In the context of [RTC-Web], some work is emerging to make real-time communications possible in a web browser. This document defines a minimal set of requirements so that such applications interoperate with SIP-RTP applications.

Status of this Memo

This Internet-Draft is submitted to IETF 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 13, 2011.

Copyright Notice

Copyright (c) 2011 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
2. Terminology .................................................. 3
3. Definitions .................................................. 4
4. Use cases ..................................................... 4
   4.1. SIP Multimedia application reachability extension .... 4
   4.2. RTC-Web applications with integration of SIP device ... 4
   4.3. RTC-Web and SIP service provider interconnection .... 4
5. Possible RTC-Web/SIP interworking architectures .......... 4
   5.1. SIP-RTP Stack in the Browser ......................... 5
   5.2. New Signalling Scheme and RTP Stack in the Browser ... 5
   5.3. New Signalling Scheme and New Media Protocol in the
        Browser ................................................... 7
   5.4. Analysis with regards to interworking ................ 8
6. Requirements .................................................. 9
   6.1. Generic requirements: ................................... 9
   6.2. Signalling level requirements: ........................ 9
   6.3. Media level requirements: ................................ 9
   6.4. Codec level requirements: ................................ 10
7. Security Considerations ...................................... 10
8. IANA Considerations .......................................... 10
9. Acknowledgements ............................................ 10
10. References ................................................... 10
    10.1. Normative references .................................... 10
    10.2. Informative references .................................. 11
Authors’ Addresses .............................................. 11
1. Introduction

In the context of [RTC-Web], some work is emerging to make real-time communications possible in a web browser. Such work will allow to use the UDP protocol to transport real time data.

This document defines a minimal set of requirements so that RTC-Web applications interoperate with applications based on SIP ([RFC3261]) and RTP ([RFC3550]) protocols.

On the one hand, bringing real-time communication capability in the web browser RTC-Web promises to offer great value to end-users, developers and service providers. This value comes from the ubiquity of the web browser and the web architecture, from the simple programatic model the web offer and indeed the innovation perspective of such solution.

On the other hand, SIP and RTP protocols are broadly used to implement real-time or near real-time applications. This is particularly true for voice, video, instant messaging, presence and content sharing and when considering available implementations in devices (hard phone, mobile phone...), network infrastructures (e.g. SIP based architecture) and service provider interconnection gateways.

Allowing both solutions to interoperate promises RTC-Web solution to have greater value as it will allow this solution to reach legacy SIP multimedia devices and networks and vice versa.

Section 4 describes some use cases. Section 5 reports the different possible architectures. Section 6 finally states a set of requirements the ongoing RTC-Web solution definition should fulfil to be able to interoperate with SIP based multimedia applications.

Note: This document does not directly address RTC-Web service provider interconnection except if this interconnection is based on SIP-RTP.

2. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in RFC 2119 [RFC2119].
3. Definitions

RTC-WEB: the term [RTC-Web] refers to the ongoing work on defining a solution that enables real time applications such as bidirectional audio and video within web applications.

SIP-RTP: the term SIP-RTP is a generic term which refers to SIP and RTP ecosystem protocol stack. This includes, non exhaustively: SIP, SDP, RTP, RTCP, SRTP...

Note: To be adjusted to fit with the current on going [RTC-Web] charter.

4. Use cases

This section presents some use cases involving interworking between RTC-Web and SIP applications. These use cases include scenarios where real-time audio and/or video are exchanged.

4.1. SIP Multimedia application reachability extension

Alice wants to access its services. Her service provider A (e.g. atlanta.example.org) hosts these services on SIP-RTP servers. Alice can use a web browser implementing an RTC-Web extension to reach its service.

4.2. RTC-Web applications with integration of SIP device

Bob wants to access its services. His service provider B (e.g. biloxi.example.org) hosts these services on RTC-Web servers. Bob can use a device implementing an SIP-RTP extension to reach its service.

4.3. RTC-Web and SIP service provider interconnection

All the users of service provider A (e.g. atlanta.example.org) use an RTC-Web application. All the users of service provider B (e.g. biloxi.example.org) use a SIP-RTP application. Both service providers want to make communications possible between all these users.

This use case is typically an inter service operators use case.

5. Possible RTC-Web/SIP interworking architectures

This section outlines different architectures to realize RTC-Web/SIP-RTP interworking. This section does not pretend to be exhaustive.
in term of architecture description but intends to propose families of models any kind of solution should fit in.

These architectures satisfy the use cases listed above. However, it must be noted that depending on the considered use case, additional components may be necessary.

In this section, the name SIP used alone is a shortcut for SIP and SDP protocols. Similarly, RTP used alone is a shortcut for RTP, RTCP, and SRTP protocols.

5.1. SIP-RTP Stack in the Browser

This architecture consists in directly implementing a SIP-RTP protocol stack in the browser, enabling a direct connection between an RTC-Web application in a browser and a SIP-RTP phone.

Architecture with SIP-RTP in the browser:

```
+-----------------+                           +-----------------+
|     RTC-Web     |                           |     SIP-RTP     |
|   application   |                           |   application   |
|-----------------|                           |                 |
|rtcweb    rtcweb |                           |                 |
| sig      media  |                           |                 |
| APIs     APIs   |                           |                 |
|-----------------|                           |                 |
|                 |                           |                 |
| -----           |                           | -----           |
| || SIP |          |<-----------SIP----------->|| SIP |          |
||stack|          |                           ||stack|          |
| -----           |                           | -----           |
|           ----- |                           |           ----- |
|          | RTP ||<-----------RTP----------->|          | RTP ||
|          |stack||                           |          |stack||
|           ----- |                           |           ----- |
-----------------                             -----------------
```

Figure 1

5.2. New Signalling Scheme and RTP Stack in the Browser

This architecture consists in implementing a new signalling scheme and an RTP stack in the browser.

A new signalling scheme means refers to two possible models:
- Session management data set by API and transported by an application protocol (e.g. HTTP or WebSockets). Figure 2 illustrates such architecture with XXX as the session management data. The HTTP stack shown in the figure is the regular HTTP stack available by default in all web browsers. Having SIP (or part of it) embedded in HTTP in one possible implementation, as indicated in [draft-sinnreich-sip-web-apis-01].
- A session management protocol different from SIP (e.g. XMPP, MEGACO). Figure 3 illustrates such architecture with YYY as the signalling protocol.

Both models relax constraints on the technology choice to implement the RTC-Web solution but add constraints on end-to-end compatibility with SIP-RTP applications by requiring the implementation of a gateway to map one protocol into another one.

**Architecture with HTTP in the browser:**

```
-----------------                            -----------------
|     RTC-Web     |                          |     SIP-RTP     |
|   application   |                          |   application   |
-----------------|                          |                 |
|rtcweb    rtcweb |
| sig      media |
| APIs     APIs |
|--|--       |---|---------|---|   XXX                    |                 |
| XXX     | XXX |   |       |   over  --------         | -----           |
| HTTP |<--HTTP-->| HTTP|SIP|<-SIP-->| SIP |          |
| stack||SIP|stack|--|--       |                 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| RTP |<--------RTP--------->| RTP |
| stack|                 |stack|
-----------------                            -----------------
```

Figure 2
Architecture with another protocol than SIP or HTTP in the browser:

<table>
<thead>
<tr>
<th>RTC-Web application</th>
<th>SIP-RTP application</th>
</tr>
</thead>
<tbody>
<tr>
<td>rtcweb</td>
<td>rtcweb</td>
</tr>
<tr>
<td>sig</td>
<td>media</td>
</tr>
<tr>
<td>APIs</td>
<td>APIs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>YYY stack</th>
<th>YYY SIP stack GW</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTP stack</td>
<td>RTP stack</td>
</tr>
</tbody>
</table>

Figure 3

5.3. New Signalling Scheme and New Media Protocol in the Browser

This architecture consists in implementing different protocols in RTC-Web and SIP-RTP frameworks, both for at the signalling level and at the media level.

Such architecture requires interworking work (protocol mapping, gateway) both for the signalling and the media protocols.

This architecture relaxes constraints on the technology choice to implement the RTC-Web solution but adds constraints on end-to-end compatibility with SIP-RTP applications by requiring the implementation of gateway(s) to adapt protocols and media payloads.
Architecture with another protocol than RTP as a media protocol:

<table>
<thead>
<tr>
<th>RTC-Web application</th>
<th>SIP-RTP application</th>
</tr>
</thead>
<tbody>
<tr>
<td>rtcweb</td>
<td>rtcweb</td>
</tr>
<tr>
<td>sig</td>
<td>media</td>
</tr>
<tr>
<td>APIs</td>
<td>APIs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sig. stack</th>
<th>&lt;-sig-&gt;</th>
<th>sig</th>
<th>SIP</th>
<th>&lt;-SIP-&gt;</th>
<th>SIP stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>med. stack</td>
<td>&lt;-med-&gt;</td>
<td>med</td>
<td>RTP</td>
<td>&lt;-RTP-&gt;</td>
<td>RTP stack</td>
</tr>
</tbody>
</table>

Figure 4

5.4. Analysis with regards to interworking

Using a full SIP-RTP stack in the browser (Section 5.1) would undoubtedly be the best solution with regards to interworking: it would avoid specifying new protocols and it would thus avoid the control plane interworking problem described in [RFC3439] (i.e. no need for protocol mapping). It nevertheless requires a granular API to configure and access the protocol stack.

Using the RTP protocol suite but another than the SIP protocol (Section 5.2) add the burden of interworking efforts at the signalling level. The level of complexity of this gateway depends on how much the signaling protocol will look like SIP. However, having HTTP (or WebSocket) as a protocol transporting the signaling data is attractive due to the central role played by this protocol in Web environments.

Using both new signalling and media protocols in the browser (Section 5.3) has been presented above for the sake of exhaustiveness but this solution is not attractive for SIP-RTP interworking: it increases the interworking efforts by requiring work at the media level (new media protocol, complexity and cost of interworking gateways...), whereas adding no identified advantages with regards to the existing RTP/UDP protocol suite.
6. Requirements

Whatever the architecture solution the RTC-Web will retain, a reasonable way-forward is to specify its protocols and APIs taking care of interworking with SIP-RTP devices. As such the following requirements are proposed to the RTC-Web working group:

6.1. Generic requirements:

GENERIC-REQ-1 The [RTC-Web] solution MUST be designed in a such way it allows interworking with SIP-RTP applications both at the signalling and media level.

6.2. Signalling level requirements:

SIG-REQ-1 The [RTC-Web] solution MUST be designed in a way it allows interoperability with SIP based multimedia applications. This is typically applicable for identifiers, credentials, state machine, and message types.

SIG-REQ-2 The [RTC-Web] solution MUST include a way to negotiate media format as in Offer/Answer model used in SIP ([RFC3264])

SIG-REQ-3 The [RTC-Web] solution MUST include a way to interoperate with ([RFC5939])

SIG-REQ-4 The [RTC-Web] solution MUST allow end to end codec negotiation between RTC-web device and SIP device

SIG-REQ-5 The [RTC-Web] solution MUST include a compatibility/mapping with SDP([RFC4566])

SIG-REQ-6 The [RTC-Web] solution SHOULD NOT require SIP-RTP extensions.

6.3. Media level requirements:

MEDIA-REQ-1 The [RTC-Web] solution MUST be designed in a way it does not mandate a gateway at media level when interworking with SIP based multimedia application, consequently it must be based on RTP/RTCP protocol suite over UDP for real-time media.
MEDIA-REQ-2  The [RTC-Web] solution MUST be compatible with a media
gateway architecture and not rely exclusively on a peer
to peer (between RTC-Web devices)...

MEDIA-REQ-3  The [RTC-Web] solution MUST allow the configuration of
some media-related parameters per session (e.g. buffer
size, packetization...).

6.4. Codec level requirements:

CODEC-REQ-1  The [RTC-Web] solution MUST allow codecs available in
existing SIP-RTP applications. A non exhaustive list is
the following: G.711, G.722, AMR, AMR-WB, H.264.

7. Security Considerations

SEC-REQ-1  RTC-Web and SIP-RTP interworking solution MUST NOT
compromise inherent security feature(s) developed and used
for both RTC-Web and SIP-RTP solutions.

8. IANA Considerations

None.

9. Acknowledgements

Thank you to Bruno Chatras, Christophe Eyrignoux, and Sebastien
Cubaud who provided early feedback.

10. References
10.1. Normative references

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H.,
Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston,
10.2. Informative references


Authors' Addresses

Xavier Marjou
France Telecom Orange
2, avenue Pierre Marzin
Lannion 22307
France

Email: xavier.marjou@orange-ftgroup.com