RTCWEB Working Group	C.H. Holmberg
Internet-Draft	S.H. Hakansson
Intended status: Informational	G.E. Eriksson
Expires: March 04, 2012	Ericsson
	September 01, 2011

Web Real-Time Communication Use-cases and Requirements

draft-ietf-rtcweb-use-cases-and-requirements-04.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

*2. Conventions

*3. Definitions

*4. Use-cases

*4.1. Introduction

*4.2. Browser-to-browser use-cases

*4.2.1. Simple Video Communication Service

*4.2.1.1. Description

*4.2.1.2. Derived Requirements

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

*4.2.2.1. Description

*4.2.2.2. Derived Requirements

*4.2.3. Simple Video Communication Service, access change

*4.2.3.1. Description

*4.2.3.2. Derived Requirements

*4.2.4. Simple Video Communication Service, QoS

*4.2.4.1. Description

*4.2.4.2. Derived Requirements

*4.2.5. Simple video communication service with inter-operator calling

*4.2.5.1. Description

*4.2.5.2. Derived requirements

*4.2.6. Hockey Game Viewer

*4.2.6.1. Description

*4.2.6.2. Derived Requirements

*4.2.7. Multiparty video communication

*4.2.7.1. Description

*4.2.7.2. Derived Requirements

*4.2.8. Multiparty on-line game with voice communication

*4.2.8.1. Description

*4.2.8.2. Derived Requirements

*4.2.9. Distributed Music Band

*4.2.9.1. Description

*4.2.9.2. Derived Requirements

*4.3. Browser - GW/Server use cases

*4.3.1. Telephony terminal

*4.3.1.1. Description

*4.3.1.2. Derived Requirements

*4.3.2. Fedex Call

*4.3.2.1. Description

*4.3.2.2. Derived Requirements

*4.3.3. Video conferencing system with central server

*4.3.3.1. Description

*4.3.3.2. Derived Requirements

*5. Requirements

*5.1. General

*5.2. Browser requirements

*5.3. API requirements

*6. IANA Considerations

*7. Security Considerations

*7.1. Introduction

*7.2. Browser Considerations

*7.3. Web Application Considerations

*8. Additional use-cases

*9. Acknowledgements

*10. Change Log

*11. References

*11.1. Normative References

*11.2. Informative References

*Authors' Addresses

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. 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.

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.
During session establishment a self-view is displayed, and once the session has been established the video sent from the remote peer is displayed displayed in addition to the self-view. The users can during the session select to remove, and re-insert the self-view. The users can change the sizes of the video displays during the session. The users can also pause sending of media (audio, video, or both), and mute incoming media.
It is essential that the communication can not be eavesdropped.
Any session participant can end the session at any time.
The users are using communication devices of different makes, with different operating systems and browsers from different vendors.
One user has an unreliable Internet connection. It sometimes has packet losses, and is sometimes goes down completely.
One user is located behind a Network Address Translator (NAT).

4.2.1.2. Derived Requirements

F1, F2, F3, F4, F5, F6, F8, F9, F10, F17, F22, F25
A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A13

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 previos one. 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, F17, F22, F23, F25, F26
A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A13

4.2.3. Simple Video Communication Service, access change

4.2.3.1. Description

This use-case is almost identical to "4.2.1 Simple Video Communication Service". 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 access 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.

4.2.3.2. Derived Requirements

F1, F2, F3, F4, F5, F6, F8, F9, F10, F17, F22, F23, F25
A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A13

4.2.4. Simple Video Communication Service, QoS

4.2.4.1. Description

This use-case is almost identical to the previos one. The use of 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.

4.2.4.2. Derived Requirements

F1, F2, F3, F4, F5, F6, F8, F9, F10, F17, F21, F22, F23, F25
A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A13

4.2.5. Simple video communication service with inter-operator calling

4.2.5.1. 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 "4.2.1 Simple Video Communication Service" is available.
The same issues with connectivity apply.

4.2.5.2. Derived requirements

F1, F2, F3, F4, F5, F6, F8, F9, F10, F17, F22, F24, F25
A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A13

4.2.6. Hockey Game Viewer

4.2.6.1. 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, 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 video communication. 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 ot 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.
It is essential that the communication can not be eavesdropped.

4.2.6.2. Derived Requirements

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

4.2.7. Multiparty video communication

4.2.7.1. 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 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.
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 can not be eavesdropped.
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 and mix the audio from all received streams locally in the browser.

4.2.7.2. Derived Requirements

F1, F2, F3, F4, F5, F6, F8, F9, F10, F11, F12, F13, F14, F17, F22
A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A13, A14, A15

4.2.8. Multiparty on-line game with voice communication

4.2.8.1. 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.
It is essential that the communication can not be eavesdropped.
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.

4.2.8.2. Derived Requirements

F1, F2, F3, F4, F5, F6, F8, F9, F11, F12, F13, F15, F17, F20
A1, A2, A3, A4, A5, A7, A8, A9, A10, A11, A12, A13, A14, A15, A16

4.2.9. Distributed Music Band

4.2.9.1. 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.).

4.2.9.2. Derived Requirements

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

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 used.
It is essential that the communication can not be eavesdropped.

4.3.1.2. Derived Requirements

F1, F2, F3, F4, F5, F6, F8, F9, F10, F17, F18
A1, A2, A3, A4, A7, A8, A9, A10, A11, A12, A13

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, F18, F19
A1, A2, A3, A4, A7, A8, A9, A10, A11, A12, A13

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 browsers 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 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.
All participant 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 exists 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, the server just selects, c) each browser sends just an high resolution stream, the server trancodes into low reslution streams as required.

4.3.3.2. 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

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 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               there should be a way to navigate
                  the IVR
----------------------------------------------------------------
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
----------------------------------------------------------------
F26               The browser MUST be able to send streams to a
```

peer in presence of NATs that block UDP traffic.

5.3. API requirements

REQ-ID	DESCRIPTION
A1	The web application MUST be able to ask the browser for permission to use 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 provided with an identifier for the stream that can be communicated to the other party of the communication, and which the other party can associate with its end of the same stream.
A16	It MUST be possible for the web application to send and receive datagrams to/from peer
A17	It MUST be possible for the web application to indicate the type of audio signal (speech, audio)
A18	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, 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.

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 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. 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. Different TURN provider scenarios (Cullen Jennings); discussed at the webrtc meeting

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

4. Local Recording and Remote recording (John): Discussed a _lot_ on the mail lists (rtcweb as well as public-webrtc) late August 2011. Not concluded at time of writing.

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

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

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

8. Remote assistance (ala VNC or RDP) - User is helping another user on their computer with either view-only or view-with-control, either of just the browser of the the entire screen. http://www.ietf.org/mail-archive/web/rtcweb/current/msg00543.html

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

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

9. Acknowledgements

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-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 desrciption/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/FW 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 communciation)

*- Changed the fromulation 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

*- 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

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.

11.2. Informative References

, "

[webrtc_reqs]	Webrt requirements, http://dev.w3.org/2011/webrtc/editor/webrtc_reqs.html", .

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