known as voice carry over). Making language a media attribute allows the standard session negotiation mechanism to handle this by providing the information and mechanism for the endpoints to make appropriate decisions.

Regarding relay services, in the case of an emergency call requiring sign language such as ASL, there are two common approaches: the caller initiates the call to a relay center, or the caller places the call to emergency services (e.g., 911 in the U.S. or 112 in Europe). In the former case, the language need is ancillary and supplemental. In the latter case, the ESInet and/or PSAP may take the need for sign language into account and bridge in a relay center. In this case, the ESInet and PSAP have all the standard information available (such as location) but are able to bridge the relay sooner in the call processing.

By making this facility part of the end-to-end negotiation, the question of which entity provides or engages the relay service becomes separate from the call processing mechanics; if the caller directs the call to a relay service then the human language negotiation facility provides extra information to the relay service but calls will still function without it; if the caller directs the call to emergency services, then the ESInet/PSAP are able to take the user's human language needs into account, e.g., by assigning to a

specific queue or call taker or bridging in a relay service or translator.

The term "negotiation" is used here rather than "indication" because human language (spoken/written/signed) is something that can be negotiated in the same way as which forms of media (audio/text/video) or which codecs. For example, if we think of non-emergency calls, such as a user calling an airline reservation center, the user may have a set of languages he or she speaks, with perhaps preferences for one or a few, while the airline reservation center will support a fixed set of languages. Negotiation should select the user's most preferred language that is supported by the call center. Both sides should be aware of which language was negotiated. This is conceptually similar to the way other aspects of each media stream are negotiated using SDP (e.g., media type and codecs).

2. Terminology

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

3. Expected Use

This facility may be used by NENA and 3GPP. NENA has already referenced it in NENA 08-01 (i3 Stage 3 version 2) in describing attributes of calls presented to an ESInet, and may add further details in that or other documents. 3GPP may reference this mechanism in general call handling and emergency call handling. Some CRs introduced in SA1 have anticipated this functionality being provided within SDP.

4. Desired Semantics

The desired solution is a media attribute that may be used within an offer to indicate the preferred language of each media stream, and within an answer to indicate the accepted language. The semantics of including multiple values for a media stream within an offer is that the languages are listed in order of preference.

(While it is true that a conversation among multilingual people often involves multiple languages, the usefulness of providing a way to negotiate this as a general facility is outweighed by the complexity of the desired semantics of the SDP attribute to allow negotiation of multiple simultaneous languages within an interactive media stream.)

5. The existing ’lang’ attribute

RFC 4566 [RFC4566] specifies an attribute ’lang’ which appears similar to what is needed here. It specifies that ’a=lang’ is declarative; multiple ’lang’ attributes indicate that the "media use multiple languages", and that "the order of the attributes indicates the order of importance of the various languages in the ... media" (we interpret this to mean that the media contains all of the languages indicated, for example, a video of an interview by a person speaking one language of a person speaking another, with subtitles in the language of the interviewer; this would list first the language of the interviewer and second the language of the person being interviewed). We need a means to negotiate which language is used in each stream. This difference means that the existing ’lang’ attribute can’t be used and we need to define a new attribute.

The text from RFC 4566 [RFC4566] is:

> a=lang:<language tag>
> This can be a session-level attribute or a media-level attribute. As a session-level attribute, it specifies the default language for the session being described. As a media- level attribute, it specifies the language for that media, overriding any session-level language specified. Multiple lang attributes can be provided either at session or media level if the session description or media use multiple languages, in which case the order of the attributes indicates the order of importance of the various languages in the session or media from most important to least important.
> The "lang" attribute value must be a single [RFC3066] language tag in US-ASCII [RFC3066]. It is not dependent on the charset attribute. A "lang" attribute SHOULD be specified when a session is of sufficient scope to cross geographic boundaries where the language of recipients cannot be assumed, or where the session is in a different language from the locally assumed norm.

A recent search of RFCs and Internet Drafts turned up only one use of the ’lang’ attribute (in a now-expired draft), and that sole use was coincidentally in exactly the way we need (erroniously assuming that the attribute was used for negotiation). The sole use was in an example in a draft not directly related to language, where the initial invitation contains two ’a=lang’ entries for a media stream (for English and Italian) and the OK accepts one of them (Italian).

The example serves as evidence of the need for an SDP attribute with the semantics as described in this document; unfortunately, the existing ’lang’ attribute is not it.

6. Proposed Solution

An SDP attribute seems the natural choice to negotiate human (natural) language of an interactive media stream. The attribute value should be a language tag per RFC 5646 [RFC5646]

6.1. Rationale

The decision to base the proposal at the media negotiation level, and specifically to use SDP, came after significant debate and discussion. From an engineering standpoint, it is possible to meet the objectives using a variety of mechanisms, but none are perfect. None of the proposed alternatives was clearly better technically in enough ways to win over proponents of the others, and none were clearly so bad technically as to be easily rejected. As is often the case in engineering, choosing the solution is a matter of balancing trade-offs, and ultimately more a matter of taste than technical merit. The two main proposals were to use SDP and SIP. SDP has the advantage that the language is negotiated with the media to which it applies, while SIP has the issue that the languages expressed may not match the SDP media negotiated (for example, a session could negotiate video at the SIP level but fail to negotiate any video media stream at the SDP layer).

The mechanism described here for SDP can be adapted to media negotiation protocols other than SDP.

6.2. New ’humintlang-send’ and ’humintlang-recv’ attributes

Rather than re-use ’lang’ we define two new media-level attributes starting with ’humintlang’ (short for "human interactive language") to negotiate which human language is used in each (interactive) media stream. There are two attributes, one ending in "-send" and the other in "-recv" to indicate the language used when sending and receiving media:

```
a=humintlang-send:<language tag>
a=humintlang-recv:<language tag>
```

Each can appear multiple times in an offer for a media stream.

In an offer, the ’humintlang-send’ values constitute a list in preference order (first is most preferred) of the languages the offerer wishes to send using the media, and the ’humintlang-recv’ values constitute a list in preference order of the languages the offerer wishes to receive using the media. In cases where the user wishes to use one media for sending and another for receiving (such as a speech-impaired user who wishes to send using text and receive

using audio), one of the two MAY be unset. In cases where a media is not primarily intended for language (for example, a video or audio stream intended for background only) both SHOULD be unset. In other cases, both SHOULD have the same values in the same order. The two SHOULD NOT be set to languages which are difficult to match together (e.g., specifying a desire to send audio in Hungarian and receive audio in Portuguese will make it difficult to successfully complete the call).

In an answer, 'humintlang-send' is the accepted language the answerer will send (which in most cases is one of the languages in the offer's 'humintlang-recv'), and 'humintlang-recv' is the accepted language the answerer expects to receive (which in most cases is one of the languages in the offer's 'humintlang-send').

Each value MUST be a language tag per RFC 5646 [RFC5646]. RFC 5646 describes mechanisms for matching language tags. While RFC 5646 provides a mechanism accommodating increasingly fine-grained distinctions, in the interest of maximum interoperability for real-time interactive communications, each 'humintlang-send' and 'humintlang-recv' value SHOULD be restricted to the largest granularity of language tags; in other words, it is RECOMMENDED to specify only a Primary-subtag and NOT to include subtags (e.g., for region or dialect) unless the languages might be mutually incomprehensible without them.

In an offer, each language tag value MAY have an asterisk appended as the last character (after the registry value). The asterisk indicates a request by the caller to not fail the call if there is no language in common. See Section 6.3 for more information and discussion.

When placing an emergency call, and in any other case where the language cannot be assumed from context, each media stream in an offer primarily intended for human language communication SHOULD specify one or both 'humintlang-send' and 'humintlang-recv' attributes (to avoid ambiguity).

Note that while signed language tags are used with a video stream to indicate sign language, a spoken language tag for a video stream in parallel with an audio stream with the same spoken language tag indicates a request for a supplemental video stream to see the speaker.

Clients acting on behalf of end users are expected to set one or both 'humintlang-send' and 'humintlang-recv' attributes on each media stream primarily intended for human communication in an offer when placing an outgoing session, but either ignore or take into

consideration the attributes when receiving incoming calls, based on local configuration and capabilities. Systems acting on behalf of call centers and PSAPs are expected to take into account the values when processing inbound calls.

6.3. Advisory vs Required

One important consideration with this mechanism is if the call fails if the callee does not support any of the languages requested by the caller.

In order to provide for maximum likelihood of a successful communication session, especially in the case of emergency calling, the mechanism defined here provides a way for the caller to indicate a preference for the call failing or succeeding when there is no language in common. However, the callee is NOT REQUIRED to honor this preference. For example, a PSAP MAY choose to attempt the call even with no language in common, while a corporate call center MAY choose to fail the call.

The mechanism for indicating this preference is that, in an offer, if the last character of any of the 'humintlang-recv' or 'humintlang-send' values is an asterisk, this indicates a request to not fail the call (similar to SIP Accept-Language syntax). Either way, the called party MAY ignore this, e.g., for the emergency services use case, a PSAP will likely not fail the call.

6.4. Silly States

It is possible to specify a "silly state" where the language specified does not make sense for the media type, such as specifying a signed language for an audio media stream.

An offer MUST NOT be created where the language does not make sense for the media type. If such an offer is received, the receiver MAY reject the media, ignore the language specified, or attempt to interpret the intent (e.g., if American Sign Language is specified for an audio media stream, this might be interpreted as a desire to use spoken English).

A spoken language tag for a video stream in conjunction with an audio stream with the same language might indicate a request for supplemental video to see the speaker.

7. IANA Considerations

IANA is kindly requested to add two entries to the 'att-field (media level only)' table of the SDP parameters registry:

Type	Name	Reference
att-field (media level only)	humintlang-send	(this document)
att-field (media level only)	humintlang-recv	(this document)

Table 1: att-field (media level only)' entries

8. Security Considerations

The Security Considerations of RFC 5646 [RFC5646] apply here (as a use of that RFC). In addition, if the 'humintlang-send' or 'humintlang-recv' values are altered or deleted en route, the session could fail or languages incomprehensible to the caller could be selected; however, this is also a risk if any SDP parameters are modified en route.

9. Changes from Previous Versions

9.1. Changes from draft-ietf-slim-...-00 to draft-ietf-slim-...-01

- FOO

9.2. Changes from draft-gellens-slim-...-03 to draft-ietf-slim-...-00

- Updated title to reflect WG adoption

9.3. Changes from draft-gellens-slim-...-02 to draft-gellens-slim-...-03

- Removed Use Cases section, per face-to-face discussion at IETF 93
- Removed discussion of routing, per face-to-face discussion at IETF 93

9.4. Changes from draft-gellens-slim-...-01 to draft-gellens-slim-...-02

- Updated NENA usage mention
- Removed background text reference to draft-saintandre-sip-xmpp-chat-04 since that draft expired

9.5. Changes from draft-gellens-slim-...-00 to draft-gellens-slim-...-01

- Revision to keep draft from expiring

9.6. Changes from draft-gellens-mmusic-...-02 to draft-gellens-slim-...-00

- Changed name from -mmusic- to -slim- to reflect proposed WG name
- As a result of the face-to-face discussion in Toronto, the SDP vs SIP issue was resolved by going back to SDP, taking out the SIP hint, and converting what had been a set of alternate proposals for various ways of doing it within SIP into an informative annex section which includes background on why SDP is the proposal
- Added mention that enabling a mutually comprehensible language is a general problem of which this document addresses the real-time side, with reference to [draft-tomkinson-multilangcontent] which addresses the non-real-time side.

9.7. Changes from draft-gellens-mmusic-...-01 to -02

- Added clarifying text on leaving attributes unset for media not primarily intended for human language communication (e.g., background audio or video).
- Added new section Appendix A ("Alternative Proposal: Caller-prefs") discussing use of SIP-level Caller-prefs instead of SDP-level.

9.8. Changes from draft-gellens-mmusic-...-00 to -01

- Relaxed language on setting -send and -receive to same values; added text on leaving on empty to indicate asymmetric usage.
- Added text that clients on behalf of end users are expected to set the attributes on outgoing calls and ignore on incoming calls while systems on behalf of call centers and PSAPs are expected to take the attributes into account when processing incoming calls.

9.9. Changes from draft-gellens-...-02 to draft-gellens-mmusic-...-00

- Updated text to refer to RFC 5646 rather than the IANA language subtags registry directly.
- Moved discussion of existing ’lang’ attribute out of "Proposed Solution" section and into own section now that it is not part of proposal.
- Updated text about existing ’lang’ attribute.
- Added example use cases.
- Replaced proposed single ’humintlang’ attribute with ’humintlang-send’ and ’humintlang-recv’ per Harald’s request/information that

it was a misuse of SDP to use the same attribute for sending and receiving.
- Added section describing usage being advisory vs required and text in attribute section.
- Added section on SIP "hint" header (not yet nailed down between new and existing header).
- Added text discussing usage in policy-based routing function or use of SIP header "hint" if unable to do so.
- Added SHOULD that the value of the parameters stick to the largest granularity of language tags.
- Added text to Introduction to be try and be more clear about purpose of document and problem being solved.
- Many wording improvements and clarifications throughout the document.
- Filled in Security Considerations.
- Filled in IANA Considerations.
- Added to Acknowledgments those who participated in the Orlando ad-hoc discussion as well as those who participated in email discussion and side one-on-one discussions.

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

- Updated text for (possible) new attribute "humintlang" to reference RFC 5646
- Added clarifying text for (possible) re-use of existing ’lang’ attribute saying that the registration would be updated to reflect different semantics for multiple values for interactive versus non-interactive media.
- Added clarifying text for (possible) new attribute "humintlang" to attempt to better describe the role of language tags in media in an offer and an answer.

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

- Changed name of (possible) new attribute from ’humlang" to "humintlang"
- Added discussion of silly state (language not appropriate for media type)
- Added Voice Carry Over example
- Added mention of multilingual people and multiple languages
- Minor text clarifications

10. Contributors

Gunnar Hellstrom deserves special mention for his reviews, assistance, and especially for contributing the core text in Appendix A.

11. Acknowledgments

Many thanks to Bernard Aboba, Harald Alvestrand, Flemming Andreasen, Francois Audet, Eric Burger, Keith Drage, Doug Ewell, Christian Groves, Andrew Hutton, Hadriel Kaplan, Ari Keranen, John Klensin, Paul Kyzivat, John Levine, Alexey Melnikov, James Polk, Pete Resnick, Peter Saint-Andre, and Dale Worley for reviews, corrections, suggestions, and participating in in-person and email discussions.

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>.

[RFC3840] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", RFC 3840, August 2004.

[RFC3841] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Caller Preferences for the Session Initiation Protocol (SIP)", RFC 3841, August 2004.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC5646] Phillips, A. and M. Davis, "Tags for Identifying Languages", BCP 47, RFC 5646, September 2009.

12.2. Informational References

[I-D.iab-privacy-considerations] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", draft-iab-privacy-considerations-09 (work in progress), May 2013.

[I-D.saintandre-sip-xmpp-chat] Saint-Andre, P., Loreto, S., Gavita, E., and N. Hossain, "Interworking between the Session Initiation Protocol (SIP) and the Extensible Messaging and Presence Protocol (XMPP): One-to-One Text Chat", draft-saintandre-sip-xmpp-chat-06 (work in progress), June 2013.

[RFC3066] Alvestrand, H., "Tags for the Identification of Languages", RFC 3066, January 2001.

[draft-tomkinson-multilangcontent] Tomkinson, N. and N. Borenstein, "Multiple Language Content Type", draft-tomkinson-multilangcontent (work in progress), April 2014.

Appendix A. Historic Alternative Proposal: Caller-prefs

The decision to base the proposal at the media negotiation level, and specifically to use SDP, came after significant debate and discussion. It is possible to meet the objectives using a variety of mechanisms, but none are perfect. Using SDP means dealing with the complexity of SDP, and leaves out real-time session protocols that do not use SDP. The major alternative proposal was to use SIP. Using SIP leaves out non-SIP session protocols, but more fundamentally, would occur at a different layer than the media negotiation. This results in a more fragile solution since the media modality and language would be negotiated using SIP, and then the specific media formats (which inherently include the modality) would be negotiated at a different level (typically SDP, especially in the emergency calling cases), making it easier to have mismatches (such as where the media modality negotiated in SIP don’t match what was negotiated using SDP).

An alternative proposal was to use the SIP-level Caller Preferences mechanism from RFC 3840 [RFC3840] and RFC 3841 [RFC3841].

The Caller-prefs mechanism includes a priority system; this would allow different combinations of media and languages to be assigned different priorities. The evaluation and decisions on what to do with the call can be done either by proxies along the call path, or by the addressed UA. Evaluation of alternatives for routing is described in RFC 3841 [RFC3841].

A.1. Use of Caller Preferences Without Additions

The following would be possible without adding any new registered tags:

Potential callers and recipients MAY include in the Contact field in their SIP registrations media and language tags according to the joint capabilities of the UA and the human user according to RFC 3840 [RFC3840].

The most relevant media capability tags are "video", "text" and "audio". Each tag represents a capability to use the media in two-way communication.

Language capabilities are declared with a comma-separated list of languages that can be used in the call as parameters to the tag "language=".

This is an example of how it is used in a SIP REGISTER:

```
REGISTER    user@example.net
Contact:    <sip:user1@example.net> audio; video; text;
            language="en,es,ase"
```

Including this information in SIP REGISTER allows proxies to act on the information. For the problem set addressed by this document, it is not anticipated that proxies will do so using registration data. Further, there are classes of devices (such as cellular mobile phones) that are not anticipated to include this information in their registrations. Hence, use in registration is OPTIONAL.

In a call, a list of acceptable media and language combinations is declared, and a priority assigned to each combination.

This is done by the Accept-Contact header field, which defines different combinations of media and languages and assigns priorities for completing the call with the SIP URI represented by that Contact. A priority is assigned to each set as a so-called "q-value" which ranges from 1 (most preferred) to 0 (least preferred).

Using the Accept-Contact header field in INVITE requests and responses allows these capabilities to be expressed and used during call set-up. Clients SHOULD include this information in INVITE requests and responses.

Example:

```
Accept-Contact:    *; text; language="en"; q=0.2
Accept-Contact:    *; video; language="ase"; q=0.8
```

This example shows the highest preference expressed by the caller is to use video with American Sign Language (language code "ase"). As a fallback, it is acceptable to get the call connected with only English text used for human communication. Other media may of course

be connected as well, without expectation that it will be usable by the caller for interactive communications (but may still be helpful to the caller).

This system satisfies all the needs described in the previous chapters, except that language specifications do not make any distinction between spoken and written language, and that the need for directionality in the specification cannot be fulfilled.

To some degree the lack of media specification between speech and text in language tags can be compensated by only specifying the important medium in the Accept-Contact field.

Thus, a user who wants to use English mainly for text would specify:

```
Accept-Contact:    *;text;language="en";q=1.0
```

While a user who wants to use English mainly for speech but accept it for text would specify:

```
Accept-Contact:    *;audio;language="en";q=0.8
Accept-Contact:    *;text;language="en";q=0.2
```

However, a user who would like to talk, but receive text back has no way to do it with the existing specification.

A.2. Additional Caller Preferences for Asymmetric Needs

In order to be able to specify asymmetric preferences, there are two possibilities. Either new language tags in the style of the humintlang parameters described above for SDP could be registered, or additional media tags describing the asymmetry could be registered.

A.2.1. Caller Preferences for Asymmetric Modality Needs

The following new media tags should be defined:

```
speech-receive
speech-send
text-receive
text-send
sign-send
sign-receive
```

A user who prefers to talk and get text in return in English would register the following (if including this information in registration data):

```
REGISTER    user@example.net
Contact:    <sip:user1@example.net> audio;text;speech-send;text-
            receive;language="en"
```

At call time, a user who prefers to talk and get text in return in English would set the Accept-Contact header field to:

```
Accept-Contact:    *; audio; text; speech-receive; text-send;
                   language="en";q=0.8
Accept-Contact:    *; text; language="en"; q=0.2
```

Note that the directions specified here are as viewed from the callee side to match what the callee has registered.

A bridge arranged for invoking a relay service specifically arranged for captioned telephony would register the following for supporting calling users:

```
REGISTER    ct@ctrelay.net
Contact:    <sip:ct1@ctreley.net> audio; text; speech-receive;
            text-send; language="en"
```

A bridge arranged for invoking a relay service specifically arranged for captioned telephony would register the following for supporting called users:

```
REGISTER    ct@ctrelay.net
Contact:    <sip:ct2@ctreley.net> audio; text; speech-send; text-
            receive; language="en"
```

At call time, these alternatives are included in the list of possible outcome of the call routing by the SIP proxies and the proper relay service is invoked.

A.2.2. Caller Preferences for Asymmetric Language Tags

An alternative is to register new language tags for the purpose of asymmetric language usage.

Instead of using "language=", six new language tags would be registered:

```
humintlang-text-recv
humintlang-text-send
humintlang-speech-recv
humintlang-speech-send
humintlang-sign-recv
humintlang-sign-send
```

These language tags would be used instead of the regular bidirectional language tags, and users with bidirectional capabilities SHOULD specify values for both directions. Services specifically arranged for supporting users with asymmetric needs SHOULD specify only the asymmetry they support.

Author’s Address

Randall Gellens

Email: rg+ietf@randy.pensive.org

SLIM
Internet-Draft
Expires: September 22, 2016

N. Rooney
GSMA

March 21, 2016

SLIM Use Cases
draft-ietf-slim-use-cases-00

Abstract

Use cases for selection of language for internet media.

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 September 22, 2016.

Copyright Notice

1. Introduction

The SLIM working group is developing standards for language selection for non-real-time and real-time communications. There are a number of relevant use cases which could benefit from this functionality including emergency service real-time communications and customer service. This document details the use cases for SLIM and gives some indication of necessary requirements. For each use case a 'Solution' is provided, indicating the implementability of the use case based on draft-ietf-slim-negotiating-human-language-00.

2. Use Cases

Use cases are listed below:

2.1. Single two-way language

The simplest use case. One language and modality both ways in media described in SDP [RFC4566] as audio or video or text. Straightforward. Works for spoken, written and signed languages. An example is when a user makes a voice call and the preferred language of that user is specified in SDP, allowing the answerer to make decisions based on that specification.

- Solution: Possible

2.2. Alternatives in the same modality

Two or more language alternatives in the same modality. Two or more languages both ways in media described in SDP as audio or video or text, but only in one modality. Straightforward. Works for spoken, written and signed languages. The answering part selects. There is a relative preference expressed by the order, and the answering part can try to fulfill that in the best way. An example is a user who makes a voice call and prefers French first and German as their second language, and the answerer selects to speak German as no French speaking abilites are available.

Solution: Possible

2.3. Fairly equal alternatives in different modalities.

Two or more modality alternatives. Two or more languages in different modalities both ways in media described in SDP as audio or video or text. An example is a hearing person also competent in sign language declares both spoken and sign language competence in audio and video. This is fairly straightforward, as long as there is no strong difference in preference for these alternatives. The

indication of sign language competence is needed to avoid invoking relay services in calls with deaf sign language users only indicating sign language.

Solution: Possible

2.4. Last resort indication

One language in the different modalities. Allows the user to indicate one last resort language when no other is available. For example, a hearing user has text capability but want to use that as last resort. (With current hunintlang specification, there is no way to describe preference level between modalities and no way to describe absolute preference.)

Solution: An answering service will have no guidance to which is the preferred modality and may select to use the modality that is the callers last resort even if the preferred alternative is available.

Another practical case can be a sign language user with a small mobile terminal that has some inconvenient means for texting, but sign language will be strongly preferred. In order to not miss any calls, the indication of text as last resort would be desirable. Possible solution: coding of an absolute preference: hi, med, lo together with the tag.

Solution: Need for absolute preference indication.

2.5. Directional capabilities in different modalities

Two or more language alternatives in the different modalities. For example, a hard-of-hearing user strongly prefers to talk and get text back. Getting spoken language input is appreciated. This can be indicated by spoken language two-ways in audio, and reception of language in text. (There is no current solution that says that the text path is important. The answering part may see it as an alternative.)

Solution: Need for preference indication per modality

2.5.1. Fail gracefully?

There currently are methods to indicate that the call shall fail if a language is not met, but that may be too drastic for some users including the one in the above scenario (2.5). It may be important to be able to connect and just say something, or use residual hearing to get something back when the voice is familiar.

Possible solution: coding of an absolute preference together with the tag could solve this case if used together with the directional indications. For example:

"preference: hi, med, lo"

Another solution would be to indicate required grouping of media, however this raises the complexity level.

2.6. Combination of modalities

Similar to Section 2.5, two or more language alternatives in the different modalities. A deaf-blind person may have highest preference for signing to the answerer and then receiving text in return. This requires the indication of sign language output in video and text reception in text, using the current directional attributes. An answering part may seek suitable modalities for each direction and find the only possible combination.

Solution: Need for preference indication per modality

2.7. Person with speech disabilities who prefer speech-to-speech service

One specific language for one specific modality with a speech-speech engine. A person who speaks in a way that is hard to understand, may be used to have support of a speech-to-speech relay service that adds clear speech when needed for the understanding. Typically, only calls with close friends and family might be possible without the relay service.

This user would indicate preference for receiving spoken language in audio. Text output can be indicated but this user might want to use that as last resort. (There is no current coding for vague or unarticulated speech or other needs for a speech-to-speech service.)

A possibility could be to indicate no preference for spoken language out, a coding of proposed assisting service and an indication of text output on a low absolute level.

Solution: Need of service indication, and absolute level of preference indication.

2.8. Person with speech disabilities who prefer to type and hear

Two or more language alternatives for multiple modalities. A person who speaks in a way that may be hard to understand, may be used to using text for output and listen to spoken language for input. This

user would indicate preference for receiving spoken language in audio. Text output modality can be indicated.

If the answering part has text and audio capabilities, there is a match. If only voice, there is a need to invoke a text relay service.

Solution: Need of service indication, and absolute level of preference indication.

2.9. All Possibilities

Mutiple languages and multiple modalities. For example: a tele-sales center calls out and wants to offer all kinds of possibilities so that the answering party can select. A tele-sales center has competence in multiple spoken languages and can invoke relay services rapidly if needed. So, it indicates in the call setup competence in a number of spoken languages in audio, a number of sign languages in video and a number of written languages in text. This would allow, as a further example, a deaf-blind person who prefers to sign out and get text back answers with only these capabilities. The center can detect that and act accordingly, this could work in the following methods:

- Solution Alternative 1: The center calls without SDP. A deafblind user includes its SDP offer and the center sees what is needed to fulfill the call.

- Solution Alternative 2: The center calls out with only the spoken language capabilities indicated that the caller can handle.

The deaf-blind answering person, or terminal or service provider detects the difference compared to the capabilities of the answering party, and adds a suitable relay service. (This does not use all the offerings of the callers competence to pull in extra services, but is maybe a more realistic case for what usually happens in practice.)

Solution: Possible in the same way as cases 1.8.

3. Final Comments

The use cases identified here try to cover all cases of when users wish to make text, voice or video communication using the language of set of languages in which they are able to speak, write or sign and for which the receivers are also able to communicate. Some of these use cases go even further to allow give some users the ability to select multiple and different languages based on their abilities and needs.

To fulfill all the use cases the currently specified directionality will be needed, as well as an indication of absolute preference. An indication of suitable service and its spoken language is needed for the speech-to-speech case, but can be useful for other cases as well. There is no clear need for explicit grouping of modalities seem to be needed.

Subsequent work in the Selection of Language for Internet Media Working Group (SLIM: https://datatracker.ietf.org/wg/slim/charter/) will work on Internet Drafts to support these use cases.

4. Security Considerations

Indications of user preferred language may give indications as to their nationality, background and abilities. It may also give indication to any possible disabilities and some existing and ongoing health issues.

5. IANA Considerations

This document has no IANA actions.

6. Informative References

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <http://www.rfc-editor.org/info/rfc4566>.

Appendix A. Acknowledgments

Gunnar Hellstrom’s experience and knowledge in this area provided a great deal of these use cases. Thanks also goes to Randall Gellens and Brian Rosen.

Author’s Address

Natasha Rooney
GSMA

Email: nrooney@gsma.com
URI: https://gsma.com