Integrating WebRTC and X3DOM: Bridging the Gap between Communications and Graphics

Haroula Andrioti, Andreas Stamoulias, Kostas Kapetanakis, Spyros Panagiotakis, Athanasios G. Malamos
Department of Informatics Engineering
Technological Educational Institute of Crete
Heraklion, Greece, GR 71004

Abstract

In recent years, with the emergence of HTML5, the Web has managed to evolve from an Internet application for information announcement and exchange to a toolkit provided for pervasive, ubiquitous and collaborative services. In this context, the Web can ultimately provide Real-Time Communication (RTC) services between browsers via the Internet. WebRTC is the standardized project that provides browsers and mobile applications with RTC capabilities via simple JavaScript APIs. This opens new horizons in web-based applications such as capabilities for online gaming, support for complicated 3D graphics in real-time, video-conferencing, exchanging of text messages, immersive technology, etc. In this paper, we introduce the integration of WebRTC capabilities within virtual 3D worlds and present several implementations that bridge WebRTC and X3DOM technologies. In the applications we have developed, virtual 3D collaborative environments are provided for the cooperation of web peers at real time, while they are able to manipulate a 3D scene without the use of plugins. In this context, an online educational game and an immersive conference tool supporting all types of real-time communication such as video chat, text messaging and the use of social media, both using WebRTC over X3DOM, are presented and thoroughly discussed.

Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and RealismC.2.1 [Computer Systems Organization]: Network Architecture and Design

Keywords: WebRTC, X3DOM, HTML5, JavaScript, Virtual Worlds, 3D Collaborative Environments

1 Introduction

The Web over the years has managed to find solutions in several issues related to the area of communications. The latter includes the bidirectional delivery of video, audio or any data over the web in real-time using peer-to-peer (P2P) connections without the need of plugins. The result was the development of Real-Time Communication (RTC). In particular, the IETF (Internet Engineering Task Force) and the W3C (World Wide Web Consortium) went into partnership leading to the development of the WebRTC (Web Real-Time Communication) technology. The goal of WebRTC (WebRTC) is to provide real-time communication among browsers in a P2P connection with no use of plugins. In this partnership, IETF is responsible for the specification of the communication protocols used in WebRTC and W3C for the specification of JavaScript APIs that enable web access to local media devices (i.e. web cameras and microphones) and the underlying communication infrastructure. Loreto and Romano (Loreto & Romano, 2012) provided a full presentation of WebRTC technology and specifically its architecture, APIs, congestion control and security aspects. In addition, Becke et al. (Becke, et al., 2013) referred to RTCWeb for browser-to-browser communication focusing especially on data transfer and how the use of Stream Control Transmission Protocol (SCTP) can improve data transfer in the DataChannel API. Jennings et al. (Jennings, Hardie, & Westerlund, 2013) presented, also, an overview of WebRTC technology for audio, video and data transfer between browsers and other applications.

Several applications have been developed so far using WebRTC for a variety of application domains. For instance, WebRTC is used for implementing video calls, exchanging text messages, video-conferencing, online gaming, etc. In the same context, Mozilla Firefox recently introduced the 'Firefox Hello' application (Weiner, 2014) based on WebRTC technology. The latter provides free video and audio calls via browser without any need for software or plugins download. This application is compatible with Firefox, Chrome and Opera browsers. Other similar applications include the OpenTok (TokBox), which provides face-to-face video chats and is supported for websites and mobile applications, as well as the various Web Call-Me and Click-to-Call buttons provided by various developers (Call.Me; Call-Me/Click-to-Call).

On the other hand, with respect to the support for advanced 3D graphics over the web, one of the latest technologies that supports the above features is X3DOM. X3DOM is an open-source framework written in JavaScript that combines WebGL and X3D technologies and provides the capability to handle 3D content over the web without the need to use plugins. Behr et al. (Behr, Eschler, Jung, & Zöllner, 2009) introduced X3DOM architecture and its evolvement. Due to its great potentials in introducing 3D content over the web, X3DOM is likely to be the dominant technology for immersive applications. Immersive technology can be defined as the technology that handles the approach of two worlds: the physical and the digital one, thus, developing the percept of immersion. (Immersive Technology) According to Abassi et al. (Abbasi & Baroudi, 2012), an immersive digital environment is mainly based on many variables such as 3D graphics, surround sound, user interactivity, functional, simple and fun. Various applications can use immersive technology such as from the fields of education, healthcare, entertainment, video gaming, collaboration activities, etc.

e-mail: haroulaa@gmail.com, andr.stamoulias@gmail.com, kapekost@gmail.com, spanag@teicrete.gr, amalamos@ie.teicrete.gr
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Request permissions from Permissions@acm.org.
Web3D '15, June 18 - 21, 2015, HERAKLION, Greece
© 2015 ACM. ISBN 978-1-4503-3647-5/15/06…$15.00
DOI: http://dx.doi.org/10.1145/2775292.2775316

In this respect, our goal is to combine WebRTC technology with X3DOM developing interactive virtual worlds based on real-time communication where features such as real-time interaction, immersive experience, virtual worlds, and interoperability can be supported.

From this perspective, in this paper, we introduce to the integration of WebRTC over X3DOM and present some applications that have been developed in Multimedia Content Laboratory (MCLab) to this end. These applications integrate the fields of Communications and Graphics. In more specific, we discuss, here, two such applications: the first provides a 3D collaborative environment for educational gaming that supports video calls, text messaging and the capability to select, insert and manipulate 3D objects in a X3D scene using the Data Channel of WebRTC. The second offers an immersive video conference experience based on a virtual world that provides real-time communication and can support not only video call, and text messaging but also real-time connection with social media such as Facebook, twitter, YouTube, Spotify, and email as well. In order to accomplish these applications, we used the WebRTC and X3DOM frameworks, HTML5, and JavaScript, enhanced with node.js and socket.io tools for our developments. The motivations and the technological background behind this work has been previously reported in (Panagiotakis, et al., Scheduled for 2015). Furthermore, the second application was demonstrated in the Web Real-Time Communication Conference in Munich 2014 (Web Real-Time Communication 2014 Conference, 2014).

This paper is structured as follows: Part 2 presents some related work which is implemented with WebRTC and X3DOM technologies. Part 3 introduces to the WebRTC Architecture. Part 4 describes the WebRTC Constituents. Part 5 presents all the technologies and tools that are being used. In Part 6 we introduce the applications that have been developed. Ultimately, in Part 7 we present our conclusions and future plans.

2 Related Work

Online gaming is evolving very fast bringing new proposals in the field. In recent years, there have been interesting and promising for the future developments in the field of web 3D gaming based on real-time communication. In particular, we observe that WebRTC technology has been used in combination with HTML5, JavaScript, and WebGL. First example of such development is the Cube Slam (Cube Slam; Uberti, 2013) web game; it was created by Google Chrome Experiment and developed based on WebRTC technology. Cube Slam takes advantage of all three APIs of WebRTC. The technologies that are used in order to develop Cube Slam are WebGL, CSS 3D, and Web Audio API for the sound. BananaBread is another web 3D game developed in 2012. (azakai, 2012) It is created to run on a web browser. BananaBread is a C++ 3D game engine and is combined with HTML5, JavaScript and WebGL technologies. Finally, the last demo was developed to support WebRTC for multiple users.

PeerCDN is a P2PContent Delivery Network which offers a further perspective of using WebRTC technology such as in applications that support file sharing. PeerCDN was developed in 2013 by Hiesey et al. (Hiesey et al., 2013) and has been obtained by Yahoo. PeerCDN uses RTCDataChannel API in order to establish the connection and transfer the data between the peers. Its main features are credibility, security, fast file transfer which leads to the improvement of response time. Furthermore, PeerCDN has managed to reduce the bandwidth and the server costs. Similar to this concept, WebTorrent (Aboukhadijeh) is a streaming torrent application that transfers files via the web using WebRTC technology.

Immersive technology has also shown rapid development especially in the field of entertainment and gaming. For instance, Oculus VR (Oculus VR) is an immersive virtual reality technology and is being used for immersive gaming. Oculus Rift offers to people the experience of interactive content. Another immersive virtual reality environment is CAVE (CAVE) which stands for Cave Automatic Virtual Environment. Cave is based on projectors that are directed to a room which has the shape of a cube. Moreover, YouTube presented recently the first interactive 360-degree video clip based on immersive technology. The above video was developed by the companies Intel and Kolor. This immersive experience brings a new era in the fields of media such as video clips, films, etc. (O'Brien, 2015; De Buck, 2015). Kapetanakis et al. (Kapetanakis, et al., 2014) introduced another aspect of X3DOM where adaptive video streaming is set on top of Web3D. They present a bridging technology between X3DOM and MPEG-DASH. Another perspective based on the X3DOM environment is introduced by Stamoulias et al. (Stamoulias, et al., 2014) and is referred to the implementation of the rigid body physics component and the creation of dynamic 3D interactive worlds. Additionally, in recent years, interactive 3D collaborative environments are developed such as to support for example online edutainment games (Kapetanakis, et al., 2013) and (Kapetanakis, Panagiotakis, & Malamos, 2013).

3 WebRTC Architecture

RTCWEB specification (Alvestrand, 2014) seals the partnership of IETF and W3C for RTCWEB/WEBRTC technology. Two model architectures are presented in Figures 1 and 2. As shown in Figure 1, there are two core interfaces which need specification. The first interface adverts to the ***Protocols*** that browsers use so apply a direct connection between them without any intervening servers. It is denoted by the lower "On-the-wire Protocols" in Figure 1. The second interface adverts to the ***APIs*** that are provided for a HTML/CSS/JavaScript application so as to benefit from browser's functionality. They are denoted as "RTC APIs" in Figure 1.

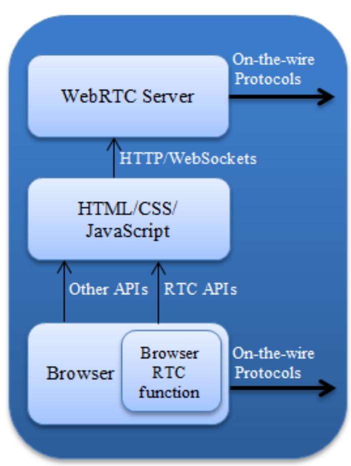

Figure 1: *Browser Model*

Browser RTC Trapezoid is inspired by the Session Initiation Protocol (SIP)-based communications. As shown in Figure 2, each browser is associated with a WebRTC server which is accountable for the session initialization and the negotiation with its opposite server. Each browser has the capability to utilize the WebRTC APIs once a WebRTC-aware JavaScript web application is downloaded through a web server. When a WebRTC session is established, both a media and a data path are instantly opened connecting the two browsers. In regard of the exchange of signaling messages either among browsers and servers and/or among servers, signaling is the one that is responsible for the management of sessions. Several protocols such as HTTP and WebSockets can be utilized among browsers and WebRTC servers in order to support their communication.

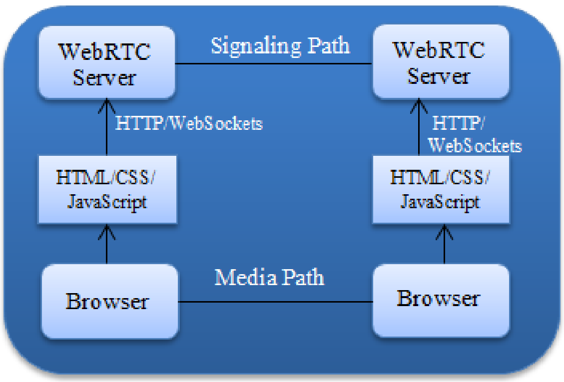

Figure 2: *Browser RTC Trapezoid*

4. WebRTC Constituents

In this part, we introduce the WEBRTC (Bergkvist et al., 2015) constituents that include: WebRTC main APIs, Signaling, and WebRTC Networks. Being more specific:

4.1 WebRTC main APIs

WebRTC consists of three main APIs which are *MediaStream* API, *RTCPeerConnection* API and *RTCDataChannel* API. It should be mentioned that all the above APIs are supported by Google Chrome, Mozilla Firefox, and Opera web browsers.

MediaStream API is used to ensure audio and video. In order to acquire the audio and video stream, *navigator.getUserMedia()* method is used. *RTCPeerConnection API* is used to provide direct communication among peers. It establishes the session negotiation between peers through WebRTC servers. The main features are efficiency, stability, and security. As far as security and user privacy are concerned, the protection of users and their devices is absolute mandatory. For that reason, Secure Real Time Protocol (SRTP) and Datagram Transport Layer Security (DTLS) protocols are being used (RFC4347, 2006; Dutton, 2012).
RTCDataChannel API enables bidirectional peer-to-peer communication between peers exchanging arbitrary data without the need of an intermediate server to transfer data. Furthermore, it enables data exchange over Stream Control Transmission Protocol (SCTP) packets (Jesup, Loreto, & Tuexen, 2015). IETF has standardized SCTP as RFC 4960 (Stewart, 2007) and is defined as a transport-layer protocol. Since SCTP is not globally supported as a transport protocol, the appropriate wrappers for its use over TCP or UDP are defined. In this context, RTCDataChannel API offers the choice whether to establish a reliable or unreliable connection. Reliable connection supports the use of SCTP over TCP (Transmission Control Protocol) protocol. Unreliable connection supports the use of SCTP over UDP (User Datagram Protocol) protocol (Ristic, 2014).

4.2 Signaling

Signaling is not specified by WebRTC. Signaling is referred as a process to coordinate communication among peers and to send control messages apart from using RTCPeerConnection API in WebRTC (Dutton, 2012). Signaling is used during session establishment or update to exchange an offer and an answer between browsers. SDP (Session Description Protocol) is the legacy protocol that can be used to this end, although WebRTC specifies also JSEP (JavaScript Session Establishment Protocol).

4.3 WebRTC Networks

WebRTC supports *STUN*, *TURN* or *ICE* NAT-traversal networks. It is a necessity to use NAT-traversal otherwise WebRTC cannot be used properly at the session negotiation phase where a public IP address needs to be announced to the opposite peer.

STUN (Session Traversal Utilities for NAT) is a client-server protocol. In Figure 3, there is a depiction of STUN's network. STUN reveals for every user its public IP address and port in order to get access to the network. This process assists a WebRTC peer to obtain its public IP address and port, and to carry the information through WebRTC signaling to another peer. Thus, STUN provides access to WebRTC media and data flows (for example real-time voice, video, and messaging) among peers.

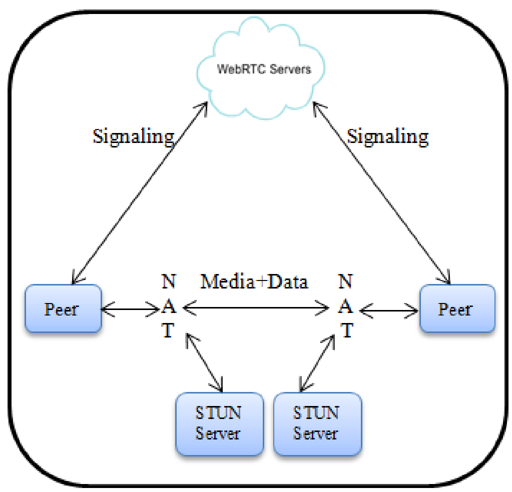

Figure 3: *STUN Network Architecture*

TURN (Traversal Using Relays around NAT) is a relay server. TURN server is used as a final solution if it fails previously to establish the connection using STUN server. It ensures that audio, video and data streams will be transmitted between peers. In Figure 4, we present a representation of TURN relay server architecture. *ICE* (Interactive Connectivity Establishment) is an umbrella framework for STUN and TURN responsible for the connection between peers. ICE tries to find the best solution selecting between STUN and TURN. Firstly, ICE tries to establish connection between peers using STUN server. If that fails, then ICE uses a TURN relay server (Dutton, 2013).

5 Background Technologies & Tools

HTML5 (Hickson, et al., 2015) is a markup language which is used to display and structure the content of a webpage on the Internet. HTML5 is the latest version of HTML and is enhanced with several features. Some of HTML5 features are the following: For instance, it supports WebSockets, includes new elements such as video, audio and canvas without using any plugin. Moreover, Drag-and-Drop feature is also supported in HTML5. HTML5 is supported by the latest versions of most browsers such as Google Chrome, Mozilla Firefox, Opera, and Apple Safari.

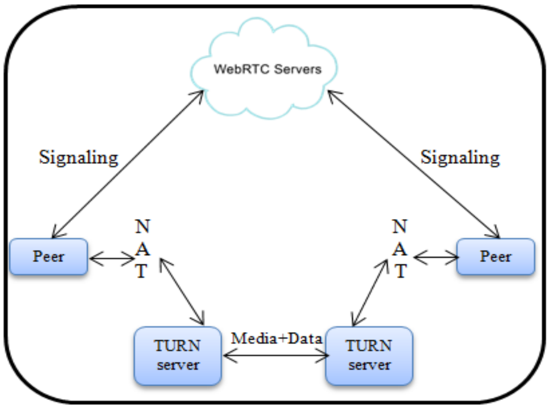

Figure 4: *TURN Network Architecture*

JavaScript (JavaScript) is a dynamic script language which is used to develop HTML web pages. JavaScript is specified by ECMAScript language specification.

X3D (X3D) is a free ISO standard, succession of VRML (Virtual Reality Modeling Language),where XML file format is used in order to work on 3D content. It is a run-time architecture that provides representation and interaction of 3D scenes. WebGL which stands for Web Graphics Library (WebGL) is a cross-platform and royalty-free web standard based on OpenGL ES 2.0 designed by Khronos Group for 3D graphics. X3DOM (X3DOM) is the result of WebGL and X3D technologies combined together based on JavaScript API specification. X3DOM is an open-source framework that provides the opportunity to work with 3D content without the support of any plugin. X3DOM combines the X3D (Extensible 3D Graphics) ISO standard and DOM (Document Object Model). X3DOM main characteristic is that it provides real-time X3D scene where the user has the ability to manage the 3D content by manipulating (add, remove or change) DOM elements. Behr et al. (Behr, et al., 2010) introduced a scalable architecture that further evolves X3DOM. X3D, WebGL and X3DOM are supported by Google Chrome, Mozilla Firefox and Opera browsers.

Node.js (Node.js) is a free and open-source platform which is built on Chrome's JavaScript runtime. Node.js characteristics include efficiency, flexibility, easy to install and use, lightweight and offers scalable network applications.Socket.io (Socket.io) is an open-source real-time engine. It supports real-time bidirectional communication and is characterized as reliable and fast. Applications that can be implemented using socket.io is real-time analytics, instant messaging and chat, binary streaming and document collaboration. Socket.io provides convenience for developers to write code for client and server side. Node.js is the most appropriate tool to use in order to write the server part of socket.io (Rauch, 2014).

6 Implementations

Our goal is to present a combination of real-time communication, using WebRTC, with 3D graphics, using X3DOM technology. In order to evaluate the synergistic integration of WebRTC and X3DOM technologies, we approached this objective by implementing two different types of applications. The first one, concerns an update and modernization of an old edutainment game called EViE-m (Kapetanakis, et al., 2013), which was firstly developed based on Java and X3D technologies combined with Xj3D browser in order to exhibit the 3D world. So what we did was to obtain the X3D world of EViE-m and extend this application in developing an educational web game supporting browser-to-browser communication, totally plugins-free, between two users. The second application has a different approach. It is based on immersive technology which has a rapid evolvement in the field of entertainment. As a result, we present an immersive environment based on real-time communication that supports the interaction between multiple peers in various actions including video/audio call, text messaging, file sharing, and connecting into social networks such as Facebook, Twitter, YouTube, Spotify, and Gmail accounts. The above applications have been developed in Multimedia Content Laboratory (MCLab).

6.1 3D Collaborative Educational Game

The goal of this application is to develop an educational web 3D game using WebRTC and X3DOM technologies. In this application, we manage to bridge two of the state-of-the art technologies from the fields of Communications and Graphics: WebRTC and X3DOM. Our scope was to implement a web-based application which provides a 3D collaborative environment where we could transmit not only generic data but also X3D objects using WebRTC technology.

In Figure 5, a diagram of our application is introduced. This diagram presents the structure of how our application was developed. As it is shown, each peer is connected with a node.js server and establishes a RTCPeerConnection via the so-named API. It should be mentioned that we also need to activate signaling in order our peers to establish successfully the connection between them. Hence, we use socket.io for the signaling part. Once the connection is ready, WebRTC component is activated. RTCDataChannel and MediaStream APIs are included in the WebRTC component. MediaStream is responsible for the video call. In order to develop the video call, HTML5 elements were used. On the other hand, RTCDataChannel API is responsible for the text chat implementation where text messages are exchanged between peers. Furthermore, each peer has the opportunity to manipulate a X3D object in the X3D/X3DOM world. Each peer is connected with a X3D models repository where 3D objects (houses in this case) are gathered. Once a house is selected, it appears in the X3D/X3DOM world where then the peer can change its position. Both peers can see each other's actions and as a result there are continuous updates of the X3D/X3DOM world, enabling interactivity among peers.

It is worth noticing the integration of RTCDataChannel API of WebRTC with X3DOM technology in selecting and transferring a X3D object into the X3D/X3DOM world. Once we have selected a X3D house to import into the X3D/X3DOM world, a new channel is opened. This is achieved by defining the initial translation and rotation of each house. Using the "*inline*" method from X3DOM, we have the opportunity to directly insert X3D objects into the X3D/X3DOM world. We utilize two variables in order to obtain the last position of each house once it has been changed inside the X3D/X3DOM world: the *Id* of each house and its *translation* (defining the position of the house). These variables maintain the above information until a user relocates the house in the X3D/X3DOM world. Once the house changes position, then new information is saved and the old is discarded. Ultimately, each peer has the opportunity to change the view angle of the virtual world.

In Figure 6, we present our application as shown when Chrome browser is opened. For our implementation purposes, we used the virtual world of an old edutainment game platform, called EViE-m (Kapetanakis, et al., 2013), so as to be used as a demo web 3D collaborative environment. This application provides each user with the capability to start a video call, exchange text messages and select and manipulate 3D objects which in this case are 3D houses. We used node.js as the web server and socket.io for the Signaling part. With respect to text messaging, when both users establish the connection, which is done by clicking on Connection button, a unique user ID is provided to each user that allows them to exchange messages. Ultimately, from the menu which is located at the bottom of the webpage, both users select and insert in the X3D/X3DOM world 3D houses and change their position. Both users can see each other's actions and have the right to intervene and change the result of the action. For instance, if one user changes the position of one's house the other user can intervene and change the position of the same house again.

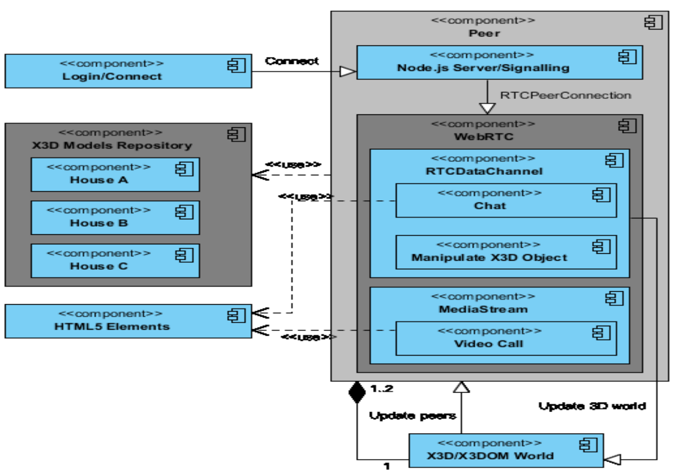

Figure 5: *Diagram of the demo 3D collaborative environment for online game application*

6.2 Immersive – Experience Application

The objective of this application concerns a 3D collaborative environment that supports real-time communication network between peers using a plugin free, peer-to-peer connection. This application was developed so as to run directly to web browsers. Additionally, social networks connection such as Facebook, Twitter, YouTube, Spotify and Gmail is supported. All the above components are composing an immersive virtual world whose management is based on WebRTC technology. Additional third party services can be supported in this application. It should be noticed that node.js server is used in our development, specifically for the signaling part, alongside socket.io API.

The diagram of the Immersive-Experience application is presented in Figure 7. As illustrated in the aforementioned figure, each peer must provide its own username and password in order to get access to the virtual world. Once the peer has access inside the virtual world, then each X3D cube (peer) has the opportunity to exploit all the tools that the virtual world provides. In order to have a successful connection establishment, node.js server is used as our web server. Additionally, socket.io is utilized in order to implement the signaling part. The web server is responsible for the creation of a new X3D cube (peer) and via WebRTC APIs; constantly updates are provided for the status of each peer. All X3D cubes are built in HTML5 canvas element and each X3D cube can be manipulated in the X3D/X3DOM world. This means that each cube can be rotated, translated or even situated on top of another cube inside the virtual world. Furthermore, each X3D cube has continuous updates in the X3D/X3DOM world. RTCPeerConnection API is responsible for establishing the connection among peers.

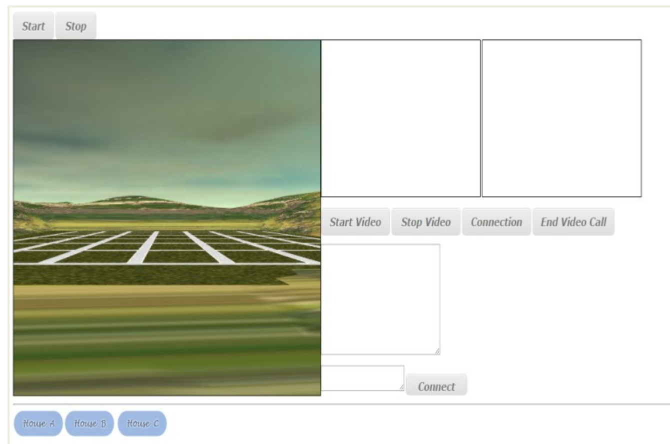

Figure 6: *Virtual 3D Collaborative Environment Online Game for two users*

As we referred in part 4.1, WebRTC consists of 3 APIs: MediaStream, RTCPeerConnection and RTCDataChannel. All of them are used in order to define various actions in the application. For instance, each peer has the opportunity to either create or join a room. As a result, each peer is provided to choose whether to have a public or a private chat with another peer. Moreover, data can be sent and received. As a result, not only text messages can be exchanged but file sharing is supported as well. In addition, video/sound call is supported among multiple peers. Ultimately, each X3D cube (peer) can connect among various social networks such as Facebook, YouTube, Twitter, Spotify, and Gmail. Furthermore each peer can interact with the other peers' social accounts from the aforementioned social networks by clicking on the corresponding cube side, but can also embed in the HTML layer a single video and sound or whole playlists from YouTube and Spotify networks utilizing iframes. All peers that had enabled this feature can then exchange and synchronize YouTube and Spotify player information with other connected peers via the RTCDataChannel and watch or listen a media clip together. The main feature is the interaction and the continuous update which is provided for each peer separately.

In Figure 8, we present our platform where the surface is full with X3D cubes which change levels at all times. On the surface the current time is presented. Each peer has the opportunity to change the position of the view of its own virtual world. We observe four different X3D cubes where each one corresponds to a different connected peer. The four sides around the Y-axis of the cube represent another user interaction component.

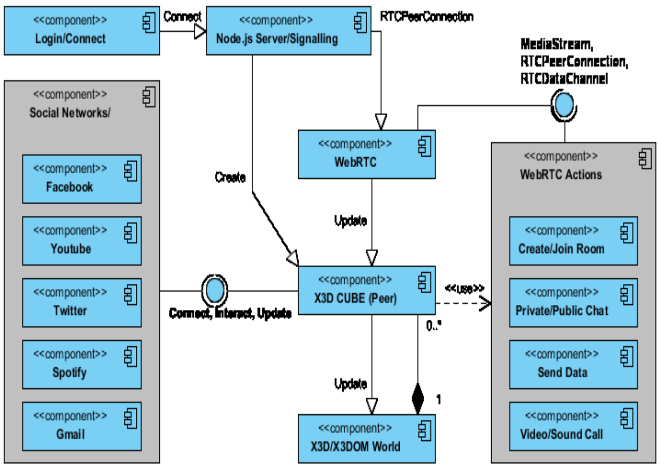

Figure 7: *Diagram of the Immersive – Experience application*

As shown in the image below (Figure 8), the first from the left cube is rotated in order to show the video panel used to chat with another user. The second and third cubes are rotated and show the log in to Facebook and Twitter accounts panels respectively. Finally the X3D cube from the right corresponds to sign in Gmail account side.

Figure 8: *Immersive – Experience application*

In Figure 9, we introduce the Immersive – Experience application as presented online. On the top of the page we observe that each peer has its own user ID. Each peer can join a room, start a VR-Chat, use the web camera and microphone in order to have a video call. In addition, text messages can be exchanged among peers as well as transmitting files. Social networks are also available in the application as shown in Figure 9. Ultimately, each peer can change the view angle of its virtual world and has the opportunity to zoom in and out or bring the virtual world to its default position.

7 Conclusion and Future Work

In this paper, we presented WebRTC technology and it's potential. We introduced two applications that where developed in Multimedia Content Laboratory (MCLab) and showed new perspectives of WebRTC combined with X3DOM technology. In our developments, we enabled 3D collaborative environments in which the first one is created for online gaming capabilities and the second one to support video chat with multiple users, text messaging, and being able to log in social media accounts such as email, Facebook, twitter, etc. A new era of web-based applications is evolving allowing us to develop applications using state-of-the art technologies. We strongly believe in the synergy between WebRTC and X3DOM because the synergistic action of these both technologies can widen the range of developing promising and potential web applications and services.

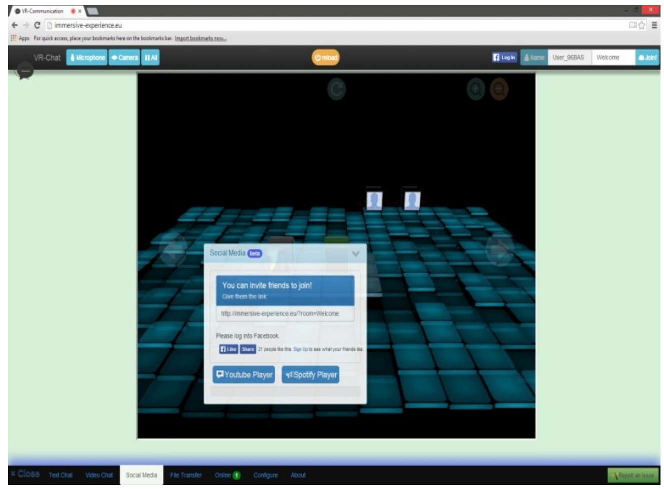

Figure 9: *Immersive – Experience application as shown online*

Lately more and more WebRTC applications appear online, as this technology enabled free and easy to use peer to peer connectivity. Other technological fields such as online gaming, e-learning and e-health can benefit from the peer-to-peer and server-free nature of WebRTC. In our implementations we demonstrated the WebRTC capabilities in the fields of video conferencing, collaborative environments and online gaming, but could become the test-bed for other applications such as in the field of healthcare (e-health) and education (e-learning). There are various features that could be added in our work such as to support multiple users and cross-platform capabilities. Ultimately, it would be interesting to see our work to be supported using cloud-based technology.

Acknowledgements

Work in this paper has been funded by the European Union and the Hellenic Ministry of Education and Religious Affairs under the "Archimedes III" research framework, T.E.I. of Crete, Project Title: "DECO - DECoration Ontology. Ontology and semantic search applications for the support in interior architecture and decoration designs", Project Co de32, Acronym: DECO.

References

ABBASI, A., & BAROUDI, U. (2012). Immersive Environment: An Emerging Future of Telecommunications. *MultiMedia, IEEE , 19*(1), 80-86.

ABOUKHADIJEH, F.WebTorrent. [accessed March 2015] https://github.com/feross/webtorrent

ALVESTRAND, H.2014. Overview: Real Time Protocols for Browser-based Applications, [accessed September 2014] https://tools.ietf.org/html/draft-ietf-rtcweb-overview-10

AZAKAI.2012.BananaBread - MDN. [accessed February 2015] https://developer.mozilla.org/en-US/demos/detail/bananabread

BECKE, M., RATHGEB, E. P., WERNER, S., RUNGELER, I., TUXEN, M., & STEWART, R. (2013, April). Data Channel Considerations for RTCWeb. *Communications Magazine, IEEE, 51*(4), 34 - 41.

BEHR, J., ESCHLER, P., JUNG, Y., & ZÖLLNER, M. (2009). X3DOM: a DOM-based HTML5/X3D integration model. *Web3D '09 Proceedings of the 14th International Conference on 3D Web Technology* (pp. 127-135). ACM, New York, NY.

BEHR, J., JUNG, Y., KEIL, J., DREVENSEK, T., ZOELLNER, M., ESCHLER, P., & FELLNER, D. (2010). A scalable architecture for the HTML5/X3D integration model X3DOM. *Web3D '10 Proceedings of the 15th International Conference on Web 3D Technology* (pp. 185-194). ACM, New York, NY.

BERGKVIST, A., BURNETT, D. C., JENNINGS, C., & NARAYANAN, A. (Eds.). 2015. WebRTC 1.0: Real-time Communication Between Browsers, W3C Editor's Draft: [accessed March 2015]http://w3c.github.io/webrtc-pc/

CALL.ME. Call.Me home page [accessed March 2015] https://www.call.me/

CALL-ME/CLICK-TO-CALL.Call-Me/Click-to-Call home page [accessed March 2015] http://callme-click-to-call.com/World-Wide Connect/Main/Home

CAVE. [accessed March 2015] http://en.wikipedia.org/wiki/Cave_automatic_virtual_environment

CUBE SLAM.Cube Slam home page [accessed March 2015] https://www.cubeslam.com/tech

DE BUCK, K.2015. Intel iQ [accessed March 2015] http://iq.intel.co.uk/a-360-4k-music-video-clip-in-five-steps

DUTTON, S.2012. Getting Started with WebRTC - HTML5 Rocks. [accessed March 2015] http://www.html5rocks.com/en/tutorials/webrtc/basics/

DUTTON,S. 2013 WebRTC in the real world: STUN, TURN and Signaling - HTML5 Rocks[accessed March 2015] http://www.html5rocks.com/en/tutorials/webrtc/infrastructure

HICKSON, I., BERJON, R., FAULKNER, S., LEITHEAD, T., NAVARA, E. D., O'CONNOR, E., . . . MARQUIS, M. (Eds.). 2015. HTML 5.1 Nightly - W3C Editor's Draft 23 March 2015 [accessed March 2015] http://www.w3.org/html/wg/drafts/html/master/#toc-semantics

HIESEY, J., ABOUKHADIJEH, F., & RAJA, A.2013.PeerCDN[accessed March 2015] http://peercdn.com/

IMMERSIVE TECHNOLOGY. [accessed March 2015] http://en.wikipedia.org/wiki/Immersive_technology

JAVASCRIPT. [accessed March 2015] http://en.wikipedia.org/wiki/JavaScript

JENNINGS, C., HARDIE, T., & WESTERLUND, M. (2013, April). Real-time Communications for the Web. *Communications Magazine, IEEE, 51*(4), 20-26.

JESUP, R., LORETO, S., & TUEXEN, M. 2015.WebRTC Data Channels - Internet-Draft. https://tools.ietf.org/html/draft-ietf-rtcweb-data-channel-13

KAPETANAKIS, K., ANDRIOTI, H., VONORTA, H., ZOTOS, M., TSIGKOS, N., & PACHOULAKIS, I. (2013). Collaboration framework in the EViE-m platform. *EAEEIE Annual Conference (EAEEIE), 2013 Proceedings of the 24th* (pp. 178 - 183). IEEE.

KAPETANAKIS, K., PANAGIOTAKIS, S., & MALAMOS, A. G. (2013). HTML5 and WebSockets;challenges in network 3D collaboration. *PCI '13 Proceedings of the 17th Panhellenic Conference on Informatics* (pp. 33-38). ACM.

KAPETANAKIS, K., PANAGIOTAKIS, S., MALAMOS, A. G., & ZAMPOGLOU, M. (2014). Adaptive video streaming on top of Web3D: A bridging technology between X3DOM and MPEG-DASH. *Telecommunications and Multimedia (TEMU), 2014 International Conference on* (pp. 226 - 231). IEEE.

LORETO, S., & ROMANO, S. P. (2012). Real-time Communications in the Web: Issues, Achievements, and Ongoing Standardization Efforts. *Internet Computing, IEEE, 16*(5), 68-73.

MCLAB. Multimedia Content Laboratory [accessed March 2015] http://www.medialab.teicrete.gr/

NODE.JS. [accessed March 2015] http://www.nodejs.org/

O'BRIEN, C.VB VentureBeat - The immersive future. [accessed March 2015] http://venturebeat.com/2015/03/27/the-immersive-future-intel and-kolor-create-youtubes-first-interactive-360-degree-video/

OCULUS VR. [accessed March 2015] https://www.oculus.com/company/

PANAGIOTAKIS, S., VAKINTIS, J., ANDRIOTI, H., STAMOULIAS, A., KAPETANAKIS, K., & MALAMOS, A. (Scheduled for 2015). Towards ubiquitous and adaptive web-based multimedia communications via the cloud. In G. Mastorakis, C. Mavromoustakis, & E. Pallis (Eds.), *Resource Management of Mobile Cloud Computing Networks and Environments*. IGI Global.

RAUCH, G.SOCKET.IO - Introducing Socket.IO 1.0[accessed March 2015] http://socket.io/blog/introducing-socket-io-1-0/#

RFC4347. 2006.Datagram Transport Layer Security IETF.[accessed September 2014] http://tools.ietf.org/html/rfc4347

RISTIC, D. 2014.WebRTC data channels for high performance data exchange. [accessed September 2014] http://www.html5rocks.com/en/tutorials/webrtc/datachannel/

SOCKET.IO.Socket.io home page [accessed March 2015] http://socket.io/

STAMOULIAS, A., MALAMOS, A. G., ZAMPOGLOU, M., & BRUTZMAN, D. (2014). Enhancing X3DOM Declarative 3D with Rigid Body Physics Support. *Web3D '14 Proceedings of the Nineteenth International ACM Conference on 3D Web Technologies* (pp. 99-107). ACM.

STEWART, R. (Ed.). 2007.Stream Control Transmission Protocol [accessed February 2015]http://tools.ietf.org/html/rfc4960

TOKBOX.TokBox home page [accessed March 2015] http://tokbox.com/

UBERTI, J. 2013.Chromium Blog: Play Cube Slam, a real-time WebRTC video game. [accessed February 2015] http://blog.chromium.org/2013/06/play-cube-slam-real-time-webrtc-video.html

WEB REAL-TIME COMMUNICATION 2014 CONFERENCE. [accessed March 2015] http://webrtcevent.com/

WEBGL. Khronos Group - WebGL [accessed February 2015] https://www.khronos.org/webgl/

WEBRTC.WebRTC home page [accessed March 2015] http://www.webrtc.org/

WEINER, C.Mozilla Firefox - Future Releases[accessed March 2015] https://blog.mozilla.org/futurereleases/2014/10/16/test-the-new-firefox-hello-webrtc-feature-in-firefox-beta/

X3D. Web 3D Consortium - Open Standards for Real-Time 3D Communication [accessed February 2015] http://www.web3d.org/x3d/what-x3d

X3DOM.X3DOM home hage [accessed March 2015] http://www.x3dom.org/