Combination of IMS-based IPTV Services with WebRTC

Tilmann Bach
Hochschule fuer Telekommunikation Leipzig (HfTL)
University of Applied Sciences, Leipzig, Germany
Email: tilmann.bach@hftl.de

Michael Maruschke
Hochschule fuer Telekommunikation Leipzig (HfTL)
University of Applied Sciences, Leipzig, Germany
Email: maruschke@hftl.de

Jens Zimmermann
Hochschule fuer Telekommunikation Leipzig (HfTL)
University of Applied Sciences, Leipzig, Germany
Email: jens.zimmermann@hftl.de

Kay Hänsge
Telekom Innovation Laboratories
Deutsche Telekom AG
Berlin, Germany
Email: kay.haensge@telekom.de

Matthias Baumgart
Telekom Innovation Laboratories
Deutsche Telekom AG
Berlin, Germany
Email: matthias.baumgart@telekom.de

***Abstract*—This paper describes the potential to combine IP Multimedia Subsystem (IMS)-based IP Television (IPTV) services with a future-oriented web browser technology, the Web Real-Time Communication (WebRTC). To enrich the quality of experience for residential customers using this upcoming technology, the article focuses on the merging of the technical capabilities arising from both the IMS-based IPTV services and the WebRTC clients. Advantages of these different technologies are introduced and involved in the authors' concept. The presented proposal reuses the standardized IMS-based IPTV architecture as well as a state of the art WebRTC browser client. The ongoing WebRTC standardization process is considered. A first prototype has been developed successfully involving the Open IMS Core testbed, several IPTV typical components and the WebRTC client.**

***Keywords*-WebRTC; IMS-based IPTV; CoD; Telco**

I. Introduction

Nowadays, Telecommunication Carriers (Telcos) provide IP-based voice and video telephony services to their customers. In order to deploy an IP-based core network, an IMS-based infrastructure is increasingly used. With decreasing sales in the field of conventional legacy voice businesses, the Telcos are forced to develop new business areas. In order to offer new multimedia services like IPTV, it is reasonable to reuse the well established IMS-based core network infrastructure with Telco specific assets like guaranteed Quality of Service (QoS), service interoperability or mobility.

In the current state of the Internet, web technologies are steadily gaining popularity and browser-based real-time communication is an essential feature for future web applications. Using new web technologies like Hypertext Markup Language Version 5 (HTML5), WebSockets, etc., an easy and timely development of new web applications is possible. Furthermore, the new upcoming technology named WebRTC enriches ordinary web browsers with real-time communication functionalities [1]. With these new opportunities, it is possible to implement real-time communication applications within a web browser. Thus, from the end-user's point of view, the installation of separate communication software or browser plug-ins is not necessary anymore. Instead, the desired communication features can be used in the browser immediately.

A browser with WebRTC features is also capable to deal with real-time streaming data which is used in Content on Demand (CoD) services like video or audio on demand. The combination of browser-based streaming capabilities and IMS-based IPTV services generates benefits for both the end-user and the Telcos. End-users have the ability to enjoy the advantages of IMS-based services like session mobility, QoS or Single Sign On mechanisms and Telcos can deliver their own applications and new features to the customers easily and directly by using web applications instead of legacy clients.

This paper discusses the combination of IMS-based IPTV services like CoD with WebRTC clients. We propose a concept reusing the IMS-based IPTV architecture to offer the CoD service to WebRTC end-users while only requiring a standard web browser. To verify this proposal, a first proof of concept has been implemented.

The present paper is structured as follows: Section II offers an overview to the current status of the standardization of the considered technologies, IPTV based on IMS core networks and WebRTC. Section III describes the authors' concept to combine WebRTC technologies with the IPTV service. The architecture and their specifics are considered and the proof of concept is presented. In Section IV, a conclusion and next steps in evolving this idea are pointed out.

II. Status Quo

A. *IMS-based IPTV*

The European Telecommunications Standards Institute (ETSI) Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN) working group has standardized a comprehensive IMS-based IPTV architecture [2][3]. Using the core IMS subsystem, various IPTV functions are supported.

Copyright (c) IARIA, 2014. ISBN: 978-1-61208-346-9

The most common services and features described are Broadcast TV, Time Shifted TV, Content on Demand (CoD), Network-Personal Video Recorder (N-PVR), Pay-Per-View (PPV), Electronic Program Guide (EPG), parental control and advertising. Having finished this standardization process in 2011, ETSI TISPAN took note of the Telcos carrier grade network capabilities, such as high availability, QoS, mobility and even more. Most of the standardized IPTV architecture components make use of the common protocols utilized in an IMS-based network such as the Session Initiation Protocol (SIP), the Session Description Protocol (SDP), the Hypertext Transfer Protocol (HTTP) and the Real-Time Transport Protocol (RTP). Thus, it has big potential to combine the IMS-based IPTV services with established IMS real-time communication like voice and video telephony [4], presence [5] and other IMS services in order to create additional and more personalized value [6]. The standardized architecture of an IMS-based IPTV service is depicted in Figure 1 and is described as follows:

- Core IMS, core network components as specified in [7]
- Service Discovery Function (SDF), provides Service Attachment Information (SAI) with information about available services and related SSF
- Service Selection Function (SSF), provides Service Selection Information (SSI) containing the metadata of the available content.
- Service Control Function (SCF), is a SIP Application Server (AS) and the reference point for IMS UEs to start and control the IPTV sessions, moreover the SCF assigns the corresponding MCF and forwards the session information to it
- Media Control Function (MCF), controls media transport of MDF and receives instructions of SCF and UE
- Media Delivery Function (MDF), contains media data and transmits them to the UE
- User Equipment (UE), interacts as IMS-based IPTV end-user.

To realize IMS-based IPTV, ETSI [2] defines Generic IPTV Capabilities, a set of of typically general signaling functions like service discovery and service control. All corresponding interfaces and used protocols are described more detailed in Section 4 of ETSI [3].

B. Web real-time communication with WebRTC

WebRTC [8] is an open project initiated by Google Incorporated. The purpose of the project is to integrate voice and video real-time communication into standard web browsers. From the first implementations, both web developers and telecommunication companies saw the potential for the future and prototyped new and promising applications. Telcos perceive the project as a risk for their classical voice business and thus they are also interested in expanding their own telecommunication service portfolio with the new WebRTC technology.

WebRTC provides an Application Programming Interface (API) definition which enables real-time communication in web browsers without the need of any additional browser plugin or additional software [1]. It empowers the browser to capture video and voice inputs of the client's device. WebRTC is still in the standardization process. The World Wide Web Consortium

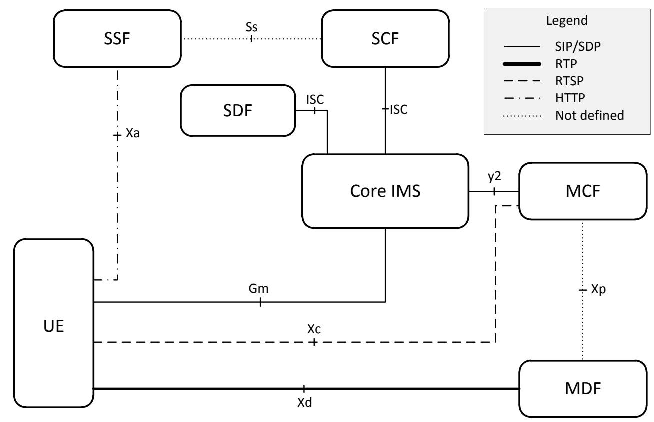

Figure 1: Simplified IMS-based IPTV functional architecture [3]

(W3C) is responsible for the web developer API and the Internet Engineering Task Force (IETF) for all corresponding protocols in an active working group named "Real-Time Communication in WEB-browsers - RTCweb" [9].

This browser extension enables developers to easily implement voice and video call web applications [10]. It also features components for file sharing. The browser implements the video, the voice and the transport engines. While there is still a discussion in the standardization process regarding the adequate audio and video codecs to be used current WebRTC implementations utilize the VP8 video compression format for video and the Opus codec for voice [11][12].

WebRTC does not define any particular signaling protocol. That is why developers can choose the most appropriate protocol for their special use case. So it is possible to implement new communication features, faster.

WebRTC requires secure transport of the RTP packets with the Secure Real-Time Transport Protocol (SRTP) [13] based on the mandatory to implement Datagram Transport Layer Security (DTLS) encryption protocol [14] used for key negotiation [15]. For solving Network Address Translation (NAT) problems, WebRTC also provides Session Traversal Utilities for NAT (STUN) [16], Traversal Using Relays around NAT (TURN) and Interactive Connectivity Establishment (ICE) [17] capabilities. WebRTC requires SDP for the negotiation of the session properties and uses the whole SDP's Offer/Answer Model. Furthermore, SDP is also used for exchanging

- The fingerprint of the certificate used in the DTLS Certificate exchange procedure
- And ICE specific parameters like the ICE Candidate objects.

The generic architecture of a WebRTC client is described by Alvestrand [18] and illustrated in Figure 2. The components can be described as follows:

- Web server, provides the web application to load and includes a server for the client to connect to for handling the whole signaling flow
- Browser, a generic web browser
- Web application, application source code executed by the web browser

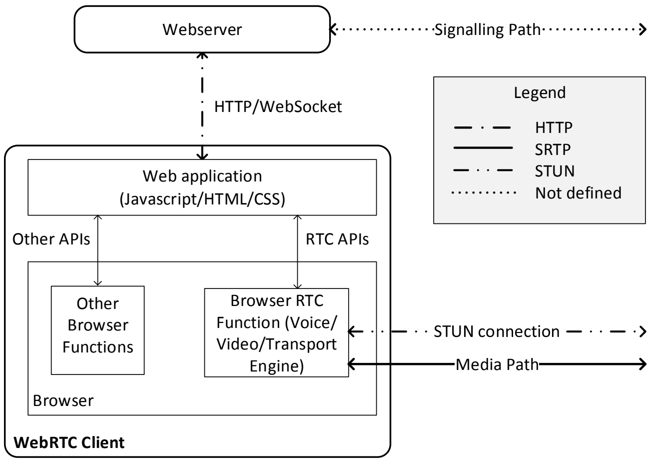

Figure 2: WebRTC client based on [18]

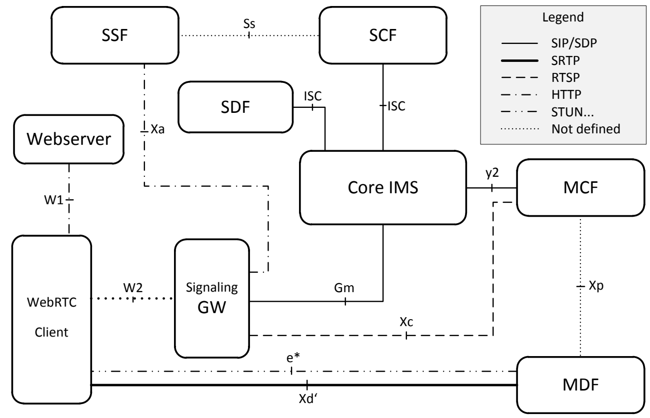

Figure 3: Architecture of proposed concept

- Browser RTC Function, WebRTC component in the web browser with voice, video and transport engines
- Signaling Path, is not specified but is needed to transfer the SDP information
- Media Path, transports the payload
- STUN Connection, is a mandatory component to bypass NAT restrictions.

For running a WebRTC client successfully, it is necessary to use a capable web browser. That means the browser has to implement the Browser RTC Function. Currently, web browsers like Google Chrome, Mozilla Firefox and Opera provide this component by default. Therefore, all devices which are able to run one of these browsers have the ability to use web based real-time communication. This includes all, desktop and tablet computers, laptops and smartphones. At the moment, there are restrictions in some operation systems like Apples iOS or Microsofts Windows Phone.

C. *WebRTC access to IMS network based architecture*

A 3rd Generation Partnership Project (3GPP) study suggests several solutions for accessing an IMS-based network architecture with WebRTC clients [19]. This clients could be connected either via wireless or wired access network technologies like Long Term Evolution (LTE), Wireless Local Area Network (WLAN) or any Digital Subscriber Line (xDSL). Overall, this study focusses on conversational real-time communication services like audio or video telephony. The technical report scopes the accessibility of typical IMS network characteristics or features like:

- Identity Management
- Accounting and Billing
- Interoperability with legacy networks like PSTN/ISDN
- Enabling of an application-oriented QoS (e.g., for voice telephony)

for Over The Top (OTT) applications such as plain WebRTC browser-to-browser applications if interconnected with a Telco network.

In contrast to [19], our paper is focused on IMS-based IPTV services, which should be made available to WebRTC Clients.

III. CONCEPT

A. *Consolidate IMS-based IPTV with WebRTC*

With the help of the Browser RTC Function (shown in Figure 2) the web browser is able to handle RTP packets without the need of any separate software or plug-in which was exposed in Section II. However, due to differences in the used media codecs and payload transport protocols the technical parameters of WebRTC and IMS-based IPTV do not match out-of-the-box. Assuming that the browser implementation of the WebRTC cannot be influenced by the Telcos, a modification of the architecture has to be enforced on the carrier side's network. This section handles this idea and introduces a potential architecture of such a consolidation.

B. *Architecture*

The proposed architecture is based on the simplified IMS-based IPTV functional architecture. Instead of an IMS IPTV UE the endpoint of this service is a WebRTC client. For the combination of both, a translation for the different signaling and user data is necessary in various network components to gain compatibility. The consolidated architecture of the proposed concept is depicted in Figure 3. Components and interfaces, which are new or modified are listed as follows:

- Components:
    - Web server (new)
    - WebRTC client (new)
    - Signaling Gateway (SGW) (new)
    - MCF (modified)
    - MDF (modified)
- Interfaces:
    - W1 (new)
    - W2 (new)
    - e* (new)
    - Xp (modified)
    - Xd (modified to Xd').

These changes and modifications made are described below. The web server is only needed for providing the WebRTC application sources which are fetched anew every time the end-users' browser accesses the web application. The WebRTC application is executed in a WebRTC capable browser. The application provides signaling functions for the communication with the core network via the inserted Signaling Gateway (SGW). Therefore, to make IMS-based IPTV services accessible to WebRTC clients, Generic IPTV Capabilities described in [2] are supported. The SGW implements the following generic capabilities:

- Service discovery and selection,
- Service control,
- Service interaction and
- Media control.

This gateway function converts session control messages coming from the WebRTC client side into SIP messages for the IMS core network side and vice versa. The SGW generates and forwards SIP messages towards the IMS core network and acts in place of the WebRTC client as a SIP capable signaling endpoint. As appears in the Figure 3 the SGW also converts the session control messages from the WebRTC client into HTTP and Real-Time Streaming Protocol (RTSP).

WebRTC strictly defines the media delivery, which requires a modification of the components of the standardized IMS-based IPTV architecture. These modifications specifically apply to the MCF and the MDF. According to [2], the MCF receives information about sessions from the SCF, finds and chooses the right MDF for media delivery and sends a response back to the user. The choice of the right MDF is based on codec information or geographical location. Information about the used protocol for media delivery is also part of the selection. Afterwards, the MCF transmits session information to the selected MDF. The important parts of information which need to be transmitted are the resource identifier of the media file to be streamed, the DTLS certificate fingerprint, the generated ICE candidates of the WebRTC client and specific connection information for establishing a transport channel.

Hence, the MDF can support the WebRTC's requirements, it shall support several new features. The concept of the MDF is depicted in Figure 4. One of the features is the MDF control engine. This engine dispatches the signaling from the MCF towards the internal MDF functions and vice versa. Another feature includes the audio and video codecs, which are also supported by the WebRTC client. Current WebRTC implementations prefer the VP8 video codec and the Opus audio codec. The codec handling is integrated in the streaming server. Further, WebRTC only allows a secure transport channel. That is why the MDF is required to support SRTP. To establish such a secure channel, the DTLS certificate fingerprint and a successful DTLS key exchange is needed [20]. SRTP keys are obtained through the DTLS key exchange [21]. The streaming server and the security capabilities of the media path are aggregated in the streaming engine. Another additional functionality of the MDF is an ICE agent. This agent supports the ICE methods regarding the SDP Offer/Answer negotiation and procedures for doing connectivity checks, which are similar to the functions documented in [19]. It also has to implement a STUN server functionality to support the STUN keep-alive usage as defined in [16]. This is used by the WebRTC client to preserve the NAT bindings.

With the help of these modifications, usage of a special gateway for media transcoding, which is described in [19], is not necessary.

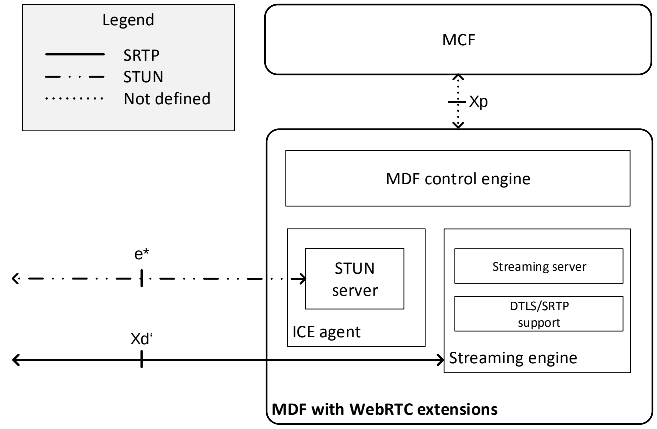

Figure 4: Detailed concept of the MDF

C. Interfaces

The interfaces shown in Figure 3 differ from the standardized architecture. The interfaces for the service interconnection are relocated from the user side to the SGW, which resides in the Telco's network infrastructure. The functionality and the used protocols of the Xa, Xc and Gm interfaces between the SGW and the service or the core functions are still conform to their specification [3]. The Ss, ISC and y2 interfaces remain unaffected despite the consolidation. In addition to these interfaces, some are modified or added and differ from the specification. These are described in more detail below.

The formerly undefined Xp interface between the MCF and the MDF is extended in the range of functions, respectively the extension of the MDF.

The W1 interface is a reference point between the WebRTC client and the web server. It is used to download HTML5, JavaScript, Cascading Style Sheets (CSS) and image files using HTTP. Via this interface, the user receives the latest WebRTC web application.

The W2 interface is located between the WebRTC client and the SGW. The used protocol for this reference point has been deliberately left open because the WebRTC does not define a signaling protocol. Therefore, the developer can choose one out of several state-of-the-art client-server protocols. Protocols like the WebSocket Protocol or the HTTP 2.0 specification could apply [22]. The meaning of protocol messages regarding this interface must cover the sense of the transfered protocol messages from the interfaces Xa, Xc and Gm.

The Xd' interface between the WebRTC client and the MDF is responsible for media delivery using SRTP [23]. The original Xd interface only supports RTP/RTCP or HTTP for media delivery, so the modified interface for WebRTC interconnection is called Xd'. This modification results from the mandatory use of a secure connection in WebRTC [13].

The e* interface is a second reference point between the WebRTC client and the MDF. This interface is added in the proposed concept. It is used for STUN connectivity checks between the both components to preserve the NAT bindings of the client. STUN is a mandatory to implement feature of WebRTC because the most WebRTC clients are behind NAT firewalls.

D. Proof of concept

To verify the functionality and the usability of the proposed concept, a testbed is prepared. With this implemented testbed, the content on demand use case (audio and video) is realized and tested. This includes the following procedures:

- A successful registration of the WebRTC client with the IMS core network is implemented.
- IMS-based IPTV generic capabilities like service discovery and selection, service control and media control are realized.
- The media delivery procedure with the WebRTC specifics like secure RTP transport.

For testing the concept the Google Chrome browser in version 31, which supports WebRTC, is used. The basis of this testbed is formed by an open-source IMS core network implementation originating from Fraunhofer FOKUS institute [24]. An Apache HTTP Web Server provides the web application. The WebRTC client is implemented by using HTML5 and JavaScript. Based on this the Graphical User Interface (GUI) of the client is a responsive web site design using the jQuery mobile framework. This framework makes web sites accessible on all smartphone, tablet and desktop devices [25]. The client's source code, based on JavaScript, utilizes the WebRTC API.

For the W2 interface, located between the WebRTC client and the SGW, a proprietary signaling protocol is defined, which is formatted in JavaScript Object Notation (JSON) and is transmitted through a WebSocket connection.

The SGW is written in C# and designed to handle several WebRTC Client sessions simultaneously. The prototyped SGW provides the main functionalities for the interaction with the Gm and the Xa interface. Based on the sipsorcery project, an enhanced SIP protocol stack supporting IMS specific extensions is implemented [26].

The implementation of the Xc interface for session controlling is not considered yet, because it is necessary only for advanced media streaming control functions, such as 'pause' or 'fast forward'.

Also the IMS-based IPTV components are prototyped, which are results from a related student's bachelor thesis [27]. All prototyped IPTV components, written in Java, are based on the technical specification [2].

The successful implementation of the MDF considers the modifications presented in Figure 4. The control engine parses the session information, passed by the MCF, and operates the ICE Agent and the streaming engine. Open-source frameworks are used for

- The ICE agent with the STUN functionalities (icedjava) [28],
- The DTLS key exchange (BouncyCastle) [29],
- The SRTP implementation (srtplight) [30],
- And the streaming server (FFmpeg) [31].

IV. Conclusion and Future Work

The presented concept of the combination of IMS-based IPTV and WebRTC has a huge sustainable potential. The authors believe that the web based real-time communication is inevitable for future telecommunication. For Telcos it is possible, with these mentioned modifications of the IMS-based IPTV architecture, to deliver the IPTV service to a wider range of end-user's devices. Future developments of new combined services will be realized, using the advantages of both technologies. The possibility and practicability of the concept is verified by an implemented prototype. The main advantages of the proposed architecture are:

- No media gateway function for live transcoding of the content is needed.
- The modified MCF and the MDF can coexist to other original media functions.
- The scalability of the IMS-based IPTV architecture remains unaffected.

The presented use case, audio and video content on demand, has a big potential for multimedia users: an online video streaming platform without using any proprietary software on the users' devices could be realized. Thus, the user can access this service having the same experience everywhere, with any device.

Currently, the implementation of session mobility for content on demand services is in progress. This includes a dialog state awareness service. Thereby users are able to obtain information about active sessions of their own several devices. The next step, the authors are focusing on, is to implement QoS characteristics known from IMS-based core networks to work with web application based WebRTC clients. For future work, combined services like IMS-based IPTV with presence services are conceivable.

Acknowledgment

The Telekom Innovation Laboratories and the Hochschule fuer Telekommunikation Leipzig are actively cooperating since 2011. Both are working together on common topics in the area of IMS-based services participating in ongoing projects relating WebRTC with Telco Assets. This paper also presents some of the results and acquired experience arising from various final students' theses.

The authors would like to thank Mr. Sven Walter to provide IMS-based IPTV typical components like SCF, SDF and MCF for the testbed.

References

[1] C. Jennings, A. Narayanan, D. Burnett, and A. Bergkvist, "WebRTC 1.0: Real-time communication between browsers," W3C, W3C Working Draft, Sep. 2013, http://www.w3.org/TR/2013/WD-webrtc-20130910/ [retrieved: Dec., 2013].

[2] ETSI, "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); IPTV Architecture; IPTV functions supported by the IMS subsystem," European Telecommunications Standards Institute (ETSI), TS 182027 v3.5.1, Mar. 2011, Available: http://www.etsi.org/deliver/etsi_ts/182000_182099/182027/03.05.01_60/ts_182027v030501p.pdf [retrieved: Dec., 2013].

[3] ETSI, "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); IMS-based IPTV stage 3 specification," European Telecommunications Standards Institute (ETSI), TS 183063 v3.5.2, Mar. 2011, Available: http://www.etsi.org/deliver/etsi_ts/183000_183099/183063/03.05.02_60/ts_183063v030502p.pdf [retrieved: Dec., 2013].

[4] 3GPP, "IP Multimedia Subsystem (IMS); Stage 2," 3rd Generation Partnership Project (3GPP), TS 23.228 v8.12.0, Mar. 2010, Available: http://ftp.3gpp.org/specs/html-info/23228.htm [retrieved: Dec., 2013].

[5] 3GPP, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Presence Service; Architecture and functional description," 3rd Generation Partnership Project (3GPP), TS 23.141 v10.0.0, Mar. 2011, Available: http://ftp.3gpp.org/specs/html-info/23141.htm [retrieved: Dec., 2013].

[6] E. Mikoczy, S. Schumann, and P. Podhradsky, "Personalization of internet protocol television (iptv) services in next-generation networks (ngn) architectures," Proceedings of the 8th International Conference on

Advances in Mobile Computing and Multimedia, November 2010, pp. 366 – 369.

[7] 3GPP, "Network architecture (Release 8)," 3rd Generation Partnership Project (3GPP), TS 23.002 V8.7.0, Dec. 2010, Available: http://3gpp.org/ftp/Specs./html-info/23002.htm [retrieved: Jan., 2014].

[8] Google Inc. Webrtc. [Online]. Available: http://www.webrtc.org/ [retrieved: Jan., 2014]

[9] IETF. Rtcweb status pages. [Online]. Available: http://tools.ietf.org/wg/rtcweb/ [retrieved: Jan., 2013]

[10] C. Jennings, T. Hardie, and M. Westerlund, "Real-time communications for the web," Communications Magazine, vol. 51, no. 4, April 2013, pp. 20 – 26.

[11] J. Valin and C. Bran, "WebRTC Audio Codec and Processing Requirements draft-ietf-rtcweb-audio-04," Internet-Draft, Internet Engineering Task Force, Jan. 2014, Available: http://tools.ietf.org/html/draft-ietf-rtcweb-audio-04 [retrieved: Jan., 2014].

[12] IETF. Ietf discussion forum for video in webrtc. [Online]. Available: http://www.ietf.org/mail-archive/web/rtcweb/current/maillist.html [retrieved: Jan., 2014]

[13] C. Perkins, M. Westerlund, and J. Ott, "Web Real-Time Communication (WebRTC): Media Transport and Use of RTP draft-ietf-rtcweb-rtp-usage-11," Internet-Draft, Internet Engineering Task Force, Dec. 2013, Available: http://tools.ietf.org/id/draft-ietf-rtcweb-rtp-usage-11.txt [retrieved: Jan., 2014].

[14] E. Rescorla and N. Modadugu, "Datagram Transport Layer Security Version 1.2," RFC 6347 (Proposed Standard), Internet Engineering Task Force, Jan. 2012, Available: http://www.ietf.org/rfc/rfc6347.txt [retrieved: Jan., 2014].

[15] E. Rescorla, "WebRTC Security Architecture draft-ietf-rtcweb-security-arch-07," Internet-Draft, Internet Engineering Task Force, Jul. 2013, Available: http://tools.ietf.org/id/draft-ietf-rtcweb-security-arch-07.txt [retrieved: Jan., 2014].

[16] J. Rosenberg, R. Mahy, P. Matthews, and D. Wing, "Session Traversal Utilities for NAT (STUN)," RFC 5389 (Proposed Standard), Internet Engineering Task Force, Oct. 2008, Available: http://www.ietf.org/rfc/rfc5389.txt [retrieved: Jan., 2014].

[17] J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols," RFC 5245 (Proposed Standard), Internet Engineering Task Force, Apr. 2010, updated by RFC 6336. Available: http://www.ietf.org/rfc/rfc5245.txt [retrieved: Jan., 2014].

[18] H. Alvestrand, "Overview: Real Time Protocols for Brower-based Applications draft-ietf-rtcweb-overview-08," Internet-Draft, Internet Engineering Task Force, Sep. 2013, Available: http://tools.ietf.org/id/draft-ietf-rtcweb-overview-08.txt [retrieved: Jan., 2014].

[19] 3GPP, "Study on Web Real Time Communication (WebRTC) access to IP Multimedia Subsystem (IMS); Stage 2," 3rd Generation Partnership Project (3GPP), TS 23.701 v12.0.0, Dec. 2013, Available: http://www.3gpp.org/DynaReport/WiVsSpec--580062.htm [retrieved: Jan., 2014].

[20] J. Fischl, H. Tschofenig, and E. Rescorla, "Framework for Establishing a Secure Real-time Transport Protocol (SRTP) Security Context Using Datagram Transport Layer Security (DTLS)," RFC 5763 (Proposed Standard), Internet Engineering Task Force, May 2010, Available: http://www.ietf.org/rfc/rfc5763.txt [retrieved: Jan., 2014].

[21] D. McGrew and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP)," RFC 5764 (Proposed Standard), Internet Engineering Task Force, May 2010, Available: http://www.ietf.org/rfc/rfc5764.txt [retrieved: Jan., 2014].

[22] I. Fette and A. Melnikov, "The WebSocket Protocol," RFC 6455 (Proposed Standard), Internet Engineering Task Force, Dec. 2011, Available: http://www.ietf.org/rfc/rfc6455.txt [retrieved: Jan., 2014].

[23] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)," RFC 3711 (Proposed Standard), Internet Engineering Task Force, Mar. 2004, updated by RFC 5506. Available: http://www.ietf.org/rfc/rfc3711.txt [retrieved: Jan., 2014].

[24] Fraunhofer FOKUS. Fokus open ims playground. [Online]. Available: http://www.fokus.fraunhofer.de/en/fokus/index.html [retrieved: Jan., 2014]

[25] jQuery Mobile: a HTML5-based user interface system. [Online]. Available: http://jquerymobile.com/ [retrieved: Feb., 2014]

[26] T. Bach. sipsorcery-fork. [Online]. Available: https://github.com/hftl-ims-research/sipsorcery-fork [retrieved: Apr., 2014]

[27] S. Walter, "Design and Implementation of an IMS-based IPTV Content on Demand Service," Bachelorthesis, Hochschule fuer Telekommunikation Leipzig, HfTL, Aug. 2013.

[28] inspired social. Open source ICE implementation. [Online]. Available: http://code.google.com/p/inspired-social/source/browse/trunk/StunServer/net/mc_cubed/icedjava?spec=svn20&r=20#icedjava [retrieved: Apr., 2014]

[29] Legion of the Bouncy Castle Inc. Bouncy Caslte Crypto API. [Online]. Available: http://www.bouncycastle.org [retrieved: Apr., 2014]

[30] steely gint. set of classes implementing a simple (S)RTP stack. [Online]. Available: https://github.com/steely-glint/srtplight [retrieved: Apr., 2014]

[31] FFmpeg: a open source cross-platform solution to record, convert and stream audio and video. [Online]. Available: http://www.ffmpeg.org/index.html [retrieved: Feb., 2014]