Use cases and reqs

Christer Holmberg

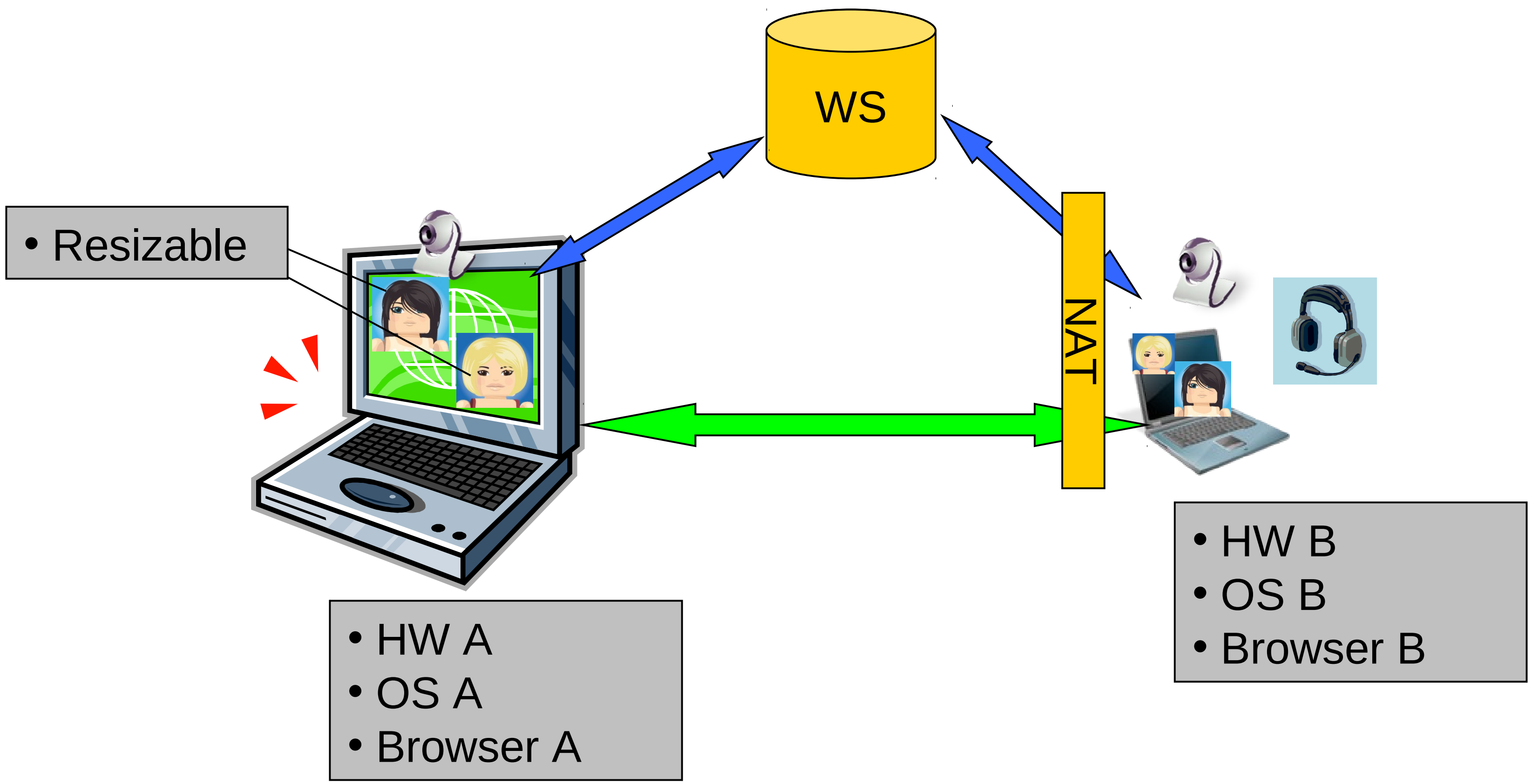
Göran Eriksson

Stefan Håkansson

Intro

- Each use case from
  http://datatracker.ietf.org/doc/draft-ietf-rtcweb-use-cases-and-requirements/?include_text=1
  presented
- Note that the doc only deals with reqs related to real time streams

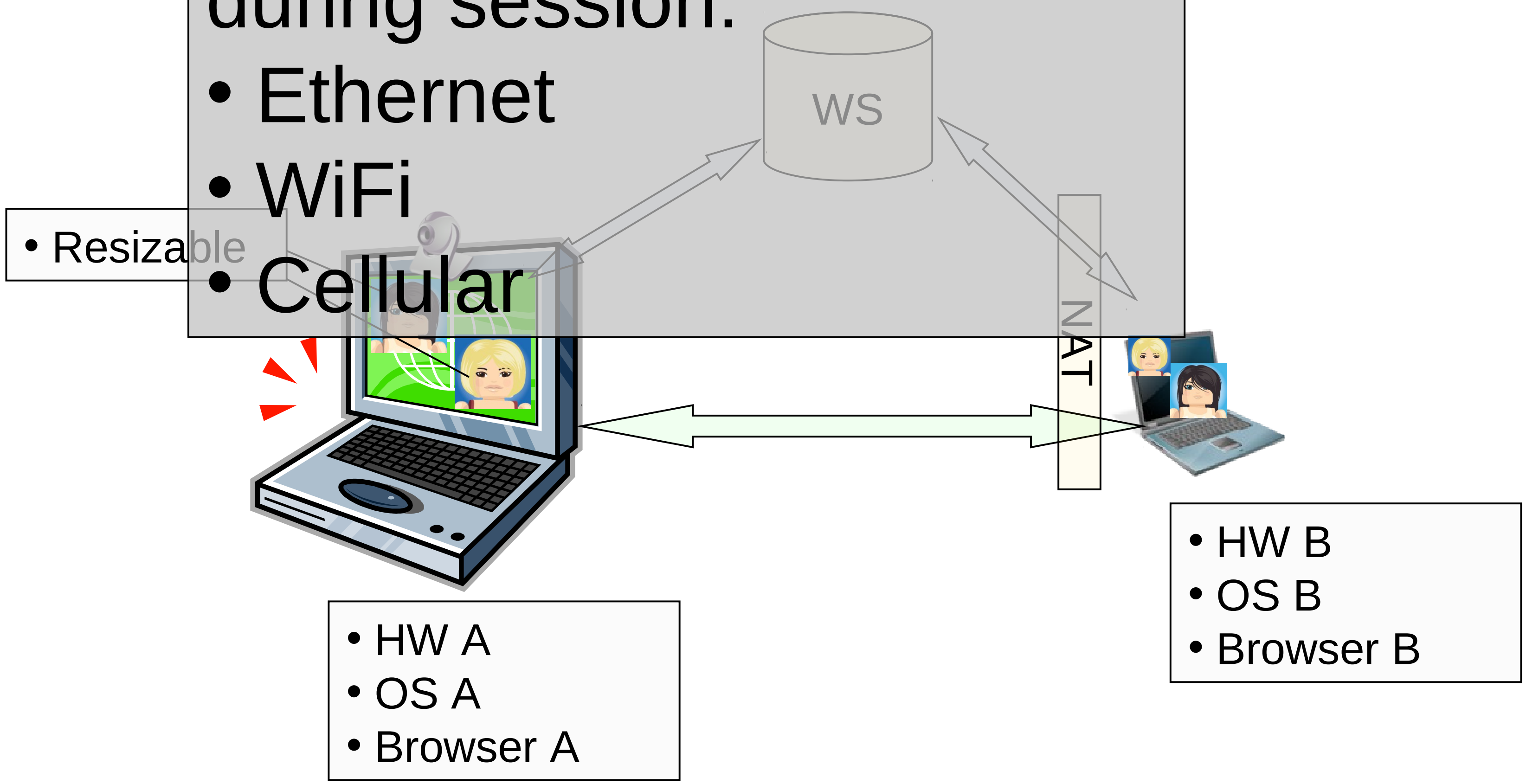
Simple Video Communication Service



Simple Video Communication Service, access change

Change network access during session:

- Ethernet
- WiFi
- Cellular



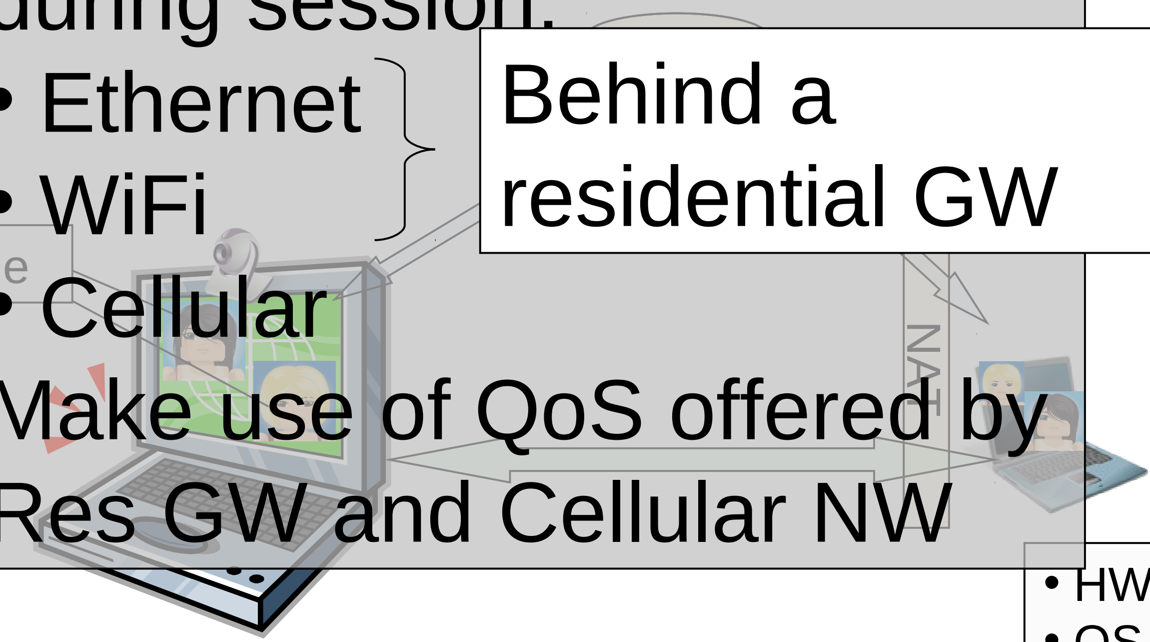
Simple Video Communication Service, QoS

Change network access during session:

- Ethernet
- WiFi
- Cellular

Make use of QoS offered by Res GW and Cellular NW

Behind a residential GW

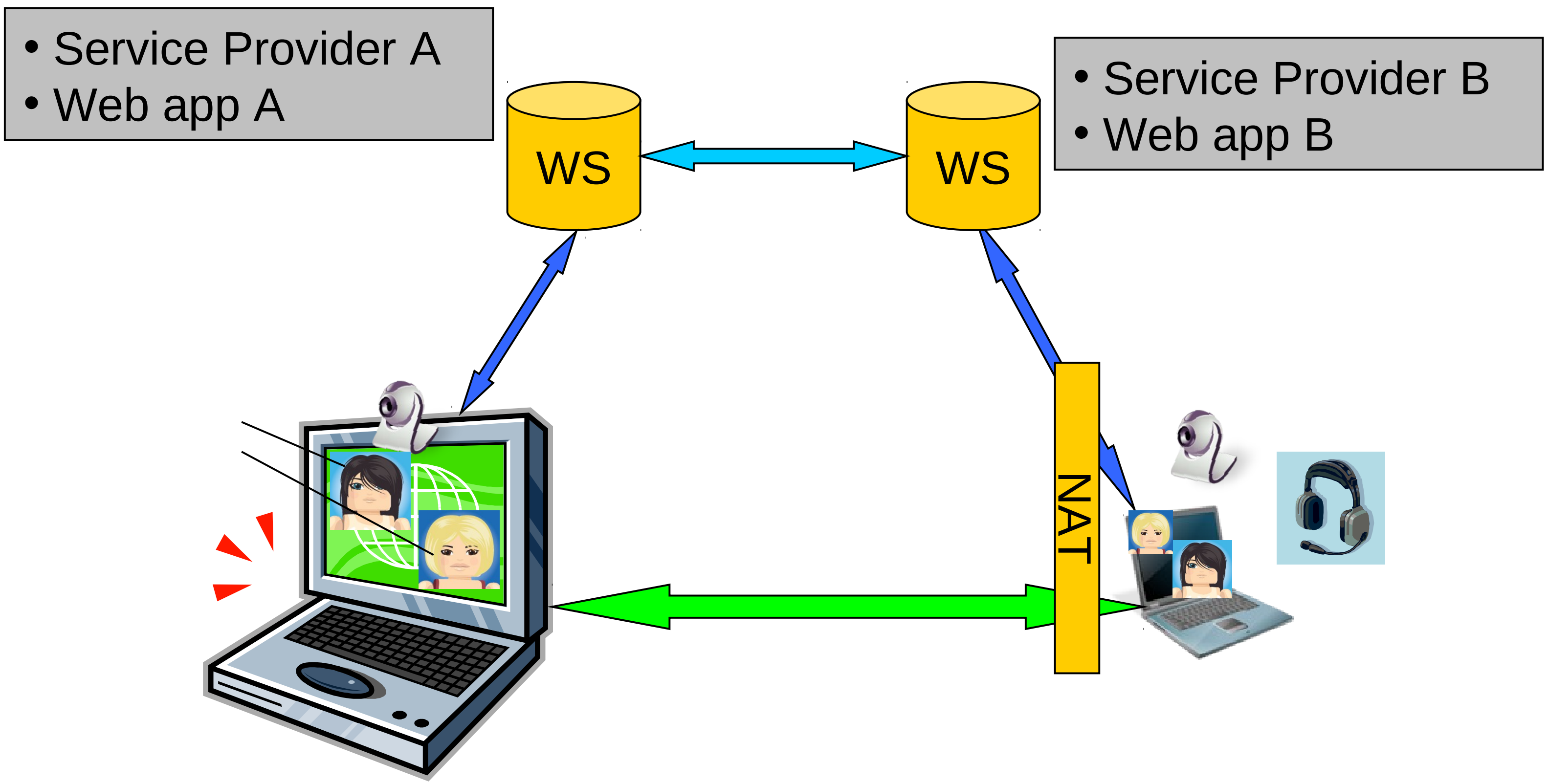


- Resizable

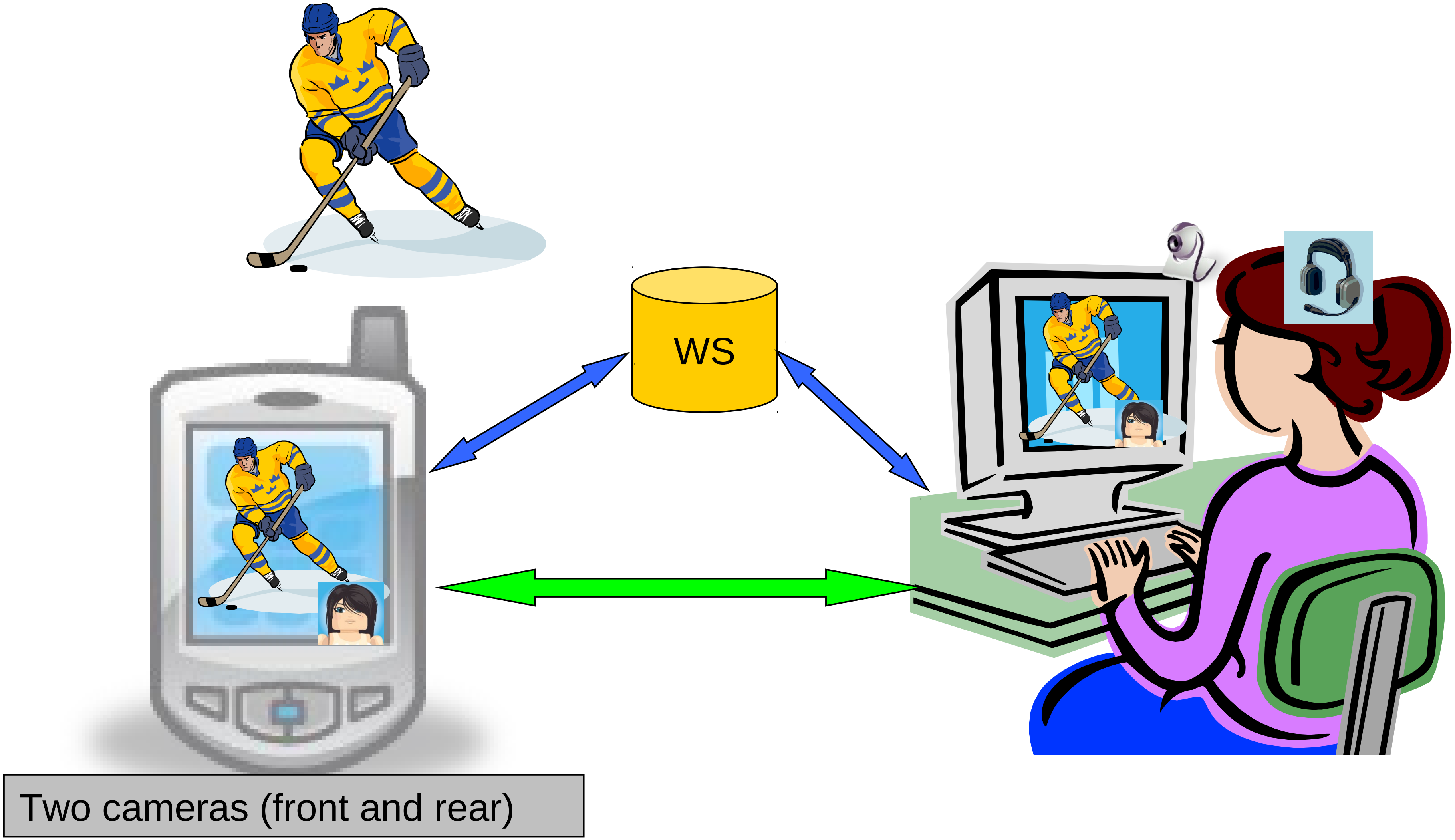
- HW A
- OS A
- Browser A

- HW B
- OS B
- Browser B

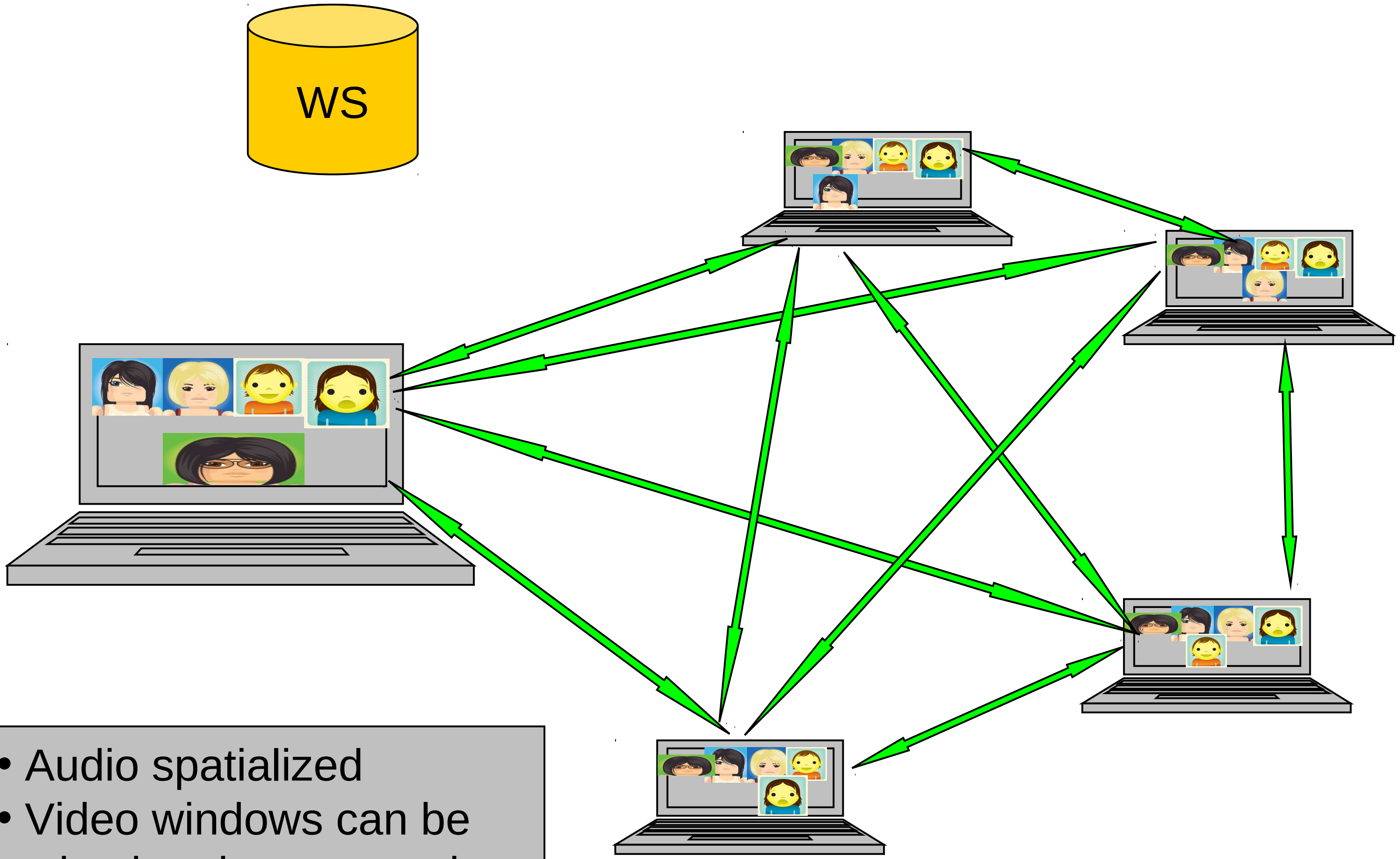
Simple Video Communication Service with inter-operator calling



Hockey game viewer

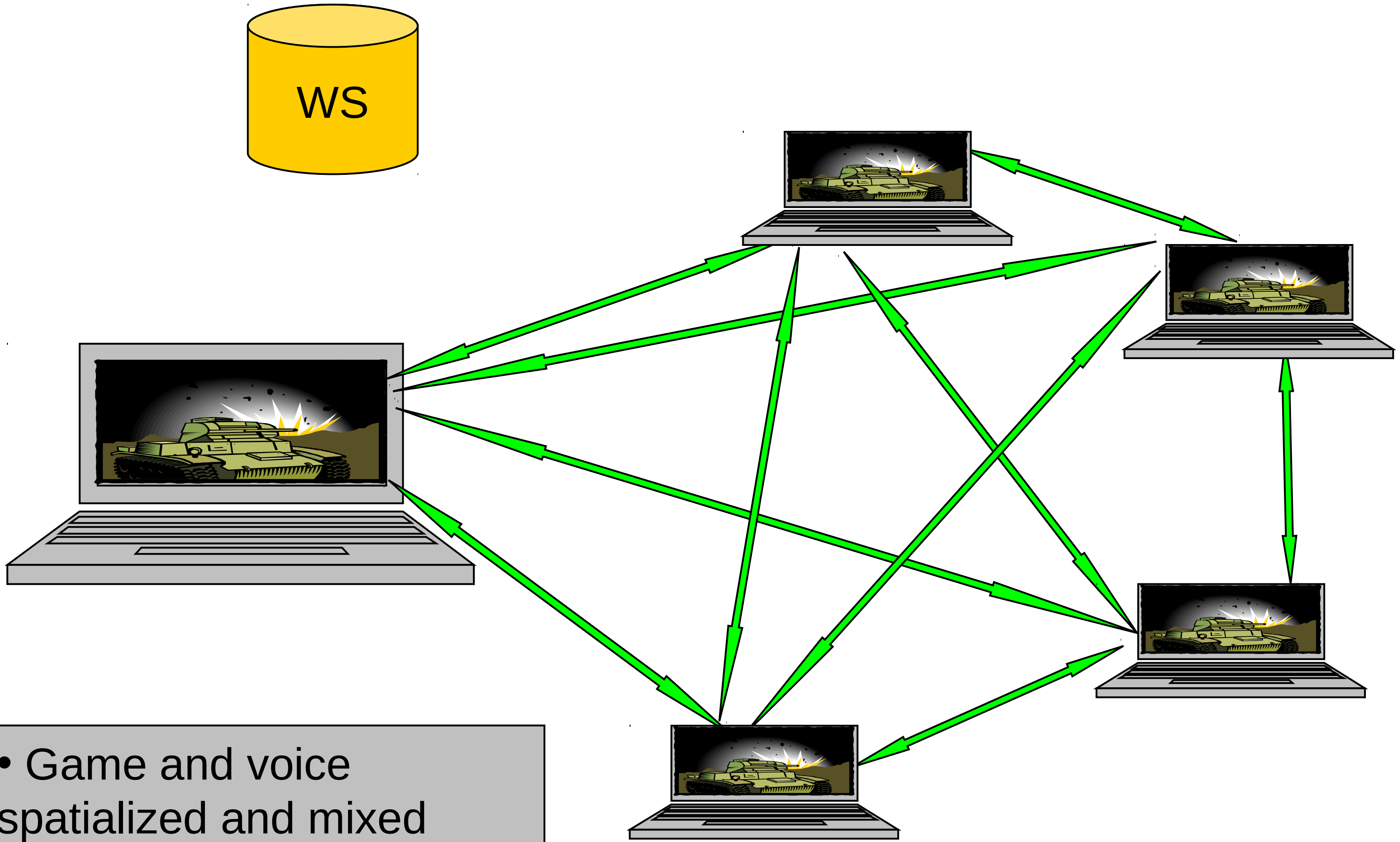


Multiparty video comm



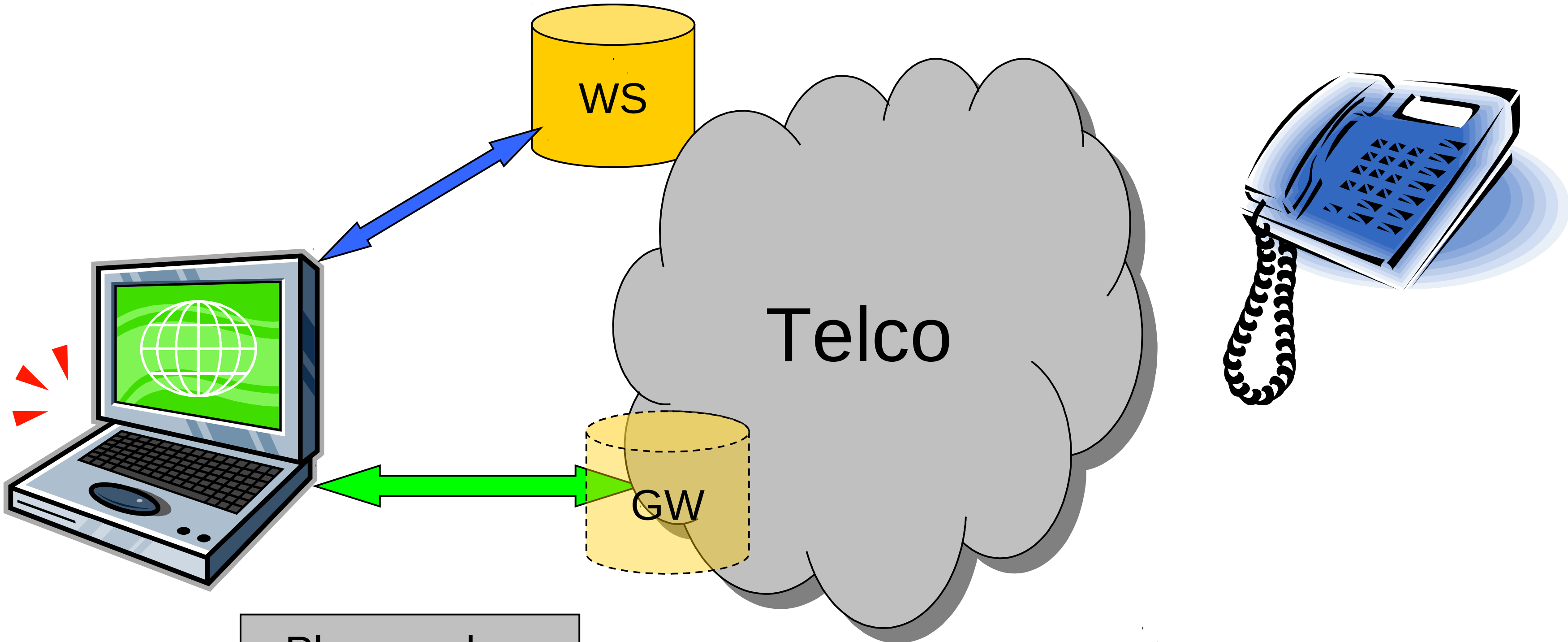
- Audio spatialized
- Video windows can be resized and rearranged

Multiparty on-line game with voice comm



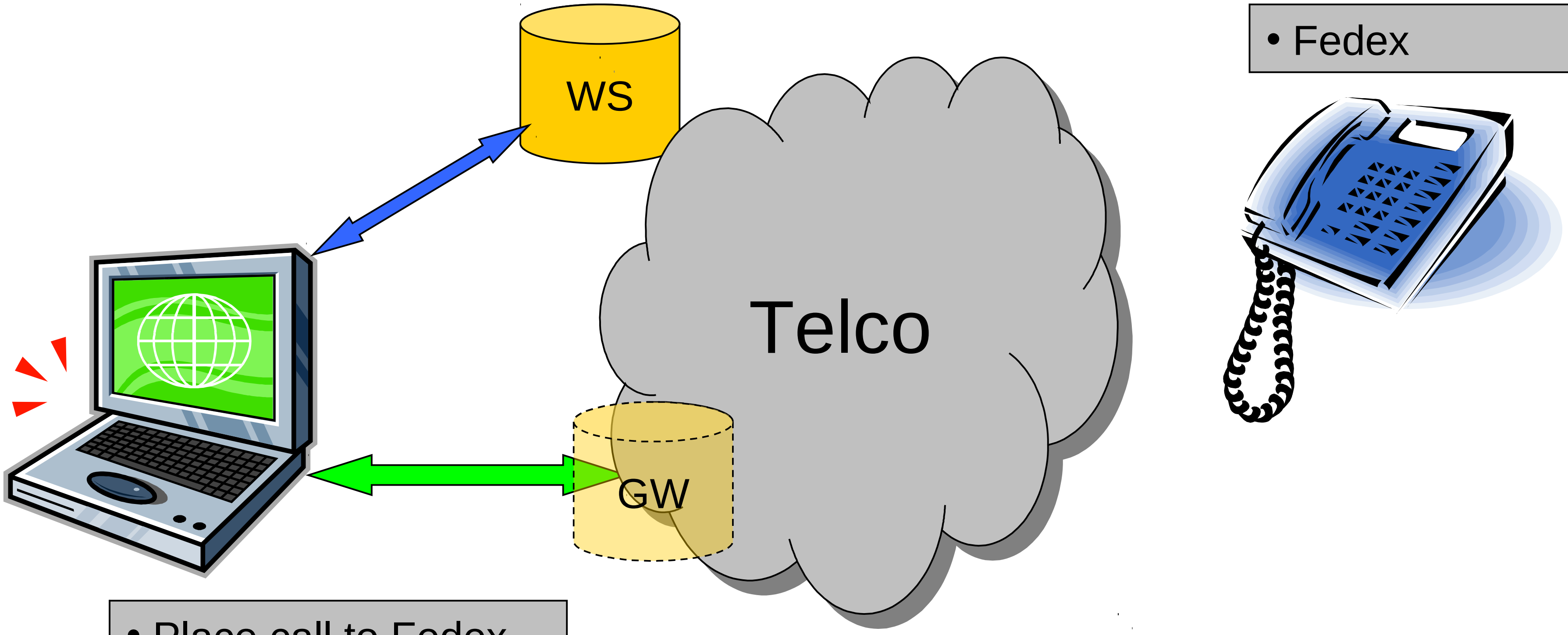
• Game and voice spatialized and mixed
• Game scene updates go peer-to-peer (latency)

Telephony terminal



- Place and receive calls

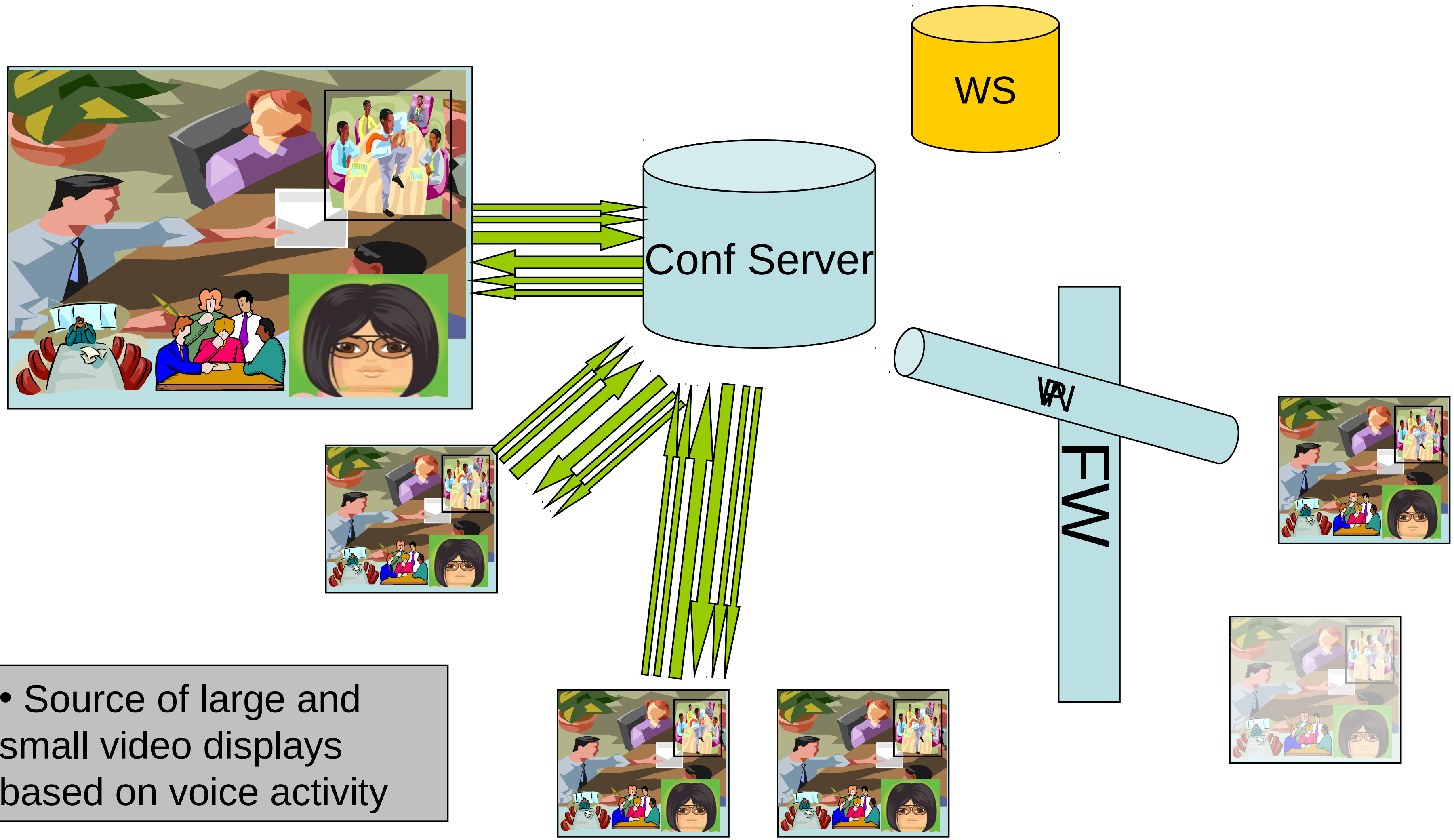
Fedex call



- Fedex

- Place call to Fedex
- Navigate Interactive Voice Response (IVR)

Video conf with server



- Source of large and small video displays based on voice activity

Proposed new use cases

- Explicit video resize use case
- Distributed music band
  - Treat mic signal as “audio”, not “speech”
- E911
  - Same so far (but would extend to QoS, location, …)
- Different TURN provider use cases
- Recording
- (Other IETF WG use cases to be browsed)

Possible documents

- One joint use case doc
- One IETF rtcweb reqs doc
- One W3C webrtc reqs doc

Simple Video Communication Service

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

Simple Video Communication Service

- 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.
- F10 The browser MUST support synchronization of audio and video.
- 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
- F25 The browser MUST support a baseline audio and video codec

Simple Video Communication Service, access change

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

Simple Video Communication Service, QoS

- F21 The browser MUST be able to take advantage of capabilities to prioritize voice and video appropriately.

Simple Video Communication Service with inter-operator calling

- F24 The browser MUST be able to initiate and accept a media session where the data needed for establishment can be carried in SIP. (Error in doc – not listed)

Hockey game viewer

- F14 The browser MUST be able to render several concurrent video streams
- (Missing req: The browser must be able to use several input devices and to generate and transmit several streams)

Multiparty video comm

- 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.

Multiparty on-line game with voice comm

- 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).
- F20 The browser must be able to send short latency datagram traffic to a peer browser

Telephony terminal

- F18 The browser MUST support an audio media format (codec) that is commonly supported by existing telephony services. QUESTION: G.711, AMR, G.719?

Fedex call

- F19 there should be a way to navigate the IVR

Video conf with server

- F7 The browser MUST support fast stream switches.
- F16 Streams MUST be able to pass through restrictive firewalls.
- F17 It MUST be possible to protect streams from eavesdropping.
- (Missing req: simulcast)

Security Consideration

- 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.
- The browser is expected to provide mechanisms for users to revice 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.

Not covered

- Security...
- Signaling reqs
- Authentication (towards STUN/TURN server)

Simple Video Communication Service

- 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.

Simple Video Communication Service

- 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.

Multiparty video comm

- A14 It MUST be possible for the web application to control panning, mixing and other processing for individual streams.

Hockey game viewer

- A15 The web application MUST be able to identify the context of a stream.

Multiparty on-line game with voice comm

- A16 It MUST be possible for the web application to send and receive datagrams to/from peer