ï»¿

                Interoperability Profile for Relay User Equipment

Abstract

   Video Relay Service (VRS) is a term used to describe a method by
   which a hearing person can communicate with a sign language speaker
   who is deaf, deafblind, or hard of hearing (HoH) or has a speech
   disability using an interpreter (i.e., a Communications Assistant
   (CA)) connected via a videophone to the sign language speaker and an
   audio telephone call to the hearing user. The CA interprets using
   sign language on the videophone link and voice on the telephone link.
   Often the interpreters may be employed by a company or agency, termed
   a "provider" in this document. The provider also provides a video
   service that allows users to connect video devices to their service
   and subsequently to CAs and other sign language speakers. It is
   desirable that the videophones used by the sign language speaker
   conform to a standard so that any device may be used with any
   provider and that direct video calls between sign language speakers
   work. This document describes the interface between a videophone and
   a provider.

Status of This Memo

   This is an Internet Standards Track document.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Further information on
   Internet Standards is available in Section 2 of RFC 7841.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   https://www.rfc-editor.org/info/rfc9248.

Copyright Notice

Table of Contents

1. Introduction

   Video Relay Service (VRS) is a form of Telecommunications Relay
   Service (TRS) that enables people with hearing disabilities who use
   sign language, such as American Sign Language (ASL), to communicate
   with voice telephone users through video equipment. These services
   also enable communication between such individuals directly in
   suitable modalities, including any combination of sign language via
   video, real-time text, and speech.

   This interoperability profile for Relay User Equipment (RUE) is a
   profile of the Session Initiation Protocol (SIP) and related media
   protocols that enables end-user equipment registration and calling
   for VRS calls. It specifies the minimal set of call flows and IETF
   and ITU-T standards that must be supported, provides guidance where
   the standards leave multiple implementation options, and specifies
   minimal and extended capabilities for RUE calls.

   Both subscriber-to-provider (interpreted) and direct subscriber-to-
   subscriber calls are supported on this interface. While there are
   some accommodations in this document to maximize backwards
   compatibility with other devices and services that are used to
   provide VRS service, backwards compatibility is not a requirement,
   and some interwork may be required to allow direct video calls to
   older devices. This document only describes the interface between
   the device and the provider, not any other interface the provider may
   have.

   The following illustrates a typical relay call. The RUE device and
   the communications assistant (sign language interpreter) have

videophones. The hearing user has a telephone (mobile or fixed).

```
                              ||== RUE Interface (this document)
                              ||
                              \/
  +------+   +------+    -        +--------+    -       +-------+
  |User  |   |RUE   |   (   )     |Provider|   (   )    |User   |
  |who is|   |Device|<-(Internet)->|        |           |who is |
  |Deaf  |<->|      |    -        |        |<-( PSTN )->|Hearing|
  +------+   +------+   --------   +--------+   ------   +-------+
                                       ^
                                       |
                             +--------------+
                             |Communications|
                             |Assistant     |
                             +--------------+
```
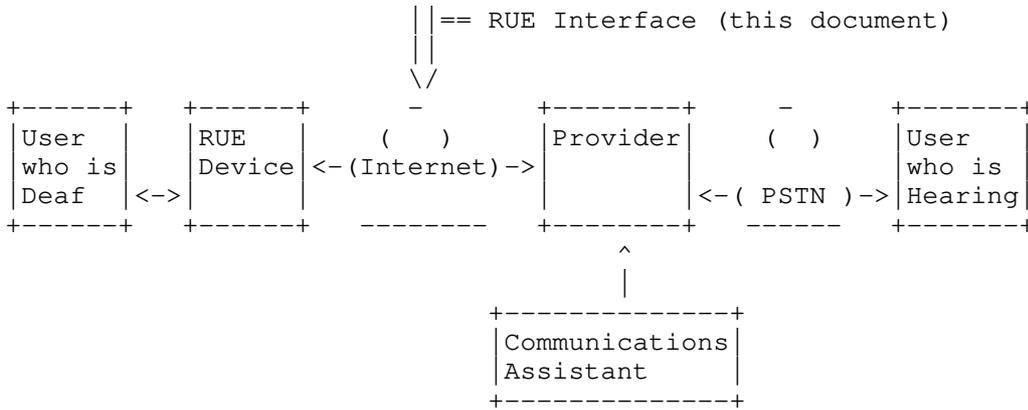
2. Terminology

   Communications Assistant (CA):
      A sign-language interpreter working for a VRS provider, providing
      functionally equivalent phone service.

   Communication modality (modality):
      A specific form of communication that may be employed by two
      users, e.g., English voice, Spanish voice, American Sign Language,
      English lipreading, or French real-time text. Here, one
      communication modality is assumed to encompass both the language
      and the way that language is exchanged. For example, English
      voice and French voice are two different communication modalities.

   Default video relay service:
      The video relay service operated by a subscriber's default VRS
      provider.

   Default video relay service provider (default provider):
      The VRS provider that registers and assigns a telephone number to
      a specific subscriber and, by default, provides the VRS for
      incoming voice calls to the user. The default provider, also by
      default, provides the VRS for outgoing relay calls. The user can
      have more than one telephone number, and each has a default
      provider.

   Outbound dial-around call:
      A relay call where the subscriber specifies the use of a VRS
      provider other than the default VRS provider. This can be
      accomplished by the user dialing a "front-door" number for a VRS
      provider and signing or texting a phone number to call ("two-
      stage"). Alternatively, this can be accomplished by the user's
      RUE software instructing the server of its default VRS provider to
      automatically route the call through the alternate provider to the
      desired Public Switched Telephone Network (PSTN) directory number
      ("one-stage"). Dial-around is per call; for any call, a user can
      use the default VRS provider or any dial-around VRS provider.

   Full Intra Request (FIR):
      A request to a video media sender, requiring that media sender to
      send a decoder refresh point at the earliest opportunity. FIR is
      sometimes known as "instantaneous decoder refresh request", "video
      fast update request", or "fast update request".

   Point-to-Point Call (P2P Call):
      A call between two RUEs, without including a CA.

   Relay call:
      A call that allows people with hearing or speech disabilities to
      use a RUE to talk to users of conventional voice services with the
      aid of a CA to relay the communication.

   Relay service (RS):
      A service that allows a registered subscriber to use a RUE to make

and receive relay calls, point-to-point calls, and relay-to-relay
calls. The functions provided by the relay service include the
provision of media links supporting the communication modalities
used by the caller and callee, user registration and validation,
authentication, authorization, automatic call distributor (ACD)
platform functions, routing (including emergency call routing),
call setup, mapping, call features (such as call forwarding and
video mail), and assignment of CAs to relay calls.

Relay service provider (provider):
   An organization that operates a relay service. A subscriber
   selects a relay service provider to assign and register a
   telephone number for their use, to register with for receipt of
   incoming calls, and to provide the default service for outgoing
   calls.

Relay user:
   Please refer to "subscriber".

Relay user E.164 Number (user E.164):
   The telephone number (in ITU-T E.164 format) assigned to the user.

Relay User Equipment (RUE):
   A SIP user agent (UA) enhanced with extra features to support a
   subscriber in requesting, receiving, and using relay calls. A RUE
   may take many forms, including a stand-alone device; an
   application running on a general-purpose computing device, such as
   a laptop, tablet, or smartphone; or proprietary equipment
   connected to a server that provides the RUE interface.

RUE interface:
   The interfaces described in this document between a RUE and a VRS
   provider who supports it.

Sign language:
   A language that uses hand gestures and body language to convey
   meaning, including, but not limited to, American Sign Language
   (ASL).

Subscriber:
   An individual who has registered with a provider and who obtains
   service by using a RUE. This is the conventional telecom term for
   an end-user customer, which in this case is a relay user. A user
   may be a subscriber to more than one VRS provider.

Video Relay Service (VRS):
   A relay service for people with hearing or speech disabilities who
   use sign language to communicate using video equipment (video RUE)
   with other people in real time. The video link allows the CA to
   view and interpret the subscriber's signed conversation and relay
   the conversation back and forth with the other party.

3. Requirements Language

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and
   "OPTIONAL" in this document are to be interpreted as described in
   BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all
   capitals, as shown here. Lower- or mixed-case uses of these key
   words are not to be interpreted as carrying special significance.

4. General Requirements

   All HTTP/HTTPS [RFC9110] [RFC9112] connections specified throughout
   this document MUST use HTTPS. Both HTTPS and all SIP connections
   MUST use TLS conforming to at least [RFC7525] and MUST support
   [RFC8446].

   All text data payloads not otherwise constrained by a specification
   in another standards document MUST be encoded as Unicode UTF-8.

Implementations MUST support IPv4 and IPv6. Dual-stack support is
NOT required, and provider implementations MAY support separate
interfaces for IPv4 and IPv6 by having more than one server in the
appropriate SRV record where there is either an A or AAAA record in
each server DNS record but not both. The same version of IP MUST be
used for both signaling and media of a call unless Interactive
Connectivity Establishment (ICE) [RFC8445] is used; in which case,
candidates may explicitly offer IPv4, IPv6, or both for any media
stream.

5. SIP Signaling

Implementations of the RUE interface MUST conform to the following
core SIP standards:

*   [RFC3261] (Base SIP)

*   [RFC3263] (Locating SIP Servers)

*   [RFC3264] (Offer/Answer)

*   [RFC3840] (User Agent Capabilities)

*   [RFC5626] (Outbound)

*   [RFC8866] (Session Description Protocol)

*   [RFC3323] (Privacy)

*   [RFC3605] (RTCP Attribute in SDP)

*   [RFC3311] (UPDATE Method)

*   [RFC5393] (Loop-Fix)

*   [RFC5658] (Record-Route Fix)

*   [RFC5954] (ABNF Fix)

*   [RFC3960] (Early Media)

*   [RFC6442] (Geolocation Header Field)

In the above documents, the RUE device conforms to the requirements
of a SIP user agent, and the provider conforms to the requirements of
the registrar and proxy server where the document specifies different
behavior for different roles. For providers offering a video mail
service, [RFC6665] (SIP Events) MUST be implemented to support the
Message-Waiting Indicator (MWI) (see Section 8).

In addition, implementations MUST conform to:

*   [RFC3327] (Path Header Field)

*   [RFC8445] and [RFC8839] (ICE)

*   [RFC3326] (Reason Header Field)

*   [RFC3515] (REFER Method)

*   [RFC3891] (Replaces Header Field)

*   [RFC3892] (Referred-By Header Field)

Implementations MUST implement full ICE, although they MAY interwork
with user agents that implement ICE-lite.

Implementations MUST include a "User-Agent" header field uniquely
identifying the RUE application, platform, and version in all SIP
requests and MUST include a "Server" header field with the same
content in SIP responses.

Implementations intended to support mobile platforms MUST support
[RFC8599] and MUST use it as at least one way to support waking up
the client from the background state.

The SIP signaling for registration and placing/receiving calls
depends on the configuration of various values into the RUE device.
Section 9.2 describes the configuration mechanism that provides
values that are used in the signaling. When the device starts, the
configuration mechanism is run, which retrieves the configuration
data; then, SIP registration occurs using the values from the
configuration. After registration, calls may be sent or received by
the RUE device.

5.1. Registration

The RUE MUST register with a SIP registrar, following [RFC3261] and
[RFC5626], at a provider it has an account with. If the
configuration (see Section 9.2) contains multiple "outbound-proxies"
in "RueConfigurationData", then the RUE MUST use them as specified in
[RFC5626] to establish multiple flows.

The Request-URI for the REGISTER request MUST contain the "provider-
domain" from the configuration. The To URI and From URI MUST be
identical URIs and formatted as follows:

* if "user-name" is provided: "username@provider-domain"

* if "user-name" is not provided: as specified in Section 5.4, use
   "phone-number" and "provider-domain" from the configuration

The RUE determines the URI to resolve by initially determining if one
or more "outbound-proxies" are configured. If they are configured,
the URI will be that of one of the "outbound-proxies". If no
"outbound-proxies" are configured, the URI will be the Request-URI
from the REGISTER request. The RUE extracts the domain from that URI
and consults the DNS record for that domain. The DNS entry MUST
contain NAPTR records conforming to [RFC3263]. One of those NAPTR
records MUST specify TLS as the preferred transport for SIP. For
example, a DNS NAPTR query for "sip: p1.red.example.net" could
return:

IN NAPTR 50 50 "s" "SIPS+D2T" "" _sips._tcp.p1.red.example.net
IN NAPTR 90 50 "s" "SIP+D2T" "" _sip._tcp.p1.red.example.net

If the RUE receives a 439 (First Hop Lacks Outbound Support) response
to a REGISTER request, it MUST reattempt registration without using
the outbound mechanism.

The registrar MAY authenticate the RUE using SIP digest
authentication. The credentials to be used MUST come from the
configuration in Section 9.2: "user-name" if provided or "phone-
number" if user-name is not provided, and "sip-password". This
"user-name"/"sip-password" combination SHOULD NOT be the same as that
used for other purposes, except as expressly described below, such as
retrieving the RUE configuration or logging into the provider's
customer service portal. [RFC8760] MUST be supported by all
implementations, and SHA-512 digest algorithms MUST be supported.

If the registration request fails with an indication that credentials
from the configuration are invalid, then the RUE MUST retrieve a
fresh version of the configuration. If credentials from a freshly
retrieved configuration are found to be invalid, then the RUE MUST
cease attempts to register and inform the RUE user of the problem.

Support for multiple simultaneous registrations with multiple
providers by the RUE is OPTIONAL for the RUE (and providers do not
need any support for this option).

Multiple simultaneous RUE SIP registrations from different RUE
devices with the same SIP URI SHOULD be permitted by the provider.

The provider MAY limit the total number of simultaneous
registrations. When a new registration request is received that
results in exceeding the limit on simultaneous registrations, the
provider MAY then prematurely terminate another registration;
however, it SHOULD NOT do this if it would disconnect an active call.

If a provider prematurely terminates a registration to reduce the
total number of concurrent registrations with the same URI, it SHOULD
take some action to prevent the affected RUE from automatically re-
registering and re-triggering the condition.

5.2. Session Establishment

5.2.1. Normal Call Origination

   After initial SIP registration, the RUE adheres to SIP [RFC3261]
   basic call flows, as documented in [RFC3665].

   A RUE device MUST route all outbound calls through an outbound proxy
   if configured.

   The SIP URIs in the To field and the Request-URI MUST be formatted as
   specified in Section 5.4 using the destination phone number or as SIP
   URIs as provided in the configuration (Section 9.2). The domain
   field of the URIs SHOULD be the "provider-domain" from the
   configuration (e.g., sip:+15551234567@red.example.com;user=phone),
   except that an anonymous call would not use the provider domain.

   Anonymous calls MUST be supported by all implementations. An
   anonymous call is signaled per [RFC3323].

   The From URI MUST be formatted as specified in Section 5.4, using the
   "phone-number" and "provider-domain" from the configuration. It
   SHOULD also contain the display-name from the configuration when
   present. (Please refer to Section 9.2.)

   Negotiated media MUST follow the requirements specified in Section 6
   of this document.

   To allow time for an unanswered call to time out and direct it to a
   videomail server, the User Agent Client MUST NOT impose a time limit
   less than the default SIP INVITE transaction timeout of 3 minutes.

5.2.2. Dial-Around Origination

   Providers and RUE devices MUST support both one-stage and two-stage
   dial-around.

   Outbound dial-around calls allow a RUE user to select any provider to
   provide interpreting services for any call. "Two-stage" dial-around
   calls involve the RUE calling a telephone number that reaches the
   dial-around provider and using signing or dual-tone multi-frequency
   (DTMF) to provide the called party's telephone number. In two-stage
   dial-around, the To URI is the "front-door" URI (see Section 9.2) in
   "ProviderConfigurationData" of the dial-around provider. The RUE
   Provider Selection service (Section 9.1) can be used by the RUE to
   obtain a list of providers; then, the provider configuration
   (Section 9.2.1) can be used to find the front-door URI for each of
   these providers.

   One-stage dial-around is a method where the called party's telephone
   number is provided in the To URI and the Request-URI, using the
   domain of the dial-around provider.

   For one-stage dial-around, the RUE MUST follow the procedures in
   Section 5.2.1 with the following exception: the domain part of the
   SIP URIs in the To field and the Request-URI MUST be the domain of
   the dial-around provider discovered as described in Section 9.1.

   The following is a partial example of a one-stage dial-around call
   from VRS user +1-555-222-0001 hosted by red.example.com to a hearing

user +1-555-123-4567 using dial-around to green.example.com for the
relay service. Only important details of the messages are shown, and
many header fields have been omitted:

```
  ,-+-.          ,----+----.      ,-----+-----.
  |RUE|          |Default  |      |Dial-Around|
  |   |          |Provider |      | Provider  |
  `-+-'          `----+----'      `-----+-----'
    |                 |                 |
    |  [1] INVITE     |                 |
    |---------------->|  [2] INVITE     |
    |                 |---------------->|
    |                 |                 |
```
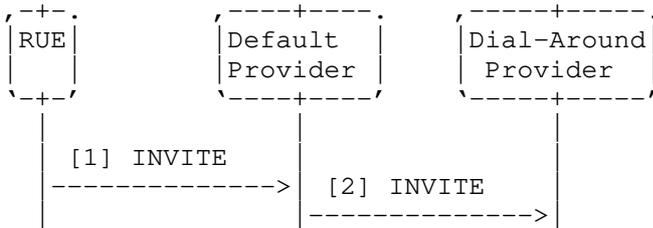
   Message Details:

   [1] INVITE Rue -> Default Provider

   INVITE sip:+15551234567@green.example.net;user=phone SIP/2.0
   To: <sip:+15551234567@green.example.net;user=phone>
   From: "Bob Smith" <sip:+15552220001@red.example.net;user=phone>

   [2] INVITE Default Provider -> Dial-Around Provider

   INVITE sip:+15551234567@green.example.net;user=phone SIP/2.0
   To: <sip:+15551234567@green.example.net;user=phone>
   From: "Bob Smith" sip:+15552220001@red.example.net;user=phone
   P-Asserted-Identity: sip:+15552220001@red.example.net

                  Figure 1: One-Stage Dial-Around

5.2.3. RUE Contact Information

   To identify the owner of a RUE, the initial INVITE for a call from a
   RUE, or the 200 OK the RUE uses to accept a call, identifies the
   owner by sending a Call-Info header field with a purpose parameter of
   "rue-owner". The URI MAY be an HTTPS URI or Content-ID URL. The
   latter is defined by [RFC2392] to locate message body parts. This
   URI type is present in a SIP message to convey the RUE ownership
   information as a MIME body. The form of the RUE ownership
   information is an xCard [RFC6351]. Please refer to [RFC6442] for an
   example of using content indirection URLs in SIP messages. Note that
   use of the content indirection URL usually implies multiple message
   bodies ("mime/multipart"). The RUE owner is the entity that has
   local control over the device that is not necessarily the legal owner
   of the equipment. It often is the user, but that is not necessarily
   true. While no minimum fields in the xCard are specified, the name,
   address, phone number, and email address of the RUE owner are
   expected to be supplied.

5.2.4. Incoming Calls

   The RUE MUST only accept inbound calls sent to it by a proxy
   mentioned in the configuration.

   If multiple simultaneous RUE SIP registrations from different RUE
   devices with the same SIP URI exist, the provider MUST parallel fork
   the call to all registered RUEs so that they ring at the same time.
   The first RUE to reply with a 200 OK answers the call, and the
   provider MUST cancel other call branches using a CANCEL request.

5.2.5. Emergency Calls

   Implementations MUST conform to [RFC6881] for handling of emergency
   calls, except that if the device is unable to determine its own
   location, it MAY send the emergency call without a Geolocation header
   field and without a Route header field (since it would be unable to
   query the Location-to-Service Translation (LoST) server for a route,
   per [RFC6881]). If an emergency call arrives at the provider without
   a Geolocation header field, the provider MUST supply location by
   adding the Geolocation header field and MUST supply the route by
   querying the LoST server with that location.

If the emergency call is to be handled using existing country-
specific procedures, the provider is responsible for modifying the
INVITE to conform to the country-specific requirements. In this
case, the location MAY be extracted from the [RFC6881] conformant
INVITE and used to propagate it to the appropriate country-specific
entities. If the configuration specifies it, an implementation of a
RUE device MAY send a Geolocation header field containing its
location in the REGISTER request. If implemented, users MUST be
offered an opt-out. Country-specific procedures might require the
location to be preloaded in some entity prior to placing an emergency
call; however, the RUE may have a more accurate and timely device
location than the manual, preloaded entry. That information MAY be
used to populate the location to appropriate country-specific
entities. Re-registration SHOULD be used to update the location, so
long as the rate of re-registration is limited if the device is
moving.

Implementations MUST implement additional data [RFC7852]. RUE
devices MUST implement data provider information, device information,
and owner/subscriber information blocks.

5.3. Mid-Call Signaling

Implementations MUST support re-INVITE to renegotiate media session
parameters (among other uses). Per Section 6.8, implementations MUST
be able to support an INFO message for full frame refresh for devices
that do not support SRTCP (please refer to Section 6.1).
Implementations MUST support an in-dialog REFER (as described in
[RFC3515] and updated by [RFC7647], and including support for
norefersub per [RFC4488]) with the Replaces header field [RFC3891] to
enable call transfer.

5.4. URI Representation of Phone Numbers

SIP URIs constructed from non-URI sources (dial strings) and sent to
SIP proxies by the RUE MUST be represented as follows, depending on
whether they can be represented as an E.164 number. In this section,
"expressed as an E.164 number" includes numbers, such as toll-free
numbers that are not actually E.164 numbers but have the same format.

A dial string that can be expressed as an E.164 phone number MUST be
represented as a SIP URI with a URI ";user=phone" tag. The user part
of the URI MUST be in conformance with "global-number", as defined in
[RFC3966]. The user part MUST NOT contain any "visual-separator"
characters, as defined in [RFC3966].

Dial strings that cannot be expressed as E.164 numbers MUST be
represented as dialstring URIs, as specified by [RFC4967], e.g.,
sip:411@red.example.net;user=dialstring.

The domain part of relay service URIs and User Address of Records
(AoR) MUST resolve (per [RFC3263]) to globally routable IPv4 and/or
IPv6 addresses.

5.5. Transport

Implementations MUST conform to [RFC8835], except for its guidance on
the WebRTC data channel, which this specification does not use. See
Section 6.2 for how RUE supports real-time text without the data
channel.

Implementations MUST support SIP outbound [RFC5626] (please also
refer to Section 5.1).

6. Media

This specification adopts the media specifications for WebRTC
[RFC8825]. Where WebRTC defines how interactive media communications
may be established using a browser as a client, this specification
assumes a normal SIP call. Various RTPs, RTCPs, SDPs, and specific

media requirements specified for WebRTC are adopted for this
document. Explicit requirements from the WebRTC suite of documents
are described below .

To use WebRTC with this document, a gateway that presents a WebRTC
server interface towards a browser and a RUE client interface towards
a provider is assumed. The gateway would interwork signaling and, as
noted below, interwork at least any real-time text media in order to
allow a standard browser-based WebRTC client to be a VRS client. The
combination of the browser client and the gateway would be a RUE
user. This document does not specify the gateway.

The following sections specify the WebRTC documents to which
conformance is required. "Mandatory to Implement" means a conforming
implementation MUST implement the specified capability. It does not
mean that the capability must be used in every session. For example,
Opus is a Mandatory-to-Implement audio codec, and all conforming
implementations must support Opus. However, an implementation
presenting a call across the RUE interface (where the call originates
in the PSTN or an older, non-RUE-compatible device, which only offers
G.711 audio) does not need to include the Opus codec in the offer,
since it cannot be used with that call. Conformance to this document
allows end-to-end RTCP and media congestion control for audio and
video.

6.1. SRTP and SRTCP

   Implementations MUST support [RFC8834], except that MediaStreamTracks
   are not used. Implementations MUST conform to Section 6.4 of
   [RFC8827].

6.2. Text-Based Communication

   Implementations MUST support real-time text [RFC4102] [RFC4103] via
   T.140 media. One original and two redundant generations MUST be
   transmitted and supported with a 300 ms transmission interval.
   Implementations MUST support [RFC9071], especially for emergency
   calls. Note that [RFC4103] is not how real-time text is transmitted
   in WebRTC, and some form of transcoder would be required to interwork
   real-time text in the data channel of WebRTC to [RFC4103] real-time
   text.

   Transport of T.140 real-time text in WebRTC is specified in
   [RFC8865], using the WebRTC data channel. [RFC8865] also has some
   advice on how gateways between [RFC4103] and [RFC8865] should
   operate. It is RECOMMENDED that [RFC8865], including multiparty
   support, be used for communication with browser-based WebRTC
   implementations. Implementations MUST support [RFC9071].

6.3. Video

   Implementations MUST conform to [RFC7742] with the following
   exceptions: only H.264, as specified in [RFC7742], is Mandatory to
   Implement, and VP8 support is OPTIONAL at both the device and
   providers. This is because backwards compatibility is desirable, and
   older devices do not support VP8.

6.4. Audio

   Implementations MUST conform to [RFC7874].

6.5. DTMF Digits

   Implementations MUST support the "audio/telephone-event" [RFC4733]
   media type. They MUST support conveying event codes 0 through 11
   (DTMF digits "0"-"9", "*","#") defined in Table 7 of [RFC4733].
   Handling of other tones is OPTIONAL.

6.6. Session Description Protocol

   The SDP offers and answers MUST conform to the SDP rules in [RFC8829]

except that the RUE interface uses SIP transport for SDP. The SDP
for real-time text MUST specify the T.140 payload type [RFC4103].

6.7. Privacy

   The RUE MUST provide for user privacy by implementing a local one-way
   mute, without signaling, for both audio and video. However, RUE MUST
   maintain any states in the network (e.g., NAT bindings) by
   periodically sending media packets on all active media sessions
   containing silence, comfort noise, blank screen, etc., per [RFC6263].

6.8. Negative Acknowledgement, Picture Loss Indicator, and Full
      Intraframe Request Features

   The NACK, FIR, and Picture Loss Indicator (PLI) features, as
   described in [RFC4585] and [RFC5104], MUST be implemented.
   Availability of these features MUST be announced with the "ccm"
   feedback value. NACK should be used when negotiated and conditions
   warrant its use and the other end supports it. Signaling picture
   losses as a PLI should be preferred. FIR should be used only in
   situations where not sending a decoder refresh point would render the
   video unusable for the users, as per Section 4.3.1.2 of [RFC5104].

   For backwards compatibility with calling devices that do not support
   the foregoing methods, implementations MUST implement SIP INFO
   messages to send and receive XML-encoded Picture Fast Update messages
   according to [RFC5168].

7. Contacts

7.1. CardDAV Login and Synchronization

   Support of vCard Extensions to WebDAV (CardDAV) by providers is
   OPTIONAL.

   The RUE MUST and providers MAY be able to synchronize the user's
   contact directory between the RUE endpoint and one maintained by the
   user's VRS provider using CardDAV [RFC6352] [RFC6764].

   The configuration (see Section 9.2) RueConfigurationData MAY supply a
   "carddav-username" and "carddav-domain" identifying a CardDAV server
   and address book for this account, plus an optional "carddav-
   password".

   To access the CardDAV server and address book, the RUE MUST follow
   Section 6 of [RFC6764], using the configured carddav-username and
   carddav-domain in place of an email address. If the request triggers
   a challenge for digest authentication credentials, the RUE MUST
   continue using matching carddav-username and carddav-password from
   the configuration. If no carddav-username and carddav-password are
   configured, the RUE MUST use the SIP user-name and sip-password from
   the configuration. If the SIP credentials fail, the RUE MUST query
   the user.

   Synchronization using CardDAV MUST be a two-way synchronization
   service, with proper handling of asynchronous adds, changes, and
   deletes at either end of the transport channel.

   The RUE MAY support other CardDAV services.

7.2. Contacts Import/Export Service

   Implementations MUST be able to export/import the list of contacts in
   xCard [RFC6351] XML format.

   The RUE accesses this service via the "contacts-uri" in the
   configuration. The URL MUST resolve to identify a web server
   resource that imports/exports contact lists for authorized users.

   The RUE stores/retrieves the contact list (address book) by issuing
   an HTTPS POST or GET request. If the request triggers a challenge

for digest authentication credentials, the RUE MUST attempt to
continue using the "contacts-username" and "contacts-password" from
the configuration. If no contacts-username is configured, the SIP
user-name from the configuration is used; if the SIP user-name is not
configured, the phone-number is used. If user-name or phone-number
is used, the sip-password is used to authenticate to the contact list
server.

8. Video Mail

   Support for video mail includes a retrieval mechanism and a Message-
   Waiting Indicator (MWI). Message storage is not specified by this
   document. RUE devices MUST support message retrieval using a SIP
   call to a specified SIP URI using DTMF to manage the mailbox, as well
   as a browser-based interface reached at a specified HTTPS URI. If a
   provider supports video mail, at least one of these mechanisms MUST
   be supported. RUE devices MUST support both. See Section 9.2 for
   how the URI to reach the retrieval interface is obtained.

   Implementations MUST support subscriptions to "message-summary"
   events [RFC3842] to the URI specified in the configuration.
   Providers MUST support MWI if they support video mail. RUE devices
   MUST support MWI.

   The "videomail" and "mwi" properties in the configuration (see
   RueConfigurationData in Section 9.2.2) give the URIs for message
   retrieval and "message-summary" subscription.

   In notification bodies, if detailed message summaries are available,
   messages with video MUST be reported using "message-context-class
   multimedia-message", as defined in [RFC3458] .

9. Provisioning and Provider Selection

   To simplify how users interact with RUE devices, the RUE interface
   separates provisioning into two parts. One provides a directory of
   providers so that a user interface can allow easy provider selection
   either for registering or for dial-around. The other provides
   configuration data for the device for each provider.

9.1. RUE Provider Selection

   To allow the user to select a relay service, the RUE MAY at any time
   obtain (typically on startup) a list of providers that provide
   service in a country. IANA has established a registry that contains
   a two-letter country code and a list entry point string (see
   Section 10.1). The entry point, when used with the following OpenAPI
   interface, returns a list of provider names for a country code
   suitable for display, with a corresponding entry point to obtain
   information about that provider. No mechanism to determine the
   country where the RUE is located is specified in this document.
   Typically, the country is the home country of the user but may be a
   local country while traveling. Some countries allow support from
   their home country when traveling abroad. Regardless, the RUE device
   will need to allow the user to choose the country.

   Each country that supports VRS using this specification MAY support
   the provider list. This document does not specify who maintains the
   list. Some possibilities are a regulator or an entity designated by
   a regulator, an agreement among providers to provide the list, or a
   user group.

   The interface to obtain the list of providers is described by an
   OpenAPI [OpenAPI] interface description. In that interface
   description, the "servers" component includes an occurrence of
   "localhost". The value from the registry of the "list entry point"
   string for the desired country is substituted for "localhost" in the
   "servers" component to obtain the server URI prefix of the interface
   to be used to obtain the list of providers for that country. The
   "Providers" path then specifies the rest of the URI used to obtain
   the list. For example, if the list entryPoint is "example.com/api",

the provider list would be obtained from
https://example.com/api/rum/v1/Providers.

The V1.0 "ProviderList" is a JSON object consisting of an array where
each entry describes one provider. Each entry consists of the
following items:

*   name: This parameter contains the text label identifying the
    provider and is meant to be displayed to the human VRS user.

*   providerEntryPoint: A string used for configuration purposes by
    the RUE (as discussed in Section 9.2). The string MUST start with
    a domain but MAY include other URI path elements after the domain.

The VRS user interacts with the RUE to select from the provider list
one or more providers with whom the user has already established an
account, wishes to establish an account, or wishes to use the
provider for a one-stage dial-around.

```
{
   "providers": [
      {
         "name": "Red",
         "entryPoint": "red.example.net"
      },
      {
         "name": "Green",
         "entryPoint": "green.example.net"
      },
      {
         "name": "Blue",
         "entryPoint": "blue.example.net"
      }
   ]
}
```

                 Figure 2: Example of a ProviderList JSON Object

9.2. RUE Configuration Service

   A RUE device may retrieve a provider configuration using a simple
   HTTPs web service. There are two entry points. One is used without
   user credentials, and the response includes configuration data for
   new user signup and dial-around. The other uses a locally stored
   username and password that results from a new user signup to
   authenticate to the interface and returns configuration data for the
   RUE.

   The interface to obtain configuration data is described by an OpenAPI
   [OpenAPI] interface description. In that interface description, the
   "servers" component string includes an occurrence of "localhost".
   The entry point string obtained from the provider list (Section 9.1)
   is substituted for "localhost" to obtain the server prefix of the
   interface. The path then specifies the rest of the URI used to
   obtain the list. For example, if the entryPoint from the provider
   list is "red.example.net", the provider configuration would be
   obtained from https://red.example.net/rum/V1/ProviderConfig and the
   RUE configuration would be obtained from
   https://red.example.net/rum/V1/RueConfig.

   In both the queries, an optional parameter may be provided to the
   interface, which is an API Key (apiKey). The implementation MAY have
   an apiKey obtained from the provider and specific to the
   implementation. The method used to obtain the apiKey is not
   specified in this document. The provider MAY refuse to provide
   service to an implementation presenting an apiKey it does not
   recognize.

   Also in both queries, the RUE device provides a client-provided,
   required parameter, which contains an instance identifier
   (instanceId). This parameter MUST be the same value each time this

instance (same implementation on same device) queries the interface.
This MAY be used by the provider, for example, to associate a
location with the instance for emergency calls. This should be
globally unique. A Universally Unique Identifier (UUID) is
suggested.

For example, a query for the RUE configuration could be
https://red.example.net/rum/V1/RueConfig?apiKey="t65667Ajjss90uuuDisK
t8999"&instanceId="5595b5a3-0687-4b8e-9913-a7f2a04fb7bd"

The data returned is a JSON object consisting of key/value
configuration parameters to be used by the RUE.

The configuration data payload includes the following data items.
Items not noted as (OPTIONAL) are REQUIRED. If other unexpected
items are found, they MUST be ignored.

9.2.1. Provider Configuration

* signup: (OPTIONAL) an array of JSON objects consisting of:

   - language: entry from the IANA "Language Subtag Registry"
      (https://www.iana.org/assignments/language-subtag-registry).
      Normally, this would be a written language tag.

   - uri: a URI to the website for creating a new account in the
      supported language. The new user signup URI may only initiate
      creation of a new account. Various vetting, approval, and
      other processes may be needed, which could take time, before
      the account is established. The result of creating a new
      account would be account credentials (e.g., username and
      password), which would be manually entered into the RUE device
      that forms the authentication parameters for the RUE
      configuration service described below in Section 9.2.2.

* dial-around: an array of JSON objects consisting of:

   - language: entry from the IANA "Language Subtag Registry".
      Normally, this would be a sign language tag.

   - front-door: a URI to a queue of interpreters supporting the
      specified language for a two-stage dial-around.

   - oneStage: a URI that can be used with a one-stage dial-around
      (Section 5.2.2) using an interpreter supporting the specified
      language.

* helpDesk: (OPTIONAL) an array of JSON objects consisting of:

   - language: entry from the IANA "Language Subtag Registry".
      Normally, this would be a sign language tag; although, it could
      be a written language tag if the help desk only supports a chat
      interface.

   - uri: a URI that reaches a help desk for callers supporting the
      specified language. The URI MAY be a SIP URI for help provided
      with a SIP call or MAY be an HTTPS URI for help provided with a
      browser interface.

   A list is specified so that the provider can offer multiple
   choices to users for language and interface styles.

9.2.2. RUE Configuration

* lifetime: (OPTIONAL) specifies how long (in seconds) the RUE MAY
   cache the configuration values. Values may not be valid when
   lifetime expires. If the RUE caches configuration values, it MUST
   cryptographically protect them against unauthorized disclosure
   (e.g., by other applications on the platform the RUE is built on).
   The RUE SHOULD retrieve a fresh copy of the configuration before
   the lifetime expires or as soon as possible after it expires. The

lifetime is not guaranteed, i.e., the configuration may change
before the lifetime value expires. In that case, the provider MAY
indicate this by generating authorization challenges to requests
and/or prematurely terminating a registration. Emergency calls
MUST continue to work. If not specified, the RUE MUST fetch new
configuration data every time it starts.

* sip-password: (OPTIONAL) a password used for SIP, STUN, and TURN
   authentication. The RUE device retains this data, which it MUST
   cryptographically protect against unauthorized disclosure (e.g.,
   by other applications on the platform the RUE is built on). If it
   is not supplied but was supplied on a prior invocation of this
   interface, the most recently supplied password MUST be used. If
   it was never supplied, the password used to authenticate to the
   configuration service is used for SIP, as well as STUN and TURN
   servers mentioned in this configuration.

* phone-number: (REQUIRED) the telephone number (in E.164 format)
   assigned to this subscriber. This becomes the user portion of the
   SIP URI identifying the subscriber.

* user-name: (OPTIONAL) a username used for authenticating to the
   provider. If not provided, phone-number is used.

* display-name: (OPTIONAL) a human-readable display name for the
   subscriber.

* provider-domain: (REQUIRED) the domain for the provider. This
   becomes the server portion of the SIP URI identifying the
   subscriber.

* outbound-proxies: (OPTIONAL) an array of URIs of SIP proxies to be
   used when sending requests to the provider.

* mwi: (OPTIONAL) a URI identifying a SIP event server that
   generates "message-summary" events for this subscriber.

* videomail: (OPTIONAL) a SIP or HTTPS URI that can be used to
   retrieve video mail messages.

* contacts: (OPTIONAL) an HTTPS URI ("contacts-uri"), (OPTIONAL)
   "contacts-username", and "contacts-password" that may be used to
   export (retrieve) the subscriber's complete contact list managed
   by the provider. At least the URI MUST be provided if the
   subscriber has contacts. If contact-username and contacts-
   password are not supplied, the sip credentials are used. If the
   contacts-username is provided, contacts-password MUST be provided.
   If contacts-password is provided, contacts-username MUST be
   provided.

* carddav: (OPTIONAL) an address ("carddav-domain"), (OPTIONAL)
   "carddav-username", and "carddav-password" identifying a "CardDAV"
   server and account that can be used to synchronize the RUE's
   contact list with the contact list managed by the provider. If
   carddav-username and carddav-password are not supplied, the sip
   credentials are used. If the carddav-username is provided,
   carddav-password MUST be provided. If carddav-password is
   provided, carddav-username MUST be provided.

* sendLocationWithRegistration: (OPTIONAL) true if the RUE should
   send a Geolocation header field with REGISTER; false if it should
   not. Defaults to false if not present.

* ice-servers: (OPTIONAL) an array of server types and URLs
   identifying STUN and TURN servers available for use by the RUE for
   establishing media streams in calls via the provider. If the same
   URL provides both STUN and TURN services, it MUST be listed twice,
   each with different server types.

9.2.3. Versions

Both web services also have a simple version mechanism that returns a
list of versions of the web service it supports. This document
describes version 1.0. Versions are displayed as a major version,
followed by a period ".", followed by a minor version, where the
major and minor versions are integers. A backwards compatible change
within a major version MAY increment only the minor version number.
A non-backwards, compatible change MUST increment the major version
number. Backwards compatibility applies to both the server and the
client. Either may have any higher or lower minor revision and
interoperate with its counterpart with the same major version. To
achieve backwards compatibility, implementations MUST ignore any
object members they do not implement. Minor version definitions
SHALL only add objects, optional members of existing objects, and
non-mandatory-to-use functions and SHALL NOT delete or change any
objects, members of objects, or functions. This means an
implementation of a specific major version and minor version is
backwards compatible with all minor versions of the major version.
The version mechanism returns an array of supported versions, one for
each major version supported, with the minor version listed being the
highest-supported minor version.

Unless the per-country provider list service is operated by a
provider at the same base URI as that provider's configuration
service, the version of the configuration service MAY be different
from the version of the provider list service.

```
{
  "versions": [
    {
     "major": 1,
     "minor": 6
    },
    {
     "major": 2,
     "minor": 13
    },
    {
     "major": 3,
     "minor": 2
    }
  ]
}
```

                 Figure 3: Example of a Version JSON Object

9.2.4. Examples

```
{
  "signUp": [
     { "language" : "en", "uri" : "https:hello-en.example.net"},
     { "language" : "es", "uri" : "https:hello-es.example.net"} ] ,
  "dial-around": [
     { "language" : "en", "front-door" : "sip:fd-en.example.net",
          "oneStage" : "sip:1stg-eng.example.com" } ,
     { "language" : "es", "front-door" : "sip:fd-es.example.net",
          "oneStage" : "sip:1stg-spn.example.com" } ] ,
  "helpDesk": [
     { "language" : "en", "uri" : "sip:help-en.example.net"} ,
     { "language" : "es", "uri" : "sip:help-es.example.net"} ]
}
```

            Figure 4: Example JSON Provider Configuration Payload

```
{
  "lifetime": 86400,
  "display-name" : "Bob Smith",
  "phone-number": "+15551234567",
  "provider-domain": "red.example.net",
  "outbound-proxies": [
    "sip:p1.red.example.net",
    "sip:p2.red.example.net" ],
```

```
      "mwi": "sip:+15551234567@red.example.net;user=phone",
      "videomail": "sip:+15551234567@vm.red.example.net;user=phone",
      "contacts": {
        "contacts-uri":
           "https://red.example.net:443/c/3617b719-2c3a-46f4-9c13",
        "contacts-username": "bob",
        "contacts-password": "XhOT4ch@ZEi&3u2xEYQNMO^5UGb"
      },
      "carddav": {
        "carddav-domain": "carddav.example.com",
        "carddav-username": "bob",
        "carddav-password": "sj887%dd*jJty%87hyys5hHT"
      },
      "sendLocationWithRegistration": false,
      "ice-servers": [
        {"stun":"stun.red.example.com:19302" },
        {"turn":"turn.red.example.com:3478"}
      ]
    }
```

Figure 5: Example JSON RUE Configuration Payload

9.2.5. Using the Provider Selection and RUE Configuration Services Together

One way to use these two services is:

1. At startup, the RUE retrieves the provider list for the country it is located in.

2. For each provider in the list:

    *   If the RUE does not have credentials for that provider, if requested by the user, use the ProviderConfig path without credentials to obtain signup, dial-around, and help desk information.

    *   If the RUE has credentials for that provider, use the RueConfig path with the locally stored credentials to configure the RUE for that provider.

9.3. OpenAPI Interface Descriptions

The interfaces in Sections 9.1 and 9.2 are formally specified with OpenAPI 3.0 [OpenAPI] descriptions in YAML form.

The OpenAPI description below is normative. If there is any conflict between the text or examples and this section, the OpenAPI description takes precedence.

9.3.1. Provider List

```
openapi: 3.0.1
info:
  title: RUM Provider List API
  version: "1.0"
servers:
  - url: https://localhost/rum/v1
paths:
  /Providers:
    get:
      summary: Get a list of providers and domains to get
              config data from
      operationId: GetProviderList
      responses:
        '200':
          description: List of providers for a country
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ProviderList'
```

```yaml
    /Versions:
      servers:
        - url: https://localhost/rum
          description: Override base path for Versions query
      get:
        summary: Retrieves all supported versions
        operationId: RetrieveVersions
        responses:
          '200':
            description: Versions supported
            content:
              application/json:
                schema:
                  $ref: '#/components/schemas/VersionsArray'
components:
  schemas:
    ProviderList:
      type: object
      required:
        - providers
      properties:
        providers:
          type: array
          items:
            type: object
            required:
              - name
              - providerEntryPoint
            properties:
              name:
                type: string
                description: Human-readable provider name
              providerEntryPoint:
                type: string
                description: Provider entry point for interface
    VersionsArray:
      type: object
      required:
        - versions
      properties:
        versions:
          type: array
          items:
            type: object
            required:
              - major
              - minor
            properties:
              major:
                type: integer
                format: int32
                description: Version major number
              minor:
                type: integer
                format: int32
                description: Version minor number
```

Figure 6: Provider List OpenAPI Description (RueProviderList.yaml)

9.3.2. Configuration

```yaml
openapi: 3.0.1
info:
  title: RUM Configuration API
  version: "1.0"
servers:
  - url: https://localhost/rum/v1
paths:
  /ProviderConfig:
    get:
      summary: Configuration data for one provider
```

```yaml
      operationId: GetProviderConfiguration
      parameters:
        - in: query
          name: apiKey
          schema:
            type: string
          description: API Key assigned to this implementation
        - in: query
          name: instanceId
          schema:
            type: string
          required: true
          description: Unique string for this implementation
                       on this device
      responses:
        '200':
          description: Configuration object
          content:
            application/json:
              schema:
                $ref:
                 '#/components/schemas/ProviderConfigurationData'
  /RueConfig:
    get:
      summary: Configuration data for one RUE
      operationId: GetRueConfiguration
      parameters:
        - in: query
          name: apiKey
          schema:
            type: string
          description: API Key assigned to this implementation
        - in: query
          name: instanceId
          schema:
            type: string
          required: true
          description: Unique string for this implementation
                       on this device
      responses:
        '200':
          description: Configuration object
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/RueConfigurationData'
  /Versions:
    servers:
      - url: https://localhost/rum
        description: Override base path for Versions query
    get:
      summary: Retrieves all supported versions
      operationId: RetrieveVersions
      responses:
        '200':
          description: Versions supported
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/VersionsArray'
components:
  schemas:
    ProviderConfigurationData:
      type: object
      required:
        - dial-around
      properties:
        signup:
          type: array
          items:
            type: object
```

```yaml
              required:
                - language
                - uri
              properties:
                language:
                  type: string
                  description: Entry from IANA "Language Subtag
                    Registry"
                uri:
                  type: string
                  format: uri
                  description: URI to signup website supporting
                    this language
        dial-around:
          type: array
          items:
            type: object
            required:
              - language
              - front-door
              - oneStage
            properties:
              language:
                type: string
                description: Entry from IANA "Language Subtag
                  Registry"
              front-door:
                type: string
                format: uri
                description: SIP URI for two-stage dial-around
              oneStage:
                type: string
                format: uri
                description: SIP URI for one-stage dial-around
        helpDesk:
          type: array
          items:
            type: object
            required:
              - language
              - uri
            properties:
              language:
                type: string
                description: Entry from IANA "Language Subtag
                  Registry"
              uri:
                type: string
                format: uri
                description: SIP URI of help desk supporting language
    RueConfigurationData:
      type: object
      required:
        - phone-number
        - provider-domain
      properties:
        lifetime:
          type: integer
          description: How long (in seconds) the RUE MAY cache the
                       configuration values
        sip-password:
          type: string
        phone-number:
          type: string
          description: Telephone number assigned this subscriber in
            E.164 format
        user-name:
          type: string
          description: A username assigned to this subscriber
        display-name:
          type: string
```

```
      description: Display name for the subscriber
provider-domain:
  type: string
  description: Domain of the provider for this subscriber
outbound-proxies:
  type: array
  items:
     type: string
     format: uri
     description: SIP URI of a proxy to be used when sending
                 requests to the provider
mwi:
  type: string
  format: uri
  description: A URI identifying a SIP event server that
       generates "message-summary" events for this subscriber
videomail:
  type: string
  format: uri
  description: An HTTPS or SIP URI that can be used to
                 retrieve video mail messages
contacts:
  type: object
  description: Server and credentials for contact
       import/export
  required:
     - contacts-uri
  properties:
     contacts-uri:
       type: string
       format: uri
       description: An HTTPS URI that may be used to export
          (retrieve) the subscriber's complete contact list
          managed by the provider
     contacts-username:
       type: string
       description: Username for authentication with the
          CardDAV server.  Use SIP user-name if not provided
     contacts-password:
       type: string
       description: Password for authentication. Use provider
          sip-password if not provided
carddav:
  type: object
  description: CardDAV server and user information that can
        be used to synchronize the RUE's contact list with
        the contact list managed by the provider
  required:
     - carddav-domain
  properties:
     carddav-domain:
       type: string
       description: CardDAV server address
     carddav-username:
       type: string
       description: Username for authentication with the
           CardDAV server.  Use SIP user-name if not provided
     carddav-password:
       type: string
       description: Password for authentication to the CardDAV
          server. Use provider sip-password if not provided
sendLocationWithRegistration:
  type: boolean
  description: True if the RUE should send a Geolocation
        header field with REGISTER; false if it should not.
        Defaults to false if not present
ice-servers:
  type: array
  items:
    type: object
    required:
```

```
                      - server-type
                      - uri
                  properties:
                    server-type:
                      type: string
                      description: Server type ("stun" or "turn")
                    uri:
                      type: string
                      format: uri
                      description: URIs identifying STUN and TURN servers
                        available for use by the RUE for establishing
                        media streams in calls via the provider
          VersionsArray:
            type: object
            required:
              - versions
            properties:
              versions:
                type: array
                items:
                  type: object
                  required:
                    - major
                    - minor
                  properties:
                    major:
                      type: integer
                      format: int32
                      description: Version major number
                    minor:
                      type: integer
                      format: int32
                      description: Version minor number
```

       Figure 7: Configuration OpenAPI Description (RueConfiguration.yaml)

10. IANA Considerations

10.1. RUE Provider List Registry

   IANA has created the "RUE Provider List" registry. The registration
   policy is "Expert Review" [RFC8126]. A regulator operated or
   designated list interface operator is preferred. Otherwise, evidence
   that the proposed list interface operator will provide a complete
   list of providers is required to add the entry to the registry.
   Updates to the registry are permitted if the expert determines that
   the proposed URI provides a more accurate list than the existing
   entry. Each entry has two fields; values for both MUST be provided
   when registering or updating an entry:

   * country code: a two-letter ISO93166 country code

   * list entry point: a string is used to compose the URI to the
      provider list interface for that country

10.2. Registration of Rue-Owner Value of the Purpose Parameter

   This document defines the new predefined value "rue-owner" for the
   "purpose" header field parameter of the Call-Info header field. The
   use for rue-owner is defined in Section 5.2.3. IANA has added a
   reference to this document in the "Header Field Parameters and
   Parameter Values" subregistry of the "Session Initiation Protocol
   (SIP) Parameters" for the header field "Call-Info" and parameter name
   "purpose".

   Header Field: Call-Info

   Parameter Name: purpose

   Predefined Values: Yes

11. Security Considerations

The RUE is required to communicate with servers on public IP
addresses and specific ports to perform its required functions. If
it is necessary for the RUE to function on a corporate or other
network that operates a default-deny firewall between the RUE and
these services, the user must arrange with their network manager for
passage of traffic through such a firewall in accordance with the
protocols and associated SRV records as exposed by the provider.
Because VRS providers may use different ports for different services,
these port numbers may differ from provider to provider.

This document requires implementation and use of a number of other
specifications in order to fulfill the RUE profile; the security
considerations described in those documents apply accordingly to the
RUE interactions.

When a CA participates in a conversation, they have access to the
content of the conversation even though it is nominally a
conversation between the two endpoints. There is an expectation that
the CA will keep the communication contents in confidence. This is
usually defined by contractual or legal requirements.

Since different providers (within a given country) may have different
policies, RUE implementations MUST include a user interaction step to
select from available providers before proceeding to actually
register with any given provider.

12. Normative References

[OpenAPI] Miller, D., Whitlock, J., Gardiner, M., Ralphson, M.,
           Ratovsky, R., and U. Sarid, "OpenAPI Specification
           v3.0.1", December 2017,
           <https://spec.openapis.org/oas/v3.0.1>.

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

[RFC2392] Levinson, E., "Content-ID and Message-ID Uniform Resource
           Locators", RFC 2392, DOI 10.17487/RFC2392, August 1998,
           <https://www.rfc-editor.org/info/rfc2392>.

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

[RFC3263] Rosenberg, J. and H. Schulzrinne, "Session Initiation
           Protocol (SIP): Locating SIP Servers", RFC 3263,
           DOI 10.17487/RFC3263, June 2002,
           <https://www.rfc-editor.org/info/rfc3263>.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model
           with Session Description Protocol (SDP)", RFC 3264,
           DOI 10.17487/RFC3264, June 2002,
           <https://www.rfc-editor.org/info/rfc3264>.

[RFC3311] Rosenberg, J., "The Session Initiation Protocol (SIP)
           UPDATE Method", RFC 3311, DOI 10.17487/RFC3311, October
           2002, <https://www.rfc-editor.org/info/rfc3311>.

[RFC3323] Peterson, J., "A Privacy Mechanism for the Session
           Initiation Protocol (SIP)", RFC 3323,
           DOI 10.17487/RFC3323, November 2002,
           <https://www.rfc-editor.org/info/rfc3323>.

[RFC3326] Schulzrinne, H., Oran, D., and G. Camarillo, "The Reason
           Header Field for the Session Initiation Protocol (SIP)",

```
              RFC 3326, DOI 10.17487/RFC3326, December 2002,
              <https://www.rfc-editor.org/info/rfc3326>.

   [RFC3327]  Willis, D. and B. Hoeneisen, "Session Initiation Protocol
              (SIP) Extension Header Field for Registering Non-Adjacent
              Contacts", RFC 3327, DOI 10.17487/RFC3327, December 2002,
              <https://www.rfc-editor.org/info/rfc3327>.

   [RFC3458]  Burger, E., Candell, E., Eliot, C., and G. Klyne, "Message
              Context for Internet Mail", RFC 3458,
              DOI 10.17487/RFC3458, January 2003,
              <https://www.rfc-editor.org/info/rfc3458>.

   [RFC3515]  Sparks, R., "The Session Initiation Protocol (SIP) Refer
              Method", RFC 3515, DOI 10.17487/RFC3515, April 2003,
              <https://www.rfc-editor.org/info/rfc3515>.

   [RFC3605]  Huitema, C., "Real Time Control Protocol (RTCP) attribute
              in Session Description Protocol (SDP)", RFC 3605,
              DOI 10.17487/RFC3605, October 2003,
              <https://www.rfc-editor.org/info/rfc3605>.

   [RFC3840]  Rosenberg, J., Schulzrinne, H., and P. Kyzivat,
              "Indicating User Agent Capabilities in the Session
              Initiation Protocol (SIP)", RFC 3840,
              DOI 10.17487/RFC3840, August 2004,
              <https://www.rfc-editor.org/info/rfc3840>.

   [RFC3842]  Mahy, R., "A Message Summary and Message Waiting
              Indication Event Package for the Session Initiation
              Protocol (SIP)", RFC 3842, DOI 10.17487/RFC3842, August
              2004, <https://www.rfc-editor.org/info/rfc3842>.

   [RFC3891]  Mahy, R., Biggs, B., and R. Dean, "The Session Initiation
              Protocol (SIP) "Replaces" Header", RFC 3891,
              DOI 10.17487/RFC3891, September 2004,
              <https://www.rfc-editor.org/info/rfc3891>.

   [RFC3892]  Sparks, R., "The Session Initiation Protocol (SIP)
              Referred-By Mechanism", RFC 3892, DOI 10.17487/RFC3892,
              September 2004, <https://www.rfc-editor.org/info/rfc3892>.

   [RFC3960]  Camarillo, G. and H. Schulzrinne, "Early Media and Ringing
              Tone Generation in the Session Initiation Protocol (SIP)",
              RFC 3960, DOI 10.17487/RFC3960, December 2004,
              <https://www.rfc-editor.org/info/rfc3960>.

   [RFC3966]  Schulzrinne, H., "The tel URI for Telephone Numbers",
              RFC 3966, DOI 10.17487/RFC3966, December 2004,
              <https://www.rfc-editor.org/info/rfc3966>.

   [RFC4102]  Jones, P., "Registration of the text/red MIME Sub-Type",
              RFC 4102, DOI 10.17487/RFC4102, June 2005,
              <https://www.rfc-editor.org/info/rfc4102>.

   [RFC4103]  Hellstrom, G. and P. Jones, "RTP Payload for Text
              Conversation", RFC 4103, DOI 10.17487/RFC4103, June 2005,
              <https://www.rfc-editor.org/info/rfc4103>.

   [RFC4488]  Levin, O., "Suppression of Session Initiation Protocol
              (SIP) REFER Method Implicit Subscription", RFC 4488,
              DOI 10.17487/RFC4488, May 2006,
              <https://www.rfc-editor.org/info/rfc4488>.

   [RFC4585]  Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey,
              "Extended RTP Profile for Real-time Transport Control
              Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585,
              DOI 10.17487/RFC4585, July 2006,
              <https://www.rfc-editor.org/info/rfc4585>.

   [RFC4733]  Schulzrinne, H. and T. Taylor, "RTP Payload for DTMF
```

                    Digits, Telephony Tones, and Telephony Signals", RFC 4733,
                    DOI 10.17487/RFC4733, December 2006,
                    <https://www.rfc-editor.org/info/rfc4733>.

   [RFC4967]   Rosen, B., "Dial String Parameter for the Session
                    Initiation Protocol Uniform Resource Identifier",
                    RFC 4967, DOI 10.17487/RFC4967, July 2007,
                    <https://www.rfc-editor.org/info/rfc4967>.

   [RFC5104]   Wenger, S., Chandra, U., Westerlund, M., and B. Burman,
                    "Codec Control Messages in the RTP Audio-Visual Profile
                    with Feedback (AVPF)", RFC 5104, DOI 10.17487/RFC5104,
                    February 2008, <https://www.rfc-editor.org/info/rfc5104>.

   [RFC5168]   Levin, O., Even, R., and P. Hagendorf, "XML Schema for
                    Media Control", RFC 5168, DOI 10.17487/RFC5168, March
                    2008, <https://www.rfc-editor.org/info/rfc5168>.

   [RFC5393]   Sparks, R., Ed., Lawrence, S., Hawrylyshen, A., and B.
                    Campen, "Addressing an Amplification Vulnerability in
                    Session Initiation Protocol (SIP) Forking Proxies",
                    RFC 5393, DOI 10.17487/RFC5393, December 2008,
                    <https://www.rfc-editor.org/info/rfc5393>.

   [RFC5626]   Jennings, C., Ed., Mahy, R., Ed., and F. Audet, Ed.,
                    "Managing Client-Initiated Connections in the Session
                    Initiation Protocol (SIP)", RFC 5626,
                    DOI 10.17487/RFC5626, October 2009,
                    <https://www.rfc-editor.org/info/rfc5626>.

   [RFC5658]   Froment, T., Lebel, C., and B. Bonnaerens, "Addressing
                    Record-Route Issues in the Session Initiation Protocol
                    (SIP)", RFC 5658, DOI 10.17487/RFC5658, October 2009,
                    <https://www.rfc-editor.org/info/rfc5658>.

   [RFC5954]   Gurbani, V., Ed., Carpenter, B., Ed., and B. Tate, Ed.,
                    "Essential Correction for IPv6 ABNF and URI Comparison in
                    RFC 3261", RFC 5954, DOI 10.17487/RFC5954, August 2010,
                    <https://www.rfc-editor.org/info/rfc5954>.

   [RFC6263]   Marjou, X. and A. Sollaud, "Application Mechanism for
                    Keeping Alive the NAT Mappings Associated with RTP / RTP
                    Control Protocol (RTCP) Flows", RFC 6263,
                    DOI 10.17487/RFC6263, June 2011,
                    <https://www.rfc-editor.org/info/rfc6263>.

   [RFC6351]   Perreault, S., "xCard: vCard XML Representation",
                    RFC 6351, DOI 10.17487/RFC6351, August 2011,
                    <https://www.rfc-editor.org/info/rfc6351>.

   [RFC6352]   Daboo, C., "CardDAV: vCard Extensions to Web Distributed
                    Authoring and Versioning (WebDAV)", RFC 6352,
                    DOI 10.17487/RFC6352, August 2011,
                    <https://www.rfc-editor.org/info/rfc6352>.

   [RFC6442]   Polk, J., Rosen, B., and J. Peterson, "Location Conveyance
                    for the Session Initiation Protocol", RFC 6442,
                    DOI 10.17487/RFC6442, December 2011,
                    <https://www.rfc-editor.org/info/rfc6442>.

   [RFC6665]   Roach, A.B., "SIP-Specific Event Notification", RFC 6665,
                    DOI 10.17487/RFC6665, July 2012,
                    <https://www.rfc-editor.org/info/rfc6665>.

   [RFC6764]   Daboo, C., "Locating Services for Calendaring Extensions
                    to WebDAV (CalDAV) and vCard Extensions to WebDAV
                    (CardDAV)", RFC 6764, DOI 10.17487/RFC6764, February 2013,
                    <https://www.rfc-editor.org/info/rfc6764>.

   [RFC6881]   Rosen, B. and J. Polk, "Best Current Practice for
                    Communications Services in Support of Emergency Calling",

                 BCP 181, RFC 6881, DOI 10.17487/RFC6881, March 2013,
                 <https://www.rfc-editor.org/info/rfc6881>.

[RFC7525]   Sheffer, Y., Holz, R., and P. Saint-Andre,
             "Recommendations for Secure Use of Transport Layer
             Security (TLS) and Datagram Transport Layer Security
             (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May
             2015, <https://www.rfc-editor.org/info/rfc7525>.

[RFC7647]   Sparks, R. and A.B. Roach, "Clarifications for the Use of
             REFER with RFC 6665", RFC 7647, DOI 10.17487/RFC7647,
             September 2015, <https://www.rfc-editor.org/info/rfc7647>.

[RFC7742]   Roach, A.B., "WebRTC Video Processing and Codec
             Requirements", RFC 7742, DOI 10.17487/RFC7742, March 2016,
             <https://www.rfc-editor.org/info/rfc7742>.

[RFC7852]   Gellens, R., Rosen, B., Tschofenig, H., Marshall, R., and
             J. Winterbottom, "Additional Data Related to an Emergency
             Call", RFC 7852, DOI 10.17487/RFC7852, July 2016,
             <https://www.rfc-editor.org/info/rfc7852>.

[RFC7874]   Valin, JM. and C. Bran, "WebRTC Audio Codec and Processing
             Requirements", RFC 7874, DOI 10.17487/RFC7874, May 2016,
             <https://www.rfc-editor.org/info/rfc7874>.

[RFC8174]   Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
             2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
             May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8445]   Keranen, A., Holmberg, C., and J. Rosenberg, "Interactive
             Connectivity Establishment (ICE): A Protocol for Network
             Address Translator (NAT) Traversal", RFC 8445,
             DOI 10.17487/RFC8445, July 2018,
             <https://www.rfc-editor.org/info/rfc8445>.

[RFC8446]   Rescorla, E., "The Transport Layer Security (TLS) Protocol
             Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018,
             <https://www.rfc-editor.org/info/rfc8446>.

[RFC8599]   Holmberg, C. and M. Arnold, "Push Notification with the
             Session Initiation Protocol (SIP)", RFC 8599,
             DOI 10.17487/RFC8599, May 2019,
             <https://www.rfc-editor.org/info/rfc8599>.

[RFC8760]   Shekh-Yusef, R., "The Session Initiation Protocol (SIP)
             Digest Access Authentication Scheme", RFC 8760,
             DOI 10.17487/RFC8760, March 2020,
             <https://www.rfc-editor.org/info/rfc8760>.

[RFC8825]   Alvestrand, H., "Overview: Real-Time Protocols for
             Browser-Based Applications", RFC 8825,
             DOI 10.17487/RFC8825, January 2021,
             <https://www.rfc-editor.org/info/rfc8825>.

[RFC8827]   Rescorla, E., "WebRTC Security Architecture", RFC 8827,
             DOI 10.17487/RFC8827, January 2021,
             <https://www.rfc-editor.org/info/rfc8827>.

[RFC8829]   Uberti, J., Jennings, C., and E. Rescorla, Ed.,
             "JavaScript Session Establishment Protocol (JSEP)",
             RFC 8829, DOI 10.17487/RFC8829, January 2021,
             <https://www.rfc-editor.org/info/rfc8829>.

[RFC8834]   Perkins, C., Westerlund, M., and J. Ott, "Media Transport
             and Use of RTP in WebRTC", RFC 8834, DOI 10.17487/RFC8834,
             January 2021, <https://www.rfc-editor.org/info/rfc8834>.

[RFC8835]   Alvestrand, H., "Transports for WebRTC", RFC 8835,
             DOI 10.17487/RFC8835, January 2021,
             <https://www.rfc-editor.org/info/rfc8835>.

   [RFC8839] Petit-Huguenin, M., Nandakumar, S., Holmberg, C., Keränen,
              A., and R. Shpount, "Session Description Protocol (SDP)
              Offer/Answer Procedures for Interactive Connectivity
              Establishment (ICE)", RFC 8839, DOI 10.17487/RFC8839,
              January 2021, <https://www.rfc-editor.org/info/rfc8839>.

   [RFC8865] Holmberg, C. and G. Hellström, "T.140 Real-Time Text
              Conversation over WebRTC Data Channels", RFC 8865,
              DOI 10.17487/RFC8865, January 2021,
              <https://www.rfc-editor.org/info/rfc8865>.

   [RFC8866] Begen, A., Kyzivat, P., Perkins, C., and M. Handley, "SDP:
              Session Description Protocol", RFC 8866,
              DOI 10.17487/RFC8866, January 2021,
              <https://www.rfc-editor.org/info/rfc8866>.

   [RFC9071] Hellström, G., "RTP-Mixer Formatting of Multiparty Real-
              Time Text", RFC 9071, DOI 10.17487/RFC9071, July 2021,
              <https://www.rfc-editor.org/info/rfc9071>.

   [RFC9110] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke,
              Ed., "HTTP Semantics", STD 97, RFC 9110,
              DOI 10.17487/RFC9110, June 2022,
              <https://www.rfc-editor.org/info/rfc9110>.

   [RFC9112] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke,
              Ed., "HTTP/1.1", STD 99, RFC 9112, DOI 10.17487/RFC9112,
              June 2022, <https://www.rfc-editor.org/info/rfc9112>.

13. Informative References

   [RFC3665] Johnston, A., Donovan, S., Sparks, R., Cunningham, C., and
              K. Summers, "Session Initiation Protocol (SIP) Basic Call
              Flow Examples", BCP 75, RFC 3665, DOI 10.17487/RFC3665,
              December 2003, <https://www.rfc-editor.org/info/rfc3665>.

   [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for
              Writing an IANA Considerations Section in RFCs", BCP 26,
              RFC 8126, DOI 10.17487/RFC8126, June 2017,
              <https://www.rfc-editor.org/info/rfc8126>.

Acknowledgements

Author's Address

   Brian Rosen
   470 Conrad Dr
   Mars, PA 16046
   United States of America
   Email: br@brianrosen.net