RTCWEB Working Group	C.H. Holmberg
Internet-Draft	S.H. Hakansson
Intended status: Standards Track	G.E. Eriksson
Expires: December 29, 2011	Ericsson
	June 27, 2011

Web Real-Time Communication Use-cases and Requirements
draft-ietf-rtcweb-use-cases-and-requirements-00.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 services provided by the browser.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

This document presents a few use-case 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 document focuses on requirements related to real-time media streams. 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 later derived.

4.2. Browser-to-browser use-cases

4.2.1. Simple Video Communication Service

4.2.1.1. Description

In the service the users have loaded, and logged into, a video communication web application into their browsers, provided by the same service provider. The web service publishes information about user login status, by pushing updates to the web application in the browsers. By selecting an online peer user, a 1-1 video communication session between the browsers of the peers is initiated. The invited user might accept or reject the session.
When the session has been established, a self-view, as well as the video sent from the remote peer, are displayed. The users can change

the display sizes during the session. The users can also pause sending of media (audio, video, or both), and mute incoming media.
Any session participant can end the session at any time.
One participant has an unreliable internet connection. It sometimes has packet losses, and is sometimes goes down completely.
One participant is located behind a Network Address Translator (NAT).

4.2.1.2. Derived Requirements

F1, F2, F3, F4, F5, F6, F8, F9, F10, F22
A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A13

4.2.2. Simple video communication service with inter-operator calling

4.2.2.1. 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.
Each web service publishes information about user login status for users that have a relationship with the other user; how this is established is out of scope.
The same functionality as in the "Simple Video Communication Service" is available.
The same issues with connectivity apply.

4.2.2.2. Derived requirements

F24: The browser MUST be able to initiate and accept a media session where the data needed for establishment can be carried in SIP.
F25: The browser MUST support a baseline audio and video codec
(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)

4.2.3. Hockey Game Viewer

4.2.3.1. 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 for viewing the game and discussing with the talent scout. 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, while a thumbnail of the rear facing camera is overlaid.

Most of the mobile phone screen is covered by a self view of the front facing camera. A thumbnail of the rear facing cameras view is overlaid.

4.2.3.2. Derived Requirements

F1, F2, F3, F4, F5, F6, F8, F9, F10, F14
A1, A2, A3, A4, A5, A7, A8, A9, A10, A11, A12, A13, A15

4.2.4. Video Size Change

4.2.4.1. Description

Alice and Bob are in a video call in their browsers and have negotiate a high resolution video. Bob decides to change the size of the windows his browser is displaying video to a small size.
Bob's browser regenerates the video codec paramters with Alice's browser to change the resolution of the video Alice sends to match the smaller size.

4.2.4.2. Derived Requirements

F22 (It SHOULD be possible to modify video codec parameters during a session.)

4.3. Telephony 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 will be shown in the same manner as when a mobile phone used.

4.3.1.2. Derived Requirements

F1, F2, F3, F4, F5, F6, F8, F9, F10, F18, F19
A1, A2, A3, A4, A7, A8, A9, A10, A11, A12, A13, A16

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

F19 (DTMF)
A16 (DTMF API)

4.4. Video conferenceing use-cases

4.4.1. Multiparty video communication

4.4.1.1. Description

In this use case the simple video communication service 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 audio sent by each participant is a mono stream. However, 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.
Each video stream received is by default displayed in a thumbnail frame within the browser, but users can change the display size.

4.4.1.2. Derived Requirements

F1, F2, F3, F4, F5, F6, F8, F9, F10, F11, F12, F13, F14
A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A13, A14, A15

4.4.2. Video conferencing system with central server

4.4.2.1. Description

An organization uses a video communication system that supports the establishment of multiparty video sessions using a central conference server.
The browsers of all participants send an audio stream (mono or stereo depending on the equipment of a participant) to the central server. The central server mixes the audio streams and sends towards the participants a mixed stereo stream.
All participants send two video streams towards the server, one low resolution and one high resolution. At 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, based on speech activity.
The organization has an internal network set up with an aggressive firewall handling access to the internet. If users can not physically access the internal network, they can establish a Virtual Private Network (VPN).
It is essential that the communication can not be eavesdropped.

4.4.2.2. Derived Requirements

F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F14, F16, F17
A1, A2, A3, A4, A5, A7, A8, A9, A10, A11, A12, A13, A15

4.5. Embedded voice communicatoin use-cases

4.5.1. Multiparty on-line game with voice communication

4.5.1.1. Description

In this use-case, the voice part of the multiparty video communication application 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.

4.5.1.2. Derived Requirements

F1, F2, F3, F4, F5, F6, F8, F9, F11, F12, F13, F15, F20
A1, A2, A3, A4, A5, A7, A8, A9, A10, A11, A12, A13, A14, A15, A17

4.6. Bandwidth/QoS/mobility use-cases

4.6.1. NIC Change

4.6.1.1. Description

Alice is using her notebook computer that is plugged in to 1G ethernet and has 802.11 wireless interface. Alice is in a call talking with Bob and decides to unplug her notebook computer and walk down to a different room, and continue the call from there.

4.6.1.2. Derived Requirements

F23: It MUST be possible to move from one network interface to another one.

4.6.2. QoS Marking

4.6.2.1. Description

Alice's browser is on a computer behind a common residential router that supports prioritization of traffic.
F21: The browser MUST be able to take advantage of capabilities to prioritize voice and video appropriately.

4.6.2.2. Derived Requirements

F19: (DTMF)

5. Requirements

5.1. General

This section contains 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 (OS), 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 to a peer in presence of NATs.
F3	Transmitted streams MUST be rate controlled.
F4	The browser MUST be able to receive, process and render streams from peers.
F5	The browser MUST be able to render good quality audio and video even in 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 any more
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 to several peers concurrently.
F12	The browser MUST be able to receive streams from multiple peers concurrently.

```
F13             The browser MUST be able to pan, mix and render
                several concurrent audio streams.
----------------------------------------------------------------
F14             The browser MUST be able to render several
                concurrent video streams
----------------------------------------------------------------
F15             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).
----------------------------------------------------------------
F16             Streams MUST be able to pass through restrictive
                firewalls.
----------------------------------------------------------------
F17             It MUST be possible to protect streams from
                eavesdropping.
----------------------------------------------------------------
F18             The browser MUST support an audio media format
                (codec) that is commonly supported by existing
                telephony services.

                QUESTION: G.711?
----------------------------------------------------------------
F19             The browser must be able to insert DTMF signals
                in a media stream
----------------------------------------------------------------
F20             The browser must be able to send short
                latency datagram traffic to a peer browser
----------------------------------------------------------------
F21             The browser MUST be able to take advantage of
                capabilities to prioritize voice and video
                appropriately.
----------------------------------------------------------------
F22             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
----------------------------------------------------------------
F23             It MUST be possible to move from one network
                interface to another one
----------------------------------------------------------------
F24             The browser MUST be able to initiate and accept a
                media session where the data needed for establishment
                can be carried in SIP.
----------------------------------------------------------------
F25             The browser MUST support a baseline audio and
                video codec
----------------------------------------------------------------
```

5.3. API requirements

REQ-ID	DESCRIPTION
A1	The web application MUST be able to query the user about the usage of cameras and microphones as input devices.
A2	The web application MUST be able to control how streams generated by input devices are used.
A3	The web application MUST be able to control the local rendering of streams (locally generated streams and streams received from a peer).
A4	The web application MUST be able to initiate sending of stream/stream components to a peer.
A5	The web application MUST be able 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	After a media stream has been established, the web application MUST be able to modify the media format for streams sent to a peer.
A7	The web application MUST be made aware of whether the establishment of a stream with a peer was successful or not.
A8	The web application MUST be able to pause/unpause the sending of a stream to a peer.
A9	The web application MUST be able to mute/unmute a stream received from a peer.
A10	The web application MUST be able to cease the sending of a stream to a peer.
A11	The web application MUST be able to cease processing and rendering of a stream received from a peer.
A12	The web application MUST be informed when a stream from a peer is no longer received.

A13	The web application MUST be informed when high loss rates occur.
A14	It MUST be possible for the web application to control panning, mixing and other processing for individual streams.
A15	The web application MUST be able to identity the context of a stream.
A16	It MUST be possible for the web application to order the browser to insert DTMF tones in a stream
A17	It MUST be possible for the web application to send and receive datagrams to/from peer

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

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.

9. Change Log

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

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

10. References

10.1. Normative References

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

10.2. Informative References

Authors' Addresses

Christer Holmberg Holmberg Ericsson Hirsalantie 11 Jorvas, 02420 Finland EMail: christer.holmberg@ericsson.com

Stefan Hakansson Hakansson Ericsson Laboratoriegrand 11 Lulea, 97128 Sweden EMail: stefan.lk.hakansson@ericsson.com

Goran AP Eriksson Eriksson Ericsson Farogatan 6 Stockholm, 16480 Sweden EMail: goran.ap.eriksson@ericsson.com