          Web Real-Time Communication Use-cases and Requirements
           draft-ietf-rtcweb-use-cases-and-requirements-10.txt

Abstract

   This document describes web based real-time communication use-cases.
   Based on the use-cases, the document also derives requirements
   related to the browser, and the API used by web applications to
   request and control media stream and data exchange services provided
   by the browser.

Status of this Memo

Copyright Notice

described in the Simplified BSD License.

Table of Contents

1. Introduction

This document presents a few use-cases of web applications that are
executed in a browser and use real-time communication capabilities.
Based on the use-cases, the document derives requirements related to
the browser and the API used by web applications in the browser.

The requirements related to the browser are named "Fn" and are
described in Section 5.2

The requirements related to the API are named "An" and are described
in Section 5.3

The document focuses on requirements related to real-time media
streams and data exchange. Requirements related to privacy,
signalling between the browser and web server etc. are currently not
considered.

2. Conventions

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

3. Definitions

TBD

4. Use-cases

4.1. Introduction

This section describes web based real-time communication use-cases,
from which requirements are derived.

The following considerations are applicable to all use cases:
o Clients can be on IPv4-only
o Clients can be on IPv6-only
o Clients can be on dual-stack
o Clients can be on wideband (10s of Mbits/sec)
o Clients can be on narrowband (10s to 100s of Kbits/sec)
o Clients can be on variable-media-quality networks (wireless)

o Clients can be on congested networks

o Clients can be on firewalled networks with no UDP allowed

o Clients can be on networks with any type (as described in [RFC4787])
   of NAT.

4.2. Browser-to-browser use-cases

4.2.1. Simple Video Communication Service

4.2.1.1. Description

Two or more users have loaded a video communication web application
into their browsers, provided by the same service provider, and
logged into the service it provides. The web service publishes
information about user login status by pushing updates to the web
application in the browsers. When one online user selects a peer
online user, a 1-1 audiovisual communication session between the
browsers of the two peers is initiated. The invited user might
accept or reject the session.

During session establishment a self-view is displayed, and once the
session has been established the video sent from the remote peer is
displayed in addition to the self-view. During the session, each
user can select to remove and re-insert the self-view as often as
desired. Each user can also change the sizes of his/her two video
displays during the session. Each user can also pause sending of
media (audio, video, or both) and mute incoming media

It is essential that the communication cannot be wiretapped
[RFC2804].

The users are provided wiht means that allow them to (through a
separate, trusted communication channel) verify that the media
origins from the other user and has not been manipulated.

The user's browsers will reject all incoming media that has been
created, injected or in any way modified by any entity not trusted by
the service provider.

The application gives the users the opportunity to stop it from
exposing the IP address to the application of the other user.

Any session participant can end the session at any time.

The two users may be using communication devices of different makes,
with different operating systems and browsers from different vendors.

One user has an unreliable Internet connection. It sometimes loses

packets, and sometimes goes down completely.

One user is located behind a Network Address Translator (NAT).

The web service monitors the quality of the service (focus on quality of audio and video) the end-users experience.

4.2.1.2. Derived Requirements

F1, F2, F3, F4, F5, F6, F8, F9, F10, F20, F25, F28, F35, F36, F38

A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A25, A26

4.2.2. Simple Video Communication Service, NAT/FW that blocks UDP

4.2.2.1. Description

This use-case is almost identical to the Simple Video Communication Service use-case (Section 4.2.1). The difference is that one of the users is behind a NAT that blocks UDP traffic.

4.2.2.2. Derived Requirements

F1, F2, F3, F4, F5, F6, F8, F9, F10, F20, F25, F28, F29

A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12

4.2.3. Simple Video Communication Service, FW that only allows http

4.2.3.1. Description

This use-case is almost identical to the Simple Video Communication Service use-case (Section 4.2.1). The difference is that one of the users is behind a FW that only allows http traffic.

Note: What about WS? Could it be a viable back-off mechanism?

4.2.3.2. Derived Requirements

F1, F2, F3, F4, F5, F6, F8, F9, F10, F20, F25, F28, F37

A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12

4.2.4. Simple Video Communication Service, global service provider

[4.2.4.1](). Description

This use-case is almost identical to the Simple Video Communication Service use-case ([Section 4.2.1]()).

What is added is that the service provider is operating over large geographical areas (or even globally).

Assuming that ICE will be used, this means that the service provider would like to be able to provide several STUN and TURN servers (via the app) to the browser; selection of which one(s) to use is part of the ICE processing. Other reasons for wanting to provide several STUN and TURN servers include support for IPv4 and IPv6, load balancing and redundancy.

[4.2.4.2](). Derived Requirements

F1, F2, F3, F4, F5, F6, F8, F9, F10, F20, F25, F28, F31

A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A22

[4.2.5](). Simple Video Communication Service, enterprise aspects

[4.2.5.1](). Description

This use-case is similar to the Simple Video Communication Service use-case ([Section 4.2.1]()).

What is added is aspects when using the service in enterprises. ICE is assumed in the further description of this use-case.

An enterprise that uses a RTCWEB based web application for communication desires to audit all RTCWEB based application session used from inside the company towards any external peer. To be able to do this they deploy a TURN server that straddle the boundary between the internal network and the external.

The firewall will block all attempts to use STUN with an external destination unless they go to the enterprise auditing TURN server. In cases where employees are using RTCWEB applications provided by an external service provider they still want to have the traffic to stay inside their internal network and in addition not load the straddling TURN server, thus they deploy a STUN server allowing the RTCWEB client to determine its server reflexive address on the internal side. Thus enabling cases where peers are both on the internal side to connect without the traffic leaving the internal network. It must be possibele to configure the browsers used in the enterprise with network specific STUN and TURN servers. This should be possible to

achieve by autoconfiguration methods. The RTCWEB functionality will
need to utilize both network specific STUN and TURN resources and
STUN and TURN servers provisioned by the web application.

4.2.5.2. Derived Requirements

F1, F2, F3, F4, F5, F6, F8, F9, F10, F20, F25, F28, F32

A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12

4.2.6. Simple Video Communication Service, access change

4.2.6.1. Description

This use-case is almost identical to the Simple Video Communication
Service use-case (Section 4.2.1).The difference is that the user
changes network access during the session:

The communication device used by one of the users have several
network adapters (Ethernet, WiFi, Cellular). The communication
device is accessing the Internet using Ethernet, but the user has to
start a trip during the session. The communication device
automatically changes to use WiFi when the Ethernet cable is removed
and then moves to cellular access to the Internet when moving out of
WiFi coverage. The session continues even though the access method
changes.

4.2.6.2. Derived Requirements

F1, F2, F3, F4, F5, F6, F8, F9, F10, F20, F25, F26, F28

A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12

4.2.7. Simple Video Communication Service, QoS

4.2.7.1. Description

This use-case is almost identical to the Simple Video Communication
Service, access change use-case (Section 4.2.6). The use of Quality
of Service (QoS) capabilities is added:

The user in the previous use case that starts a trip is behind a
common residential router that supports prioritization of traffic.
In addition, the user's provider of cellular access has QoS support
enabled. The user is able to take advantage of the QoS support both
when accessing via the residential router and when using cellular.

4.2.7.2. Derived Requirements

F1, F2, F3, F4, F5, F6, F8, F9, F10, F20, F24, F25, F26, F28

A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12

4.2.8. Simple Video Communication Service with sharing

4.2.8.1. Description

This use-case has the audio and video communication of the Simple
Video Communication Service use-case (Section 4.2.1).

But in addition to this, one of the users can share what is being
displayed on her/his screen with a peer. The user can choose to
share the entire screen, part of the screen (part selected by the
user) or what a selected applicaton displays with the peer.

4.2.8.2. Derived Requirements

F1, F2, F3, F4, F5, F6, F8, F9, F10, F20, F25, F28, F30

A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A21

4.2.9. Simple Video Communication Service with file exchange

4.2.9.1. Description

This use-case has the audio and video communication of the Simple
Video Communication Service use-case (Section 4.2.1).

But in addition to this, the users can send and receive files stored
in the file system of the device used.

4.2.9.2. Derived Requirements

F1, F2, F3, F4, F5, F6, F8, F9, F10, F20, F25, F28, F30, F33

A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A21, A24

4.2.10. Simple video communication service with inter-operator calling

4.2.10.1. Description

Two users have logged into two different web applications, provided
by different service providers.

The service providers are interconnected by some means, but exchange

no more information about the users than what can be carried using
SIP.

NOTE: More profiling of what this means may be needed.

For each user Alice who has authorized another user Bob to receive
login status information, Alice's service publishes Alice's login
status information to Bob. How this authorization is defined and
established is out of scope.

The same functionality as in the the Simple Video Communication
Service use-case (Section 4.2.1) is available.

The same issues with connectivity apply.

4.2.10.2. Derived requirements

F1, F2, F3, F4, F5, F6, F8, F9, F10, F20, F25, F27, F28

A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A20

4.2.11. Hockey Game Viewer

4.2.11.1. Description

An ice-hockey club uses an application that enables talent scouts to,
in real-time, show and discuss games and players with the club
manager. The talent scouts use a mobile phone with two cameras, one
front facing and one rear facing.

The club manager uses a desktop, equipped with one camera, for
viewing the game and discussing with the talent scout.

Before the game starts, and during game breaks, the talent scout and
the manager have a 1-1 audiovisual communication session. Only the
rear facing camera of the mobile phone is used. On the display of
the mobile phone, the video of the club manager is shown with a
picture-in-picture thumbnail of the rear facing camera (self-view).
On the display of the desktop, the video of the talent scout is shown
with a picture-in-picture thumbnail of the desktop camera (self-
view).

When the game is on-going, the talent scout activates the use of the
front facing camera, and that stream is sent to the desktop (the
stream from the rear facing camera continues to be sent all the
time). The video stream captured by the front facing camera (that is
capturing the game) of the mobile phone is shown in a big window on
the desktop screen, with picture-in-picture thumbnails of the rear

facing camera and the desktop camera (self-view). On the display of
the mobile phone the game is shown (front facing camera) with
picture-in-picture thumbnails of the rear facing camera (self-view)
and the desktop camera. As the most important stream in this phase
is the video showing the game, the application used in the talent
scout's mobile sets higher priority for that stream.

It is essential that the communication cannot be wiretapped
[RFC2804].

4.2.11.2. Derived Requirements

F1, F2, F3, F4, F5, F6, F8, F9, F10, F17, F20, F34

A1, A2, A3, A4, A5, A7, A8, A9, A10, A11, A12, A17, A23

4.2.12. Multiparty video communication

4.2.12.1. Description

In this use-case is the Simple Video Communication Service use-case
(Section 4.2.1) is extended by allowing multiparty sessions. No
central server is involved - the browser of each participant sends
and receives streams to and from all other session participants. The
web application in the browser of each user is responsible for
setting up streams to all receivers.

In order to enhance intelligibility, the web application pans the
audio from different participants differently when rendering the
audio. This is done automatically, but users can change how the
different participants are placed in the (virtual) room. In addition
the levels in the audio signals are adjusted before mixing.

Another feature intended to enhance the use experience is that the
video window that displays the video of the currently speaking peer
is highlighted.

Each video stream received is by default displayed in a thumbnail
frame within the browser, but users can change the display size.

It is essential that the communication cannot be wiretapped
[RFC2804].

Note: What this use-case adds in terms of requirements is
capabilities to send streams to and receive streams from several
peers concurrently, as well as the capabilities to render the video
from all recevied streams and be able to spatialize, level adjust and
mix the audio from all received streams locally in the browser. It

also adds the capability to measure the audio level/activity.

4.2.12.2. Derived Requirements

F1, F2, F3, F4, F5, F6, F8, F9, F10, F11, F12, F13, F14, F15, F16,
F17, F20, F25

A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A13, A14, A15,
A16, A17

4.2.13. Multiparty on-line game with voice communication

4.2.13.1. Description

This use case is based on the previous one. In this use-case, the
voice part of the multiparty video communication use case is used in
the context of an on-line game. The received voice audio media is
rendered together with game sound objects. For example, the sound of
a tank moving from left to right over the screen must be rendered and
played to the user together with the voice media.

Quick updates of the game state is required, and have higher priority
than the voice.

It is essential that the communication cannot be wiretapped
[RFC2804].

Note: the difference regarding local audio processing compared to the
"Multiparty video communication" use-case is that other sound objects
than the streams must be possible to be included in the
spatialization and mixing. "Other sound objects" could for example
be a file with the sound of the tank; that file could be stored
locally or remotely.

4.2.13.2. Derived Requirements

F1, F2, F3, F4, F5, F6, F8, F9, F11, F12, F13, F14, F15, F16, F18,
F20, F23, F34

A1, A2, A3, A4, A5, A7, A8, A9, A10, A11, A12, A13, A14, A15, A16,
A17, A18, A23

4.2.14. Distributed Music Band

4.2.14.1. Description

In this use-case, a music band is playing music while the members are
at different physical locations. No central server is used, instead

all streams are set up in a mesh fashion.

Discussion: This use-case was briefly discussed at the Quebec webrtc
meeting and it got support. So far the only concrete requirement
(A17) derived is that the application must be able to ask the browser
to treat the audio signal as audio (in contrast to speech). However,
the use case should be further analysed to determine other
requirements (could be e.g. on delay mic->speaker, level control of
audio signals, etc.).

4.2.14.2. Derived Requirements

F1, F2, F3, F4, F5, F6, F8, F9, F11, F12, F13, F14, F15, F16

A1, A2, A3, A4, A5, A7, A8, A9, A10, A11, A12, A13, A14, A15, A16,
A19

4.3. Browser - GW/Server use cases

4.3.1. Telephony terminal

4.3.1.1. Description

A mobile telephony operator allows its customers to use a web browser
to access their services. After a simple log in the user can place
and receive calls in the same way as when using a normal mobile
phone. When a call is received or placed, the identity is shown in
the same manner as when a mobile phone is used.

It is essential that the communication cannot be wiretapped
[RFC2804].

Note: With "place and receive calls in the same way as when using a
normal mobile phone" it is meant that you can dial a number, and that
your mobile telephony operator has made available your phone contacts
on line, so they are available and can be clicked to call, and be
used to present the identity of an incoming call. If the callee is
not in your phone contacts the number is displayed. Furthermore,
your call logs are available, and updated with the calls made/
received from the browser. And for people receiving calls made from
the web browser the usual identity (i.e. the phone number of the
mobile phone) will be presented.

4.3.1.2. Derived Requirements

F1, F2, F3, F4, F5, F6, F8, F9, F10, F20, F21

A1, A2, A3, A4, A7, A8, A9, A10, A11, A12

4.3.2. Fedex Call

4.3.2.1. Description

   Alice uses her web browser with a service something like Skype to be
   able to phone PSTN numbers. Alice calls 1-800-gofedex. Alice should
   be able to hear the initial prompts from the fedex IVR and when the
   IVR says press 1, there should be a way for Alice to navigate the
   IVR.

4.3.2.2. Derived Requirements

   F1, F2, F3, F4, F5, F6, F8, F9, F10, F21, F22

   A1, A2, A3, A4, A7, A8, A9, A10, A11, A12

4.3.3. Video conferencing system with central server

4.3.3.1. Description

   An organization uses a video communication system that supports the
   establishment of multiparty video sessions using a central conference
   server.

   The browser of each participant send an audio stream (type in terms
   of mono, stereo, 5.1, ... depending on the equipment of the
   participant) to the central server. The central server mixes the
   audio streams (and can in the mixing process naturally add effects
   such as spatialization) and sends towards each participant a mixed
   audio stream which is played to the user.

   The browser of each participant sends video towards the server. For
   each participant one high resolution video is displayed in a large
   window, while a number of low resolution videos are displayed in
   smaller windows. The server selects what video streams to be
   forwarded as main- and thumbnail videos respectively, based on speech
   activity. As the video streams to display can change quite
   frequently (as the conversation flows) it is important that the delay
   from when a video stream is selected for display until the video can
   be displayed is short.

   The organization has an internal network set up with an aggressive
   firewall handling access to the Internet. If users cannot physically
   access the internal network, they can establish a Virtual Private
   Network (VPN).

   It is essential that the communication cannot be wiretapped
   [RFC2804].

   All participants are authenticated by the central server, and
   authorized to connect to the central server. The participants are
   identified to each other by the central server, and the participants
   do not have access to each others' credentials such as e-mail
   addresses or login IDs.

   Note: This use-case adds requirements on support for fast stream
   switches F7, on encryption of media and on ability to traverse very
   restrictive FWs. There exist several solutions that enable the
   server to forward one high resolution and several low resolution
   video streams: a) each browser could send a high resolution, but
   scalable stream, and the server could send just the base layer for
   the low resolution streams, b) each browser could in a simulcast
   fashion send one high resolution and one low resolution stream, and
   the server just selects or c) each browser sends just a high
   resolution stream, the server transcodes into low resolution streams
   as required.

4.3.3.2. Derived Requirements

   F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F17, F19, F20

   A1, A2, A3, A4, A5, A7, A8, A9, A10, A11, A12, A17

5. Requirements

5.1. General

   This section contains the requirements derived from the use-cases in
   section 4.

   NOTE: It is assumed that the user applications are executed on a
   browser. Whether the capabilities to implement specific browser
   requirements are implemented by the browser application, or are
   provided to the browser application by the underlying operating
   system, is outside the scope of this document.

5.2. Browser requirements

REQ-ID	DESCRIPTION
F1	The browser MUST be able to use microphones and cameras as input devices to generate streams.
F2	The browser MUST be able to send streams and data to a peer in the presence of NATs.

F3   Transmitted streams and data MUST be rate
                    controlled.

F4   The browser MUST be able to receive, process and
                    render streams and data ("render" does not
                    apply for data) from peers.

F5   The browser MUST be able to render good quality
                    audio and video even in the presence of
                    reasonable levels of jitter and packet losses.

                    TBD: What is a reasonable level?

F6   The browser MUST be able to handle high loss and
                    jitter levels in a graceful way.

F7   The browser MUST support fast stream switches.

F8   The browser MUST detect when a stream from a
                    peer is not received anymore

F9   When there are both incoming and outgoing audio
                    streams, echo cancellation MUST be made
                    available to avoid disturbing echo during
                    conversation.

                    QUESTION: How much control should be left to the
                    web application?

F10   The browser MUST support synchronization of
                    audio and video.

                    QUESTION: How much control should be left to the
                    web application?

F11   The browser MUST be able to transmit streams and
                    data to several peers concurrently.

F12   The browser MUST be able to receive streams and
                    data from multiple peers concurrently.

F13   The browser MUST be able to apply spatialization
                    effects to audio streams.

F14   The browser MUST be able to measure the level
                    in audio streams.

```
-------------------------------------------------------------------
F15            The browser MUST be able to change the level
               in audio streams.
-------------------------------------------------------------------
F16            The browser MUST be able to render several
               concurrent video streams
-------------------------------------------------------------------
F17            The browser MUST be able to mix several
               audio streams.
-------------------------------------------------------------------
F18            The browser MUST be able to process and mix
               sound objects (media that is retrieved from
               another source than the established media
               stream(s) with the peer(s) with audio streams.
-------------------------------------------------------------------
F19            Streams and data MUST be able to pass through
               restrictive firewalls.
-------------------------------------------------------------------
F20            It MUST be possible to protect streams and data
               from wiretapping.
-------------------------------------------------------------------
F21            The browser MUST support an audio media format
               (codec) that is commonly supported by existing
               telephony services.

               QUESTION: G.711?
-------------------------------------------------------------------
F22            There should be a way to navigate
               a DTMF based IVR
-------------------------------------------------------------------
F23            The browser must be able to send short
               latency unreliable datagram traffic to a
               peer browser.
-------------------------------------------------------------------
F24            The browser SHOULD be able to take advantage
               of available capabilities (supplied by network
               nodes) to prioritize voice, video and data
               appropriately.
-------------------------------------------------------------------
F25            The browser SHOULD use encoding of streams
               suitable for the current rendering (e.g.
               video display size) and SHOULD change parameters
               if the rendering changes during the session
-------------------------------------------------------------------
F26            It MUST be possible to move from one network
               interface to another one
-------------------------------------------------------------------
F27            The browser MUST be able to initiate and
```

	accept a media session where the data needed
	for establishment can be carried in SIP.
F28	The browser MUST support a baseline audio and video codec
F29	The browser MUST be able to send streams and data to a peer in the presence of NATs that block UDP traffic.
F30	The browser MUST be able to use the screen (or a specific area of the screen) or what a certain application displays on the screen to generate streams.
F31	The browser MUST be able to use several STUN and TURN servers
F32	There browser MUST support that STUN and TURN servers to use are supplied by other entities than the service provided (i.e. the network provider).
F33	The browser must be able to send reliable data traffic to a peer browser.
F34	The browser MUST support priortization of streams and data.
F35	The browser MUST enable verification, given the right circumstances and by use of other trusted communication, of that streams and data received have not been manipulated by any party.
F36	The browser MUST reject incoming media and data, either modified, created or injected, by any entity not trusted by the site.
F37	The browser MUST be able to send streams and data to a peer in the presence of FWs that only allows http(s) traffic.
F38	The browser MUST be able to collect statistics, related to the transport of audio and video between peers, needed to estimate quality of service.

5.3. API requirements

```
REQ-ID          DESCRIPTION
------------------------------------------------------------------
A1              The Web API MUST provide means for the
                application to ask the browser for permission
                to use cameras and microphones as input devices.
------------------------------------------------------------------
A2              The Web API MUST provide means for the web
                application to control how streams generated
                by input devices are used.
------------------------------------------------------------------
A3              The Web API MUST provide means for the web
                application to control the local rendering of
                streams (locally generated streams and streams
                received from a peer).
------------------------------------------------------------------
A4              The Web API MUST provide means for the web
                application to initiate sending of
                stream/stream components to a peer.
------------------------------------------------------------------
A5              The Web API MUST provide means for the web
                application to control the media format (codec)
                to be used for the streams sent to a peer.

                NOTE: The level of control depends on whether
                the codec negotiation is handled by the browser
                or the web application.
------------------------------------------------------------------
A6              The Web API MUST provide means for the web
                application to modify the media format for
                streams sent to a peer after a media stream
                has been established.
------------------------------------------------------------------
A7              The Web API MUST provide means for
                informing the web application of whether the
                establishment of a stream with a peer was
                successful or not.
------------------------------------------------------------------
A8              The Web API MUST provide means for the web
                application to mute/unmute a stream or stream
                component(s). When a stream is sent to a peer
                mute status must be preserved in the stream
                received by the peer.
------------------------------------------------------------------
A9              The Web API MUST provide means for the web
                application to cease the sending of a stream
```

<div style="margin-left:3em">

to a peer.

A10   The Web API MUST provide means for the web
application to cease processing and rendering
of a stream received from a peer.

A11   The Web API MUST provide means for
informing the web application when a
stream from a peer is no longer received.

A12   The Web API MUST provide means for
informing the web application when high
loss rates occur.

A13   The Web API MUST provide means for the web
application to apply spatialization effects to
audio streams.

A14   The Web API MUST provide means for the web
application to detect the level in audio
streams.

A15   The Web API MUST provide means for the web
application to adjust the level in audio
streams.

A16   The Web API MUST provide means for the web
application to mix audio streams.

A17   For each stream generated, the Web API MUST
provide an identifier that is accessible by the
application. The identifier MUST be accessible
also for a peer receiving that stream and MUST
be unique relative to all other stream
identifiers in use by either party.

A18   The Web API MUST provide a mechanism for sending
and receiving isolated discrete chunks of data.

A19   The Web API MUST provide means for the web
application indicate the type of audio signal
(speech, audio)for audio stream(s)/stream
component(s).

A20   It must be possible for an initiator or a
responder Web application to indicate the types
of media he's willing to accept incoming
streams for when setting up a connection (audio,

</div>

                    video, other). The types of media he's willing
                    to accept can be a subset of the types of media
                    the browser is able to accept.

    A21   The Web API MUST provide means for the
                    application to ask the browser for permission
                    to the screen, a certain area on the screen
                    or what a certain application displays on the
                    screen as input to streams.

    A22   The Web API MUST provide means for the
                    application to specify several STUN and/or
                    TURN servers to use.

    A23   The Web API MUST provide means for the
                    application to specify the priority to
                    apply for outgoing streams and data.

    A24   The Web API MUST provide a mechanism for sending
                    and receiving files.

    A25   It must be possible for the application to
                    refrain from exposing the IP address

    A26   The Web API MUST provide means for the
                    application to obtain the statistics (related
                    to transport, and collected by the browser)
                    needed to estimate quality of service.
    ----

6. IANA Considerations

    TBD

7. Security Considerations

7.1. Introduction

    A malicious web application might use the browser to perform Denial
    Of Service (DOS) attacks on NAT infrastructure, or on peer devices.
    Also, a malicious web application might silently establish outgoing,
    and accept incoming, streams on an already established connection.

    Based on the identified security risks, this section will describe
    security considerations for the browser and web application.

7.2. Browser Considerations

   The browser is expected to provide mechanisms for getting user
   consent to use device resources such as camera and microphone.

   The browser is expected to provide mechanisms for informing the user
   that device resources such as camera and microphone are in use
   ("hot").

   The browser is expected to provide mechanisms for users to revise and
   even completely revoke consent to use device resources such as camera
   and microphone.

   The browser is expected to provide mechanisms for getting user
   consent to use the screen (or a certain part of it) or what a certain
   application displays on the screen as source for streams.

   The browser is expected to provide mechanisms for informing the user
   that the screen, part thereof or an application is serving as a
   stream source ("hot").

   The browser is expected to provide mechanisms for users to revise and
   even completely revoke consent to use the screen, part thereof or an
   application is serving as a stream source.

   The browser is expected to provide mechanisms in order to assure that
   streams are the ones the recipient intended to receive.

   The browser is expected to provide mechanisms that allows the users
   to verify that the streams received have not be manipulated (F35).

   The browser needs to ensure that media is not sent, and that received
   media is not rendered, until the associated stream establishment and
   handshake procedures with the remote peer have been successfully
   finished.

   The browser needs to ensure that the stream negotiation procedures
   are not seen as Denial Of Service (DOS) by other entities.

7.3. Web Application Considerations

   The web application is expected to ensure user consent in sending and
   receiving media streams.

8. Additional use-cases

   Several additional use-cases have been discussed. At this point

these use-cases are not included as requirement deriving use-cases
for different reasons (lack of documentation, overlap with existing
use-cases, lack of consensus). For completeness these additional
use-cases are listed below:

1.   Use-cases regarding different situations when being invited to a
     "session", e.g. browser open, browser open but another tab
     active, browser open but active in session, browser closed, ...
     (Matthew Kaufman); discussed at webrtc meeting

2.   E911 (Paul Beaumont) http://www.ietf.org/mail-archive/web/
     rtcweb/current/msg00525.html, followed up by Stephan Wenger

3.   Local Recording and Remote recording (John): Discussed a _lot_
     on the mail lists (rtcweb as well as public-webrtc) lAugust and
     September 2011. Concrete proposal: http://www.ietf.org/
     mail-archive/web/rtcweb/current/msg01006.html (remote) and http:
     //www.ietf.org/mail-archive/web/rtcweb/current/msg00734.html
     (local)

4.   Emergency access for disabled (Bernard Aboba) http://
     www.ietf.org/mail-archive/web/rtcweb/current/msg00478.html

5.   Clue use-cases (Roni Even) http://tools.ietf.org/html/
     draft-ietf-clue-telepresence-use-cases-01

6.   Rohan red cross (Cullen Jennings); http://www.ietf.org/
     mail-archive/web/rtcweb/current/msg00323.html

7.   Security camera/baby monitor usage http://www.ietf.org/
     mail-archive/web/rtcweb/current/msg00543.html

8.   Large multiparty session http://www.ietf.org/mail-archive/web/
     rtcweb/current/msg00530.html

9.   Call center http://www.ietf.org/mail-archive/web/rtcweb/current/
     msg04203.html

10. Enterprise policies http://www.ietf.org/mail-archive/web/rtcweb/
     current/msg04271.html

11. Low-complex multiparty central node http://www.ietf.org/
     mail-archive/web/rtcweb/current/msg04430.html

12. Multiparty central node that is not allowed to decipher http://
     www.ietf.org/mail-archive/web/rtcweb/current/msg04457.html

13. Enable company coop without being able to decipher http://
     www.ietf.org/mail-archive/web/rtcweb/current/msg04461.html

9. Acknowledgements

Dan Burnett has reviewed and proposed a lot of things that enhances
the document. Most of this has been incorporated in rev -05.

Stephan Wenger has provided a lot of useful input and feedback, as
well as editorial comments.

Harald Alvestrand and Ted Hardie have provided comments and feedback
on the draft.

Harald Alvestrand and Cullen Jennings have provided additional use-
cases.

Thank You to everyone in the RTCWEB community that have provided
comments, feedback and improvement proposals on the draft content.

[10](#). Change Log

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

Changes from [draft-ietf-rtcweb-use-cases-and-requirements-09](#)

o Changed "video communication session" to "audiovisual
   communication session.

Changes from [draft-ietf-rtcweb-use-cases-and-requirements-08](#)

o Changed "eavesdropping" to "wiretapping" and referenced [RFC2804](#).
o Removed informal ref webrtc_req; that document has been abandoned
   by the W3C webrtc WG.
o Added use-case where one user is behind a FW that only allows
   http; derived req. F37.
o Changed F24 slightly; MUST-> SHOULD, inserted "available".
o Added a clause to "Simple video communication service" saying that
   the service provider monitors the quality of service, and derived
   reqs F38 and A26.

Changes from [draft-ietf-rtcweb-use-cases-and-requirements-07](#)

o Added "and data exchange" to 1. Introduction.
o Removed cone and symmetric NAT from 4.1 Introduction, refers to
   [RFC4787](#) instead.
o Added text on enabling verifyication of that the media has not
   been manipulated by anyone to use-case "Simple Video Communication
   Service", derived req. F35
o Added text on that the browser should reject media (data) that has
   been created/injected/modified by non-trusted party, derived req.
   F36
o Added text on enabling the app to refrain from revealing IP
   address to use-case "Simple Video Communication Service", derived
   req. A25
o Added use-case "Simple Video Communication Service with file
   exchange", derived reqs F33 and A24
o Added priority of video streams to "Hockey game viewer" use case,
   added priority of data to "on-line game use-case", derived reqs
   F34 and A23

o In F22, "the IVR" -> "a DTMF based IVR".

o Updated req F23 to clarify that requirements such as NAT
   traversal, prtoection from eavesdropping, rate control applies
   also to datagram.

Changes from draft-ietf-rtcweb-use-cases-and-requirements-06

o Renaming of requirements (FaI1 -> F31), (FaI2 -> F32) and (AaI1 ->
   A22)

Changes from draft-ietf-rtcweb-use-cases-and-requirements-05

o Added use-case "global service provider", derived reqs associated
   with several STUN/TURN servers

o Added use-case "enterprise aspects", derived req associated with
   enabling the network provider to supply STUN and TURN servers

o The requirements from the above are ICE specific and labeled
   accordingly

o Separated the requirements phrased like "processing such as pan,
   mix and render" for audio to be specific reqs on spatialization,
   level measurement, level adjustment and mixing (discussed on the
   lists in
   http://www.ietf.org/mail-archive/web/rtcweb/current/msg01648.html
   and http://lists.w3.org/Archives/Public/public-webrtc/2011Sep/
   0102.html)

o Added use-case on sharing as decided in
   http://www.ietf.org/mail-archive/web/rtcweb/current/msg01700.html,
   derived reqs F30 and A21

o Added the list of common considerations proposed in mail
   http://www.ietf.org/mail-archive/web/rtcweb/current/msg01562.html
   to the Introduction of the use-case section

Changes from draft-ietf-rtcweb-use-cases-and-requirements-04

o Most changes based on the input from Dan Burnett
   http://www.ietf.org/mail-archive/web/rtcweb/current/msg00948.html

o Many editorial changes

o 4.2.1.1 Clarified

o Some clarification added to 4.3.1.1 as a note

o F-requirements updated (see reply to Dan's mail).

o Almost all A-requirements updated to start "The Web API MUST
   provide ..."

o A8 removed, A9 rephrased to cover A8 and old A9

o A15 rephrased

o For more details, and discussion, look att the response to Dan's
   mail
   http://www.ietf.org/mail-archive/web/rtcweb/current/msg01177.html

Changes from draft-ietf-rtcweb-use-cases-and-requirements-03

o Editorials
o Changed when the self-view is displayed in 4.2.1.1, and added
   words about allowing users to remove and re-insert it.
o Clarified 4.2.6.1
o Removed the "mono" stuff from 4.2.7.1
o Added that communication should not be possible to eavesdrop to
   most use cases - and req. F17
o Re-phrased 4.3.3.1 to not describe the technical solution so much,
   and removed "stereo" stuff. Solution possibilities are now in a
   note.
o Re-inserted API requirements after discussion in the W3C webrtc
   WG. (Re-phrased A15 and added A18 compared to version -02).

Changes from draft-ietf-rtcweb-use-cases-and-requirements-02

o Removed desrciption/list of API requirements, instead
o Reference to W3C webrtc_reqs document for API requirements

Changes from draft-ietf-rtcweb-ucreqs-01

o Changed Intended status to Information
o Changed "Ipr" to "trust200902"
o Added use case "Simple video communication service, NAT/FW that
   blocks UDP", and derived new req F26
o Added use case "Distributed Music Band" and derived new req A17
o Added F24 as requirement derived from use case "Simple video
   communication service with inter-operator calling"
o Added section "Additional use cases"
o Added text about ID handling to multiparty with central server use
   case
o Re-phrased A1 slightly

Changes from draft-ietf-rtcweb-ucreqs-00

o - Reshuffled: Just two main groups of use cases (b2b and b2GW/
   Server); removed some specific use cases and added them instead as
   flavors to the base use case (Simple video communciation)
o - Changed the fromulation of F19
o - Removed the requirement on an API for DTMF
o - Removed "FX3: There SHOULD be a mapping of the minimum needed
   data for setting up connections into SIP, so that the restriction
   to SIP-carriable data can be verified. Not a rew on the browser
   but rather on a document"
o - (see
   http://www.ietf.org/mail-archive/web/rtcweb/current/msg00227.html
   for more details)

   o -Added text on informing user of that mic/cam is being used and
      that it must be possible to revoce permission to use them in
      section 7.
   Changes from draft-holmberg-rtcweb-ucreqs-01
   o - Draft name changed to draft-ietf-rtcweb-ucreqs
   o - Use-case grouping introduced
   o - Additional use-cases added
   o - Additional reqs added (derived from use cases): F19-F25, A16-A17

   Changes from draft-holmberg-rtcweb-ucreqs-00
   o - Mapping between use-cases and requirements added (Harald
      Alvestrand, 090311)
   o - Additional security considerations text (Harald Alvestrand,
      090311)
   o - Clarification that user applications are assumed to be executed
      by a browser (Ted Hardie, 080311)
   o - Editorial corrections and clarifications

11. References

11.1. Normative References

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

   [RFC2804] IAB and IESG, "IETF Policy on Wiretapping", RFC 2804,
              May 2000.

11.2. Informative References

Authors' Addresses

   Christer Holmberg
   Ericsson
   Hirsalantie 11
   Jorvas 02420
   Finland

   Email: christer.holmberg@ericsson.com

Stefan Hakansson
Ericsson
Laboratoriegrand 11
Lulea 97128
Sweden

Email: stefan.lk.hakansson@ericsson.com

Goran AP Eriksson
Ericsson
Farogatan 6
Stockholm 16480
Sweden

Email: goran.ap.eriksson@ericsson.com