RTCWEB Working Group C. Holmberg
Internet-Draft S. Hakansson
Intended status: Standards Track G. Eriksson
Expires: September 8, 2011 Ericsson
March 7, 2011

Web Real-Time Communication Use-cases and Requirements
draft-holmberg-rtcweb-ucreqs-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.

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 retrieved.

4.2. Use-case: Simple Video Communication Service

4.2.1. 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).

4.3. Use-case: Multiparty video communication

4.3.1. 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. Use-case: Multiparty on-line game with voice communication

4.4.1. 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.

4.5. Use-case: Video conferencing system with central server

4.5.1. 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.

4.6. Use-case: Hockey game viewer

4.6.1. 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.

4.7. Use-case: Telephony terminal

4.7.1. 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.

5. Requirements

5.1. General

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

NOTE: 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 provided 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 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?

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 layout 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.
```

----

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 describes 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.

7.3. Web Application Considerations

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

8. Acknowledgements

TBD

9. Change Log

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

Changes from draft-holmberg-rtcweb-ucreqs-old
o TBD

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
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 Eriksson
Ericsson
Farogatan 6
Stockholm 16480
Sweden

Email: goran.ap.eriksson@ericsson.com