Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 17, 2014

H. Alvestrand
Google

February 13, 2014

WebRTC MediaStream Identification in the Session Description Protocol
draft-ietf-mmusic-msid-04

Abstract

This document specifies a grouping mechanism for RTP media streams that can be used to specify relations between media streams.

This mechanism is used to signal the association between the SDP concept of "m-line" and the WebRTC concept of "MediaStream" / "MediaStreamTrack" using SDP signaling.

This document is a work item of the MMUSIC WG, whose discussion list is mmusic@ietf.org.

Requirements Language

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 RFC 2119 [RFC2119].

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 17, 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 3
1.1. Structure Of This Document 3
1.2. Why A New Mechanism Is Needed 3
1.3. Application to the WEBRTC MediaStream 4
2. The Msid Mechanism 4
3. The Msid-Semantic Attribute 5
4. Applying Msid to WebRTC MediaStreams 6
4.1. Handling of non-signalled tracks 7
5. WebRTC MediaStream Control 8
6. IANA Considerations 9
7. Security Considerations 11
8. Acknowledgements 11
9. References 11
9.1. Normative References 11
9.2. Informative References 12
Appendix A. Design considerations, open questions and and alternatives 12
Appendix B. Usage with multiple MediaStreams per M-line 13
B.1. Mechanism design with multiple SSRCs 13
B.2. Usage with the SSRC attribute 14
Appendix C. Change log 14
C.1. Changes from alvestrand-rtcweb-msid-00 to -01 14
C.2. Changes from alvestrand-rtcweb-msid-01 to -02 15
C.3. Changes from alvestrand-rtcweb-msid-02 to mmusic-msid-00 15
C.4. Changes from alvestrand-mmusic-msid-00 to -01 15
C.5. Changes from alvestrand-mmusic-msid-01 to -02 15
C.6. Changes from alvestrand-mmusic-msid-02 to ietf-mmusic-00 15
C.7. Changes from mmusic-msid-00 to -01 16
C.8. Changes from mmusic-msid-01 to -02 16
C.9. Changes from mmusic-msid-02 to -03 16
C.10. Changes from mmusic-msid-03 to -04 16
Author's Address 16

1. Introduction

1.1. Structure Of This Document

This document adds a new grouping relation between M-lines that can associate application layer identifiers with the binding between media streams, attaching identifiers to the media streams and attaching identifiers to the groupings they form.

Section 1.2 gives the background on why a new mechanism is needed.

Section 2 gives the definition of the new mechanism.

Section 4 gives the application of the new mechanism for providing necessary semantic information for the association of MediaStreamTracks to MediaStreams in the WebRTC API .

1.2. Why A New Mechanism Is Needed

When media is carried by RTP [RFC3550], each RTP media stream is distinguished inside an RTP session by its SSRC; each RTP session is distinguished from all other RTP sessions by being on a different transport association (strictly speaking, 2 transport associations, one used for RTP and one used for RTCP, unless RTCP multiplexing [RFC5761] is used).

SDP gives a description based on m-lines. According to the model used in [I-D.roach-mmusic-unified-plan], each m-line describes exactly one media source, and if mulitple media sources are carried in an RTP session, this is signalled using BUNDLE [I-D.ietf-mmusic-sdp-bundle-negotiation]; if BUNDLE is not used, each media source is carried in its own RTP session.

There exist cases where an application using RTP and SDP needs to signal some relationship between RTP media streams that may be carried in either the same RTP session or different RTP sessions. For instance, there may be a need to signal a relationship between a video track and an audio track, and where the generator of the SDP does not yet know if they will be carried in the same RTP session or different RTP sessions.

The SDP grouping framework [RFC5888] can be used to group m-lines. However, there is sometimes the need for an application to specify some application-level information about the association between the m-line and the group. This is not possible using the SDP grouping framework.

1.3. Application to the WEBRTC MediaStream

The W3C WebRTC API specification [W3C.WD-webrtc-20120209] specifies that communication between WebRTC entities is done via MediaStreams, which contain MediaStreamTracks. A MediaStreamTrack is generally carried using a single SSRC in an RTP session (forming an RTP media stream. The collision of terminology is unfortunate.) There might possibly be additional SSRCs, possibly within additional RTP sessions, in order to support functionality like forward error correction or simulcast. This complication is ignored below.

In the RTP specification, media streams are identified using the SSRC field. Streams are grouped into RTP Sessions, and also carry a CNAME. Neither CNAME nor RTP session correspond to a MediaStream. Therefore, the association of an RTP media stream to MediaStreams need to be explicitly signaled.

The WebRTC work has come to agreement (documented in [I-D.roach-mmusic-unified-plan]) that one M-line is used to describe each MediaStreamTrack, and that the BUNDLE mechanism [I-D.ietf-mmusic-sdp-bundle-negotiation] is used to group MediaStreamTracks into RTP sessions. Therefore, the need is to specify the ID of a MediaStreamTrack and its containing MediaStream for each M-line, which can be accomplished with a media-level attribute.

This usage is described in Section 4.

2. The Msid Mechanism

This document registers a new SDP [RFC4566] media-level "msid" attribute. This new attribute allows endpoints to associate RTP media streams that are carried in the same or different m-lines, as well as allowing application-specific information to the association.

The value of the "msid" attribute consists of an identifier and optional application-specific data, according to the following ABNF [RFC5234] grammar:

```
; "attribute" is defined in RFC 4566.
attribute =/ msid-attr
msid-attr = "msid:" identifier [ " " appdata ]
identifier = token
appdata = token
```

An example MSID value for a group with the identifier "examplefoo" and application data "examplebar" might look like this:

```
msid:examplefoo examplebar
```

The identifier is a string of ASCII characters chosen from 0-9, a-z, A-Z and - (hyphen), consisting of between 1 and 64 characters. It MUST be unique among the identifier values used in the same SDP session. It is RECOMMENDED that is generated using a random-number generator.

Application data is carried on the same line as the identifier, separated from the identifier by a space.

The identifier uniquely identifies a group within the scope of an SDP description.

There may be multiple msid attributes on a single m-line. There may also be multiple m-lines that have the same value for identifier and application data.

Endpoints can update the associations between RTP media streams as expressed by msid attributes at any time; the semantics and restrictions of such grouping and ungrouping are application dependent.

3. The Msid-Semantic Attribute

A session-level attribute is defined for signaling the semantics associated with an msid grouping. This allows msid groupings with different semantics to coexist.

This OPTIONAL attribute gives the group identifier and its group semantic; it carries the same meaning as the ssrc-group-attr of RFC 5576 section 4.2, but uses the identifier of the group rather than a list of SSRC values.

This attribute MUST be present if "a=msid" is used.

An empty list of identifiers is an indication that the sender understands the indicated semantic, but has no msid groupings of the given type in the present SDP.

An identifier of "*" is an indication that all "a=msid" lines in the SDP have this specific semantic.

The ABNF of msid-semantic is:

```
attribute =/ msid-semantic-attr
msid-semantic-attr = "msid-semantic:" token identifier-list
identifier-list = (" " identifier)* / " *"
token = <as defined in RFC 4566>
```

The semantic field may hold values from the IANA registriy "Semantics for the msid-semantic SDP attribute" (which is defined by this memo).

An example msid-semantic might look like this, if a semantic LS was registered by IANA for the same purpose as the existing LS grouping semantic:

```
a=msid-semantic:LS xyzzy forolow
```

This means that the SDP description has two lip sync groups, with the group identifiers xyzzy and forolow, respectively.

4. Applying Msid to WebRTC MediaStreams

This section creates a new semantic for use with the framework defined in Section 2, to be used for associating m-lines representing MediaStreamTracks within MediaStreams as defined in [W3C.WD-webrtc-20120209].

The semantic token for this semantic is "WMS" (short for WebRTC Media Stream).

The value of the msid corresponds to the "id" attribute of a MediaStream.

The appdata for a WebRTC MediaStreamTrack consists of the "id" attribute of a MediaStreamTrack.

If two different m-lines have MSID attributes with the same value for identifier and appdata, it means that these two m-lines are both intended for the same MediaStreamTrack. So far, no semantic for such a mixture have been defined, but this specification does not forbid the practice.

When an SDP description is updated, a specific msid continues to refer to the same MediaStream. Once negotiation has completed on a session, there is no memory; an msid value that appears in a later negotiation will be taken to refer to a new MediaStream.

The following are the rules for handling updates of the list of m-lines and their msid values.

- When a new msid value occurs in the description, the recipient can signal to its application that a new MediaStream has been added.
- When a description is updated to have more m-lines with the same msid value, but different appdata values, the recipient can signal to its application that new MediaStreamTracks have been added to the media stream.
- When a description is updated to no longer list the msid value on a specific m-line, the recipient can signal to its application that the corresponding media stream track has been closed.
- When a description is updated to no longer list the msid value on any m-line, the recipient can signal to its application that the media stream has been closed.

In addition to signaling that the track is closed when it disappears from the SDP, the track will also be signaled as being closed when all associated SSRCs have disappeared by the rules of [RFC3550] section 6.3.4 (BYE packet received) and 6.3.5 (timeout).

The association between SSRCs and m-lines is specified in [I-D.roach-mmusic-unified-plan].

4.1. Handling of non-signalled tracks

Non-WebRTC entities will not send msid. This means that there will be some incoming RTP packets that the recipient has no predefined MediaStream id value for.

Handling will depend on whether or not any MSIDs are signaled in the relevant m-line(s). There are two cases:

- No msid-semantic:WMS attribute is present. The SDP session is assumed to be a backwards-compatible session. All incoming media, on all m-lines that are part of the SDP session, are assumed to belong to independent media streams, each with one track. The identifier of this media stream and of the media stream track is a randomly generated string; the label of this media stream will be set to "Non-WMS stream".
- An msid-semantic:WMS attribute is present. In this case, the session is WebRTC compatible, and the packets are either caused by a bug or by timing skew between the arrival of the media packets and the SDP description. These packets MAY be discarded, or they

MAY be buffered for a while in order to allow immediate startup of the media stream when the SDP description is updated. The arrival of media packets MUST NOT cause a new MediaStreamTrack to be signaled.

If a WebRTC entity sends a description, it MUST include the msid-semantic:WMS attribute, even if no media streams are sent. This allows us to distinguish between the case of no media streams at the moment and the case of legacy SDP generation.

It follows from the above that the WebRTC entity must have the SDP of the other party before it can decide correctly whether or not a "default" MediaStream should be created. RTP media packets that arrive before the remote party's SDP MUST be buffered or discarded, and MUST NOT cause a new MediaStreamTrack to be signalled.

It follows from the above that media stream tracks in the "default" media stream cannot be closed by signaling; the application must instead signal these as closed when the SSRC disappears according to the rules of RFC 3550 section 6.3.4 and 6.3.5.

NOTE IN DRAFT: Previous versions of this memo suggested adding all incoming SSRCs to a single MediaStream. This is problematic because we do not know if the SSRCs are synchronized or not before we learn the CNAME of the SSRCs, which only happens when an RTCP packet arrives. How to identify a non-WMS stream is still open for discussion - including whether it's necessary to do so. Using the stream label seems like an easy thing to do for debuggability - it's not signalled, and is intended for human consumption anyway.

Another alternative is to group the incoming media streams based on CNAME; this preseerves the synchronization semantics of CNAME, but means that one cannot signal the MediaStreamTrack before the CNAME of the SSRC is known (which will happen only on arrival of the relevant RTCP packet).

5. WebRTC MediaStream Control

A need has been identified for signalling from a receiver to a sender some information about the receiver's desires for the sender to take action on a MediaStream track.

This section describes such a mechanism. It is intended to be used for streams that are signalled using the semantics in section Section 4.

This mechanism consists of a single new field, "msid-control", which

can have the following values:

- msid-control: enable - the receiver positively acknowledges that it wants the content of this media stream track. This has the same semantics as leaving out the field altogether, but is specified for completeness.

- msid-control: disable - the receiver desires that the sender stops sending media on this track, but allows for a later round of negotiation to resume transmission. The sender and receiver are expected to continue including the corresponding SSRCs in RTCP reports, keeping the information on the SSRCs from timing out.

- msid-control: stop - the receiver desires that the sender stops sending media on this track, and guarantees that it will never ask for data to be sent on this track again. The sender is expected to stop reporting on the corresponding SSRCs, and MAY send a BYE message when it stops sending.

- msid-control: reject - the exact same semantics as for msid-control: stop apply, but this form is only used if the stream has never been enabled. The intended use is for support of rejecting a MediaStream, rather than stopping it (such a function has not been specified so far).

The msid-control field is significant only for the direction from the receiver to the sender; if a single m-line is used for MediaStreams in both directions, only the streams sent by the receiver of the SDP message will be affected.

If the MediaStream in both directions is cancelled by msid-control: stop or msid-control: reject, the m-line MAY be disabled by setting its port number to 0. If there is a MediaStream in use in either direction, whether it's enabled or disabled, the m-line port number MUST NOT be set to 0.

6. IANA Considerations

This document requests IANA to register the "msid" attribute and the "msid-control" attribute in the "att-field (media level only)" registry within the SDP parameters registry, according to the procedures of [RFC4566]

The required information for "msid" is:

- Contact name, email: IETF, contacted via mmusic@ietf.org, or a successor address designated by IESG
- Attribute name: msid
- Long-form attribute name: Media stream group Identifier
- The attribute value contains only ASCII characters, and is therefore not subject to the charset attribute.
- The attribute gives an association over a set of m-lines. It can be used to signal the relationship between a WebRTC MediaStream and a set of m-lines.
- The details of appropriate values are given in RFC XXXX.

The required information for "msid-control" is:

- Contact name, email: IETF, contacted via mmusic@ietf.org, or a successor address designated by IESG
- Attribute name: msid-control
- Long-form attribute name: Media stream control
- The attribute value contains only ASCII characters, and is therefore not subject to the charset attribute.
- The attribute states the desires of a recipient with regards to a WebRTC MediaStream in the context of an m-line.
- The details of appropriate values are given in RFC XXXX.

This document requests IANA to create a new registry called "Semantics for the msid-semantic SDP attribute", which should have exactly the same rules as for the "Semantics for the ssrc-group SDP attribute" registry (Expert Review), and to register the "WMS" semantic within this new registry.

The required information is:

- Description: WebRTC Media Stream, as given in RFC XXXX.
- Token: WMS
- Standards track reference: RFC XXXX

IANA is requested to replace "RFC XXXX" with the RFC number of this

document upon publication.

7. Security Considerations

An adversary with the ability to modify SDP descriptions has the ability to switch around tracks between media streams. This is a special case of the general security consideration that modification of SDP descriptions needs to be confined to entities trusted by the application.

If implementing buffering as mentioned in section Section 4.1, the amount of buffering should be limited to avoid memory exhaustion attacks.

No other attacks that are relevant to the browser's security have been identified that depend on this mechanism.

8. Acknowledgements

This note is based on sketches from, among others, Justin Uberti and Cullen Jennings.

Special thanks to Miguel Garcia and Paul Kyzivat for their work in reviewing this draft, with many specific language suggestions.

9. References

9.1. Normative References

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

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol (SDP)", RFC 5576, June 2009.

[W3C.WD-webrtc-20120209]
Bergkvist, A., Burnett, D., Jennings, C., and A. Narayanan, "WebRTC 1.0: Real-time Communication Between Browsers", World Wide Web Consortium WD WD-webrtc-20120209, February 2012, <http://www.w3.org/TR/2012/WD-webrtc-20120209>.

9.2. Informative References

[I-D.ietf-mmusic-sdp-bundle-negotiation]
Holmberg, C., Alvestrand, H., and C. Jennings, "Multiplexing Negotiation Using Session Description Protocol (SDP) Port Numbers", draft-ietf-mmusic-sdp-bundle-negotiation-04 (work in progress), June 2013.

[I-D.roach-mmusic-unified-plan]
Roach, A., Uberti, J., and M. Thomson, "A Unified Plan for Using SDP with Large Numbers of Media Flows", draft-roach-mmusic-unified-plan-00 (work in progress), July 2013.

[I-D.westerlund-avtcore-multiplex-architecture]
Westerlund, M., Perkins, C., and H. Alvestrand, "Guidelines for using the Multiplexing Features of RTP", draft-westerlund-avtcore-multiplex-architecture-03 (work in progress), February 2013.

[RFC5761] Perkins, C. and M. Westerlund, "Multiplexing RTP Data and Control Packets on a Single Port", RFC 5761, April 2010.

[RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", RFC 5888, June 2010.

Appendix A. Design considerations, open questions and and alternatives

This appendix should be deleted before publication as an RFC.

One suggested mechanism has been to use CNAME instead of a new attribute. This was abandoned because CNAME identifies a synchronization context; one can imagine both wanting to have tracks from the same synchronization context in multiple MediaStreams and wanting to have tracks from multiple synchronization contexts within one MediaStream (but the latter is impossible, since a MediaStream is defined to impose synchronization on its members).

Another suggestion has been to put the msid value within an attribute

of RTCP SR (sender report) packets. This doesn't offer the ability to know that you have seen all the tracks currently configured for a media stream.

There has been a suggestion that this mechanism could be used to mute tracks too. This is not done at the moment.

Discarding of incoming data when the SDP description isn't updated yet (section 3) may cause clipping. However, the same issue exists when crypto keys aren't available. Input sought.

There's been a suggestion that acceptable SSRCs should be signaled in a response, giving a recipient the ability to say "no" to certain SSRCs. This is not supported in the current version of this document.

Appendix B. Usage with multiple MediaStreams per M-line

This appendix is included to document the usage of msid as a source-specific attribute. Prior to the acceptance of the Unified Plan document, some implementations used this mechanism to distinguish between multiple MediaStreamTracks that were carried in the same M-line.

It reproduces some of the original justification text for this mechanism that is not relevant when Unified Plan is used.

B.1. Mechanism design with multiple SSRCs

When media is carried by RTP [RFC3550], each RTP media stream is distinguished inside an RTP session by its SSRC; each RTP session is distinguished from all other RTP sessions by being on a different transport association (strictly speaking, 2 transport associations, one used for RTP and one used for RTCP, unless RTCP multiplexing [RFC5761] is used).

There exist cases where an application using RTP and SDP needs to signal some relationship between RTP media streams that may be carried in either the same RTP session or different RTP sessions. For instance, there may be a need to signal a relationship between a video track in one RTP session and an audio track in another RTP session. In traditional SDP, it is not possible to signal that these two tracks should be carried in one session, so they are carried in different RTP sessions.

Traditionally, SDP was used to describe the RTP sessions, with one m-line being used to describe each RTP session. With the advent of

extensions like BUNDLE [I-D.ietf-mmusic-sdp-bundle-negotiation], this association may be more complex, with multiple m-lines being used to describe one RTP session; the rest of this document therefore talks about m-lines, not RTP sessions, when describing the signalling mechanism.

The SSRC grouping mechanism ("a=ssrc-group") [RFC5576] can be used to associate RTP media streams when those RTP media streams are described by the same m-line. The semantics of this mechanism prevent the association of RTP media streams that are spread across different m-lines.

The SDP grouping framework [RFC5888] can be used to group m-lines. When an m-line describes one and only one RTP media stream, it is possible to associate RTP media streams across different m-lines. However, if an m-line has multiple RTP media streams, using multiple SSRCs, the SDP grouping framework cannot be used for this purpose.

There are use cases (some of which are discussed in [I-D.westerlund-avtcore-multiplex-architecture]) where neither of these approaches is appropriate; In those cases, a new mechanism is needed.

In addition, there is sometimes the need for an application to specify some application-level information about the association between the SSRC and the group. This is not possible using either of the frameworks above.

B.2. Usage with the SSRC attribute

When the MSID attribute was used with the SSRC attribute, it had to be registered in the "Attribute names (source level)" registry rather than the "Attribute names (media level only)" registry, and the msid line was prefixed with "a=ssrc:<ssrc> ". Apart from that, usage of the attribute with SSRC-bound flows was identical with the current proposal.

Appendix C. Change log

This appendix should be deleted before publication as an RFC.

C.1. Changes from alvestrand-rtcweb-msid-00 to -01

Added track identifier.

Added inclusion-by-reference of draft-lennox-mmusic-source-selection for track muting.

Some rewording.

C.2. Changes from alvestrand-rtcweb-msid-01 to -02

Split document into sections describing a generic grouping mechanism and sections describing the application of this grouping mechanism to the WebRTC MediaStream concept.

Removed the mechanism for muting tracks, since this is not central to the MSID mechanism.

C.3. Changes from alvestrand-rtcweb-msid-02 to mmusic-msid-00

Changed the draft name according to the wishes of the MMUSIC group chairs.

Added text indicting cases where it's appropriate to have the same appdata for multiple SSRCs.

Minor textual updates.

C.4. Changes from alvestrand-mmusic-msid-00 to -01

Increased the amount of explanatory text, much based on a review by Miguel Garcia.

Removed references to BUNDLE, since that spec is under active discussion.

Removed distinguished values of the MSID identifier.

C.5. Changes from alvestrand-mmusic-msid-01 to -02

Changed the order of the "msid-semantic: " attribute's value fields and allowed multiple identifiers. This makes the attribute useful as a marker for "I understand this semantic".

Changed the syntax for "identifier" and "appdata" to be "token".

Changed the registry for the "msid-semantic" attribute values to be a new registry, based on advice given in Atlanta.

C.6. Changes from alvestrand-mmusic-msid-02 to ietf-mmusic-00

Updated terminology to refer to m-lines rather than RTP sessions when discussing SDP formats and the ability of other linking mechanisms to refer to SSRCs.

Changed the "default" mechanism to return independent streams after considering the synchronization problem.

Removed the space from between "msid-semantic" and its value, to be consistent with RFC 5576.

C.7. Changes from mmusic-msid-00 to -01

Reworked msid mechanism to be a per-m-line attribute, to align with [I-D.roach-mmusic-unified-plan]

C.8. Changes from mmusic-msid-01 to -02

Corrected several missed cases where the word "ssrc" was not changed to "M-line".

Added pointer to unified-plan (which should be moved to point to -jsep)

Removed suggestion that ssrc-group attributes can be used with "msid-semantic", it is now only the msid-semantic registry.

C.9. Changes from mmusic-msid-02 to -03

Corrected even more cases where the word "ssrc" was not changed to "M-line".

Added the functionality of using an asterisk (*) in the msid-semantic line, in order to remove the need for listing all msids in the msid-semantic line whne only one msid-semantic is in use.

Removed some now-unnecessary text.

C.10. Changes from mmusic-msid-03 to -04

Changed title to reflect focus on WebRTC MediaStreams

Added a section on receiver-side media stream control, using the "msid-control" attribute.

Author's Address

Harald Alvestrand
Google
Kungsbron 2
Stockholm, 11122
Sweden

Email: harald@alvestrand.no