RTCWEB Working Group
Internet-Draft
Intended status: Informational
Expires: August 10, 2014

C. Holmberg
S. Hakansson
G. Eriksson
Ericsson

February 6, 2014

Web Real-Time Communication Use-cases and Requirements
draft-ietf-rtcweb-use-cases-and-requirements-13.txt

Abstract

This document describes web based real-time communication use-cases. Requirements on the browser functionality are derived from the use-cases.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 10, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 2
2. Conventions 3
3. Use-cases 3
3.1. Introduction 3
3.2. Common requirements 4
3.3. Browser-to-browser use-cases 4
3.3.1. Simple Video Communication Service 4
3.3.2. Simple Video Communication Service, NAT/Firewall that blocks UDP 5
3.3.3. Simple Video Communication Service, Firewall that only allows traffic via a HTTP Proxy 5
3.3.4. Simple Video Communication Service, global service provider 5
3.3.5. Simple Video Communication Service, enterprise aspects 6
3.3.6. Simple Video Communication Service, access change . . 7
3.3.7. Simple Video Communication Service, QoS 7
3.3.8. Simple Video Communication Service with screen sharing 8
3.3.9. Simple Video Communication Service with file exchange 8
3.3.10. Hockey Game Viewer 8
3.3.11. Multiparty video communication 9
3.3.12. Multiparty on-line game with voice communication . . 10
3.4. Browser - GW/Server use cases 10
3.4.1. Telephony terminal 11
3.4.2. Fedex Call 11
3.4.3. Video conferencing system with central server 11
4. Requirements 12
4.1. General 12
4.2. Browser requirements 13
5. IANA Considerations 16
6. Security Considerations 16
6.1. Introduction 16
6.2. Browser Considerations 16
6.3. Web Application Considerations 17
7. Acknowledgements 17
8. Change Log 17
9. Normative References 23
Appendix A. API requirements 23
Authors' Addresses 26

1. Introduction

This document presents a few use-cases of web applications that are executed in a browser and use real-time communication capabilities. In most of the use-cases all end-user clients are web applications,

but there are some use-cases where at least one of the end-user clients is of another type (e.g. a mobile phone or a SIP UA).

Based on the use-cases, the document derives requirements related to browser functionality. These requirements are named "Fn", where n is an integer, and are described in Section 4.2.

This document was developed in an initial phase of the work with rather minor updates at later stages. It has not really served as a tool in deciding features or scope for the WGs efforts so far. It is proposed to be used in a later phase to evaluate the protocols and solutions developed by the WG.

This document also lists requirements related to the API to be used by web applications as an appendix. The reason is that the W3C WebRTC WG has decided to not develop its own use-case/requirement document, but instead use this document. These requirements are named "An", where n is an integer, and are described in Appendix A-

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. Use-cases

3.1. 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:

- Clients can be on IPv4-only
- Clients can be on IPv6-only
- Clients can be on dual-stack
- Clients can be connected to networks with different throughput capabilities
- Clients can be on variable-media-quality networks (wireless)
- Clients can be on congested networks

- Clients can be on firewalled networks with no UDP allowed

- Clients can be on networks with a NAT using any type of Mapping and Filtering behaviors (as described in RFC4787).

3.2. Common requirements

The requirements retrived from the Simple Video Communication Service use-case (Section 3.3.1) by default apply to all other use-cases, and are considred common. For each individual use-case, only the additional requirements are listed. The following requirements can be derived from, and apply to, each of the documented use-cases. For each individual use-case, only requirements that are not part of the common requirements are listed.

3.3. Browser-to-browser use-cases

3.3.1. Simple Video Communication Service

3.3.1.1. 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 media and data be encrypted, authenticated and integrity protected on a per-packet basis and that media and data packets failing the integrity check not be delivered to the application.

The application gives the users the opportunity to stop it from exposing the host 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 with different operating systems and browsers from different vendors.

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

3.3.1.2. Common Requirements

F1, F2, F3, F4, F5, F8, F9, F10, F20, F25, F28, F35, F36, F38, F39

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

3.3.2. Simple Video Communication Service, NAT/Firewall that blocks UDP

3.3.2.1. Description

This use-case is almost identical to the Simple Video Communication Service use-case (Section 3.3.1). The difference is that one of the users is behind a NAT/Firewall that blocks UDP traffic.

3.3.2.2. Additional Requirements

F29

3.3.3. Simple Video Communication Service, Firewall that only allows traffic via a HTTP Proxy

3.3.3.1. Description

This use-case is almost identical to the Simple Video Communication Service use-case (Section 3.3.1). The difference is that one of the users is behind a Firewall that only allows traffic via a HTTP Proxy.

3.3.3.2. Additional Requirements

F37

3.3.4. Simple Video Communication Service, global service provider

3.3.4.1. Description

This use-case is almost identical to the Simple Video Communication Service use-case (Section 3.3.1).

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.

Note that ICE support being mandatory does not preclude a WebRTC endpoint from supporting additional traversal mechanisms.

3.3.4.2. Additional Requirements

F31

A22

3.3.5. Simple Video Communication Service, enterprise aspects

3.3.5.1. Description

This use-case is similar to the Simple Video Communication Service use-case (Section 3.3.1).

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 sessions used from inside the company towards any external peer. To be able to do this they deploy a TURN server that straddles the boundary between the internal and the external network.

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 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 possible to configure the browsers used in the enterprise with network specific STUN and TURN servers. This should be possible to achieve by auto-configuration 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.

3.3.5.2. Additional Requirements

F32

3.3.6. Simple Video Communication Service, access change

3.3.6.1. Description

This use-case is almost identical to the Simple Video Communication Service use-case (Section 3.3.1). The difference is that the user changes network access during the session.

The communication device used by one of the users has 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.

3.3.6.2. Additional Requirements

F26

3.3.7. Simple Video Communication Service, QoS

3.3.7.1. Description

This use-case is almost identical to the Simple Video Communication Service, access change use-case (Section 3.3.6). 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.

3.3.7.2. Additional Requirements

F24, F26

3.3.8. Simple Video Communication Service with screen sharing

3.3.8.1. Description

This use-case has the audio and video communication of the Simple Video Communication Service use-case (Section 3.3.1).

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 application displays with the peer.

3.3.8.2. Additional Requirements

F30

A21

3.3.9. Simple Video Communication Service with file exchange

3.3.9.1. Description

This use-case has the audio and video communication of the Simple Video Communication Service use-case (Section 3.3.1).

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

3.3.9.2. Additional Requirements

F30, F33

A21, A24

3.3.10. Hockey Game Viewer

3.3.10.1. 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. On the

mobile phone, only the camera facing the talent scout is used. On the user 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.

3.3.10.2. Additional Requirements

F17, F24

A17, A23

3.3.11. Multiparty video communication

3.3.11.1. Description

In this use-case, the Simple Video Communication Service use-case (Section 3.3.1) 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 the user experience, the web application renders the audio coming from different particiapants so that it is experienced to come from different spatial locations. 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.

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 received 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.

3.3.11.2. Additional Requirements

F11, F12, F13, F14, F15, F16, F17

A13, A14, A15, A16, A17

3.3.12. Multiparty on-line game with voice communication

3.3.12.1. 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.

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.

3.3.12.2. Additional Requirements

F12, F13, F14, F15, F16, F18, F23, F24

A13, A14, A15, A16, A17, A18, A23

3.4. Browser - GW/Server use cases

3.4.1. Telephony terminal

3.4.1.1. 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.

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.

3.4.1.2. Additional Requirements

F21, F27

3.4.2. Fedex Call

3.4.2.1. Description

Alice uses her web browser with a service that allows her to call PSTN numbers. Alice calls 1-800-gofedex. Alice should be able to hear the initial prompts from the fedex Interactive Voice Responder (IVR) and when the IVR says press 1, there should be a way for Alice to navigate the IVR.

3.4.2.2. Additional Requirements

F21, F22

3.4.3. Video conferencing system with central server

3.4.3.1. 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 sends 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.

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 Firewalls. 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.

3.4.3.2. Additional Requirements

F17

A17

4. Requirements

4.1. General

This section contains the requirements on the browser derived from the use-cases in Section 3.

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.

4.2. 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 (meaning that the browser must, regardless of application behavior, reduce send rate when there is congestion).
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 should be able to render good quality audio and video even in the presence of reasonable levels of jitter and packet losses.
F7	The browser must support insertion of reference frames in outgoing media streams when requested by a peer.
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.
F10	The browser must support synchronization of audio and video.
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 when playing audio streams.
F14	The browser must be able to measure the voice activity level in audio streams.
F15	The browser must be able to change the voice activity 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.
F20	It must be possible to protect streams and data from wiretapping [RFC2804].
F21	The browser must support an audio media format (codec) that is commonly supported by existing telephony services.
F22	There should be a way to navigate a Dual-tone multi-frequency signaling (DTMF) based Interactive voice response (IVR) System
F23	The browser must be able to send short latency unreliable datagram traffic to a peer browser [RFC5405].
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	The communication session must survive across a

	change of the network interface used by the session
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 and Firewalls that block UDP traffic.
F30	The browser must be able to generate streams using the entire user display, a specific area of the user's display or the information being displayed by a specific application.
F31	The browser must be able to use several STUN and TURN servers
F32	The browser must support the use of STUN and TURN servers that are supplied by entities other than the web application (i.e. the network provider).
F33	The browser must be able to send reliable data traffic to a peer browser.
F35	The browser must enable verification, given the right circumstances and by use of other trusted communication, that streams and data received have not been manipulated by any party.
F36	The browser must encrypt, authenticate and integrity protect media and data on a per-packet basis, and must drop incoming media and data packets that fail the per-packet integrity check. In addition, the browser must support a mechanism for cryptographically binding media and data security keys to the user identity (see R-ID-BINDING in [RFC5479]).
F37	The browser must be able to send streams and data to a peer in the presence of Firewalls that only allows traffic via a HTTP Proxy, when Firewall policy allows WebRTC 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 experience.
F39	The browser must make it possible to set up a call between two parties without one party learning the other party's host IP address.

5. IANA Considerations

There are no IANA actions in this document.

6. Security Considerations

6.1. 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.

6.2. 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.

6.3. Web Application Considerations

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

7. Acknowledgements

The authors wish to thank Bernard Aboba, Gunnar Hellstrom, Martin Thomson, Lars Eggert, Matthew Kaufman, Emil Ivov, Eric Rescorla, Eric Burger, John Leslie, Dan Wing, Richard Barnes, Barry Dingle, Dale Worley, Ted hardie, Mary Barnes, Dan Burnett, Stephan Wenger, Harald Alvestrand, Cullen Jennings, Andrew Hutton and everyone else in the RTCWEB community that have provided comments, feedback, text and improvement proposals on the document.

8. Change Log

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

Changes from draft-ietf-rtcweb-use-cases-and-requirements-10

- Described that the API requirements are really from a W3C perspective and are supplied as an appendix in the introduction. Moved API requirements to an Appendix.

- Removed the "Conventions" section with the key-words and reference to RFC2119. Also changed uppercase MUST's/SHOULD's to lowercase.

- Added a note on the proposed use of the document to the introduction.

- Removed the note talking about WS from the "Firewall that only allows http" use-case.

- Removed the word "Skype" that was used as example in one of the use-cases.

- Clarified F3 (the req saying the everything the browser sends must be rate controlled).

- Removed the TBD saying we need to define reasonable levels from the requirement saying that quality must be good even in presence of packet losses (F5), and changed "must" to "should" (Based on a list discussion involving Bernard).

- Removed F6 ("The browser must be able to handle high loss and jitter levels in a graceful way."), also after a list discussion.

- Clarified F7 (used to say that the browser must support fast stream switches, now says that reference frames must be inserted when requested).

- Removed the questions from F9 (echo cancellation), F10 (synchronization), F21 (telephony codec).

- Exchanged "restrictive firewalls" for "limited middleboxes" in F19 (as proposed by Martin).

- Expanded DTMF and IVR in F22 (proposed by Martin)

- Added ref to RFC5405 in F23 (proposed by Lars Eggert).

- Exchanged "service provided" for "web application" in F32.

- Changed the text in 3.2.1 that motivates F36 (new text "It is essential that media and data be encrypted, authenticated ... bound to the user identity."); and rewrote F36, included a ref to RFC5479.

- Changed "quality of service" to "quality of experience" in F38.

- Added F39.

- Used new formulation of A17 (proposed by Martin).
- Updated A20.
- Updated A25.

Changes from draft-ietf-rtcweb-use-cases-and-requirements-09

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

Changes from draft-ietf-rtcweb-use-cases-and-requirements-08

- Changed "eavesdropping" to "wiretapping" and referenced RFC2804.
- Removed informal ref webrtc_req; that document has been abandoned by the W3C webrtc WG.
- Added use-case where one user is behind a Firewall that only allows http; derived req. F37.
- Changed F24 slightly; MUST-> SHOULD, inserted "available".
- 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

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

- 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

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

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

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

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

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

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

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

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

- 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)

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

- 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

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

- Many editorial changes

- 4.2.1.1 Clarified

- Some clarification added to 4.3.1.1 as a note
- F-requirements updated (see reply to Dan's mail).
- Almost all A-requirements updated to start "The Web API MUST provide ..."
- A8 removed, A9 rephrased to cover A8 and old A9
- A15 rephrased
- For more details, and discussion, look at 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

- Editorials
- Changed when the self-view is displayed in 4.2.1.1, and added words about allowing users to remove and re-insert it.
- Clarified 4.2.6.1
- Removed the "mono" stuff from 4.2.7.1
- Added that communication should not be possible to eavesdrop to most use cases - and req. F17
- 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.
- 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

- Removed description/list of API requirements, instead
- Reference to W3C webrtc_reqs document for API requirements

Changes from draft-ietf-rtcweb-ucreqs-01

- Changed Intended status to Information
- Changed "Ipr" to "trust200902"

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

Changes from draft-ietf-rtcweb-ucreqs-00

- - 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 communication)
- - Changed the formulation of F19
- - Removed the requirement on an API for DTMF
- - 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"
- - (see http://www.ietf.org/mail-archive/web/rtcweb/current/msg00227.html for more details)
- -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

- - Draft name changed to draft-ietf-rtcweb-ucreqs
- - Use-case grouping introduced
- - Additional use-cases added
- - 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

9. 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.

[RFC5405] Eggert, L. and G. Fairhurst, "Unicast UDP Usage Guidelines for Application Designers", BCP 145, RFC 5405, November 2008.

[RFC5479] Wing, D., Fries, S., Tschofenig, H., and F. Audet, "Requirements and Analysis of Media Security Management Protocols", RFC 5479, April 2009.

Appendix A. API requirements

This section contains the requirements on the API derived from the use-cases in Section 3.

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 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	The Web API must provide a way to identify streams such that an application is able to match streams on a sending peer with the same stream on all receiving peers.
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 to 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 it is willing to accept incoming streams for when setting up a connection (audio, video, other). The types of media to be accepted 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 instruct the browser to refrain from exposing the host IP address to the application
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.

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