

Video Relay Service for Deaf people using WebRTC

Andre J. Henney:
University of the Western Cape,
Robert Sobukwe Road, Cape Town,
7535, South Africa
Tel: +27 21 959 3010
Email: ahenney@uwc.ac.za

William D. Tucker,
University of the Western Cape,
Robert Sobukwe Road, Cape
Town, 7535, South Africa
Tel: +27 21 959 3010
Email: btucker@uwc.ac.za

Abstract— This paper reports on an experimental open source video relay service prototype that helps Deaf people communicate with hearing people by accessing a third party sign language interpreter on a mobile device. Deaf people are disadvantaged in many ways when communicating with the hearing world in real world scenarios, such as hospital visits and in cases of emergency. When possible, Deaf people can enlist the assistance of a family member, community worker or sign language interpreter to assist with such scenarios, however this assistance is pre-arranged and Deaf people would prefer on-the-fly assistance. Our application will assist Deaf people to contact any available sign language interpreter to facilitate communication between the Deaf person and a hearing person using a split screen model, effectively creating a three-way conversation between the Deaf person, the hearing person and the sign language interpreter. The prototype was developed using the WebRTC platform, with JavaScript for browser operability and hardware platform independence. Our hope is that the research can be used to persuade mobile network operators of the need for free or heavily discounted data connection to relay services for Deaf mobile customers.

I. INTRODUCTION

DEAF people throughout the world, including those in South Africa, experience difficulty in basic communication with a largely hearing world. Video relay services (VRS) allow Deaf (capital D in Deaf, refers to those people who use sign language as their primary language, and may or may not include people who are deaf and/or hard of hearing) and hearing communicants to understand one another via a remote sign language interpreter (as shown in Fig 1). In some cases, governments and vendors have put infrastructure in place that makes video relay services accessible to Deaf people at little or no cost. South Africa is severely behind in this regard; we have no such relay service, even at full cost. Many South African Deaf people engage non-governmental organizations, such as Deaf Community of Cape Town, DeafSA, National Institute for the Deaf and others to assist with interpreting services at governmental or healthcare offices.

The reason why Deaf people experience difficulty is due to the communication divide [1]-[2] that exists between Deaf people who communicate in sign language and the rest of the population who do not understand sign language. Kritzinger, Schneider, Swartz and Braathen [3] interviewed 16 Deaf participants from Worcester in the Western Cape to

investigate if there were any other reasons Deaf people find it difficult to access health care services. The study concluded that amongst other reasons, communication difficulties were the most noticeable. In [4], the authors bring into context the different laws that apply to Deaf South Africans when it comes to access to health care. They further explain that the law allows for the provision of interpreting services at health care facilities to assist with communication. Neglecting to provide interpreting services may result in incorrect diagnosis by health care professionals and misunderstood treatment plans for Deaf patients; ultimately leading to the violation of Deaf patients' rights. In reality, Deaf people will only benefit from interpreting services when they are actually available. There is a serious shortage of qualified sign language interpreters (SLI) as explained in [5], especially in health care [6].

In order to supplement the lack of SLI services in the South African context, Glaser and Tucker [7] together with a non-governmental organization (NGO) called Deaf Community of Cape Town (DCCT) researched the use of mobile phones to assist with the communication divide between the Deaf and hearing people. This work led to the design of a mobile communication aid called SignSupport, co-designed by Chininthorn, Glaser, Freudenthal, and Tucker [8] and Deaf communities across Cape Town. The design was based on the interaction between a Deaf person and a pharmacist when dispensing medication. Motlhabi, Glaser, Parker and Tucker [9] applied the in [8] design by developing a mobile prototype for SignSupport. SignSupport used pre-recorded sign language videos stored on the device for the Deaf patient and touch screen interactions for the dispensing pharmacist. This combination provided assisted communication between a Deaf person and the pharmacist, although with limited capabilities.

Buttussi, Chittaro, Carchietti and Coppo [10] developed a similar mobile application to allow emergency medical staff to interact with Deaf people in emergency situations, where Deaf people are not able to respond to verbal communication. The mobile application allowed emergency staff to obtain vital medical related information from the Deaf person, using translated sign language questions stored on the mobile phone as videos.

Both cases mentioned that interaction is limited to the amount of videos stored on the mobile phone which is restricted by 1) the size of the memory on the mobile phone

and 2) the quality of the videos stored; the better the quality the bigger the video file size.

To overcome the restrictiveness of the aforementioned applications, and to clarify what scripted communication cannot do, or simply to ask and answer questions, a more generalized solution is proposed: using Web Real Time Communication (WebRTC) to build a video relay service.

WebRTC is a collection of JavaScript APIs that allows for real time peer-to-peer (P2P) communication in web browsers and multiple devices [11][12]. The main benefit of WebRTC is, it allows for real time communication applicable to browser and mobile functionalities such audio and video calling by requesting access to cameras and microphones [13]. Our prototype application has a split screen (as opposed to picture-in-picture, see Fig 2) for the Deaf patient and the remote sign language interpreter, such that each can clearly see each other from the torso up to converse in sign language. For example, for clarification of the SignSupport for pharmacy scenario, we would have the situation shown in Fig 1.

A recent survey conducted with Deaf people in the Western Cape on their mobile phone usage showed that Deaf participants use video telephony applications to communicate with family, friends and NGO staff. Table 1 provides a breakdown of these applications compared to national statistics as compiled in “Digital Statistics In South Africa 2017”, a white paper compiled by Qwerty Digital [14] reporting on the digital statistics in South Africa. It states that 13 million South Africans use their mobile phones for social media.

TABLE 1: VIDEO CALLING APPLICATION USED BY DEAF COMPARED TO THE SOUTH AFRICAN NATIONAL STATISTICS

Video calling application	% Used by Deaf people as per the survey	% Used by South African as per the national statistics
Whatsapp	14	45
Facebook	3	49
IMO	33	n/a
Skype	7	19
Twitter	3	26
Facetime	3	n/a
Other	1	n/a

The results of the survey provides motivation for the development the mobile video relay application presented in this paper, as requested by the Deaf community. The only other requirement of the community is provide a 50/50 split screen view (Fig 2), as opposed to the traditional picture-in-picture view provided by the applications indicated in Table 1.

The rest of the paper is organized as follows: Section II discusses related work in video relay services, both commercial and freeware. We also provide an overview of some WebRTC implementations in various healthcare scenarios. This project is motivated by health care applications, although we are not specifically focused only on health care implementation. Section III looks at the WebRTC architecture, the three main component of WebRTC and their roles. Section IV presents our VRS prototype architecture, based on SimpleRTC, and shows how the video relay can be implemented with WebRTC and the signaling mechanisms provided by PubNub. Section V gives an overview of the paper and looks into future work and recommendations.

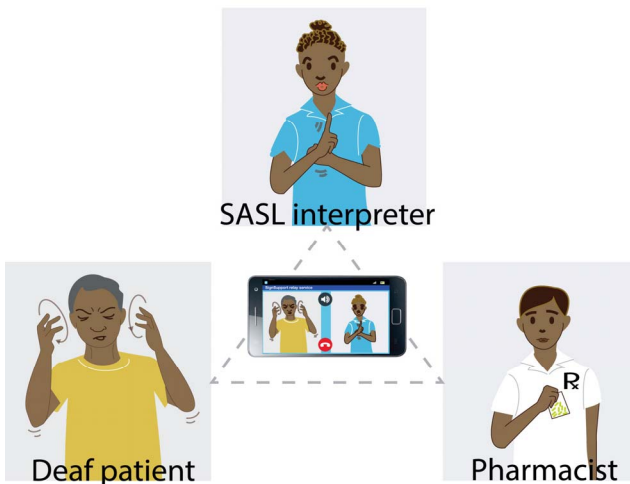


Fig 1: Three-Way WebRTC VRS communication

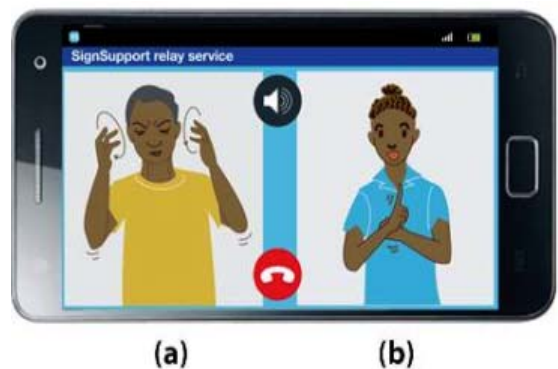


Fig 2: 50/50 Split screen application on mobile phone. The person on the left (a) is Deaf, and the person on the right (b) is a sign language interpreter. The hearing person speaks to the sign language interpreter using the phone’s microphone, and the sign language interpreter speaks to the hearing person over the speaker.

II. RELATED WORK

Related work comes primarily in two flavours: work done with video relay systems, and work done with WebRTC. To the best of our knowledge, the two have not been combined to provide a WebRTC system for VRS.

Video relay systems

Earlier investigation into video relay services for Deaf people [15] led to the review of many commercial applications. For example, IWRelay is an iPhone application developed by Healinc Telecom. Other mobile applications with similar functionality for Android are nTouch, PurpleVRS, and Sorenson Video Center (a mobile version of videophone implementation). A video relay service company from the United States, called Convo Relay, which is owned and operated by Deaf [16], developed VRS applications available on Android and IOS devices, Windows and MacOS operating systems. There are no cost to caller when enlisting interpreter services, as they funded by the Telecommunications Relay Services (TRS) fund in the United States [17].

The abovementioned applications are commercially available at a cost to the end user, or they require an in-depth knowledge of how to setup the application, or are not available in South Africa. Local Deaf people have identified the need for an application that is free, easy to install, easy to use and navigate, and low cost.

In the academic space, MobileASL [18][19][20] was a project of Washington State University to develop video compression codecs and software that allows Deaf people in the United States (US) to use American Sign Language two way communication over mobile networks using real-time video. MobileASL was mainly developed for mobile phones running Windows Mobile.

MobileASL was successfully tested in the US, not in South Africa, and it only runs on a Windows Mobile device, with no Android version available. Our prototype is platform independent and will run on most mobile operating systems, using WebRTC enabled browsers.

Social video relay systems

Table 1 shows that Deaf people use social networking applications such Facebook Messenger, IMO, Skype and WhatsApp for video conferencing. Although according to our survey, Deaf people in South Africa favour these applications, these applications do not enable the 50/50 split screen solution that local Deaf people suggested we use for three way video relay.

WebRTC systems and Healthcare

Researchers in Australia [12] used WebRTC to develop a low cost video conferencing system for tele-health. Their motivation was to use EasyRTC, which is an open source WebRTC video conferencing system to improve communication between home-based patients and a clinical care coordinator as part of a tele-home monitoring system. The system would allow the health care professional to have

online meetings with patients who may not be able to attend in-person for various reasons. The benefit of such a system would be improved patient care and access to healthcare. Patients were provided with a Samsung Galaxy Note mobile phone with the Chrome web browser pre-installed. They use Chrome to initiate video conferencing requests with any available clinical care coordinator through a web portal hosted on a Web/App server. The Web/App server forms part of the video conferencing architecture, and allows the clinical care coordinator (located in remote sites) to monitor for any video conferencing requests from patients. The implemented architecture also consists of a database server that stores video conferencing requests, and other relevant information such as date, time and duration of video conferencing sessions. To complete the video conferencing architecture, a media server assists with the real time video and audio communication. Session traversal tools included with the media server assist with obtaining the public IP addresses and port numbers of patient and clinical care coordinators devices, which usually reside behind firewall and network address translation (NAT). Therefore, both patients and clinical care coordinators use the same web portal to communicate with each other via a secure peer-to-peer channel using a WebRTC enabled web browser, from different locations, with WebRTC handling the connection setup, management and signaling of the video conferencing calls.

Antunes, Silva and Barranca [21] developed a similar tele-medicine solution using WebRTC for communication between patients and health professionals. Their solution added functionality such as bi-directional file sharing, and whiteboard sharing between participants. In one of their tests, they attached a digital microscope to a USB interface, where images were sent over WebRTC. They implemented the basic WebRTC infrastructure by implementing a web-based application to which both clients connect using JavaScript APIs in order to establish peer-to-peer connectivity. They used two web-browsers, namely Firefox and Chrome as clients, a web-server for WebRTC hosting, and a signalling server for the management of client connections. Interactive Connectivity Established (ICE) servers assisted with IP address and network address translation discovery, using Session Traversal Utilities for NAT (STUN) and Traversal Using Relays around NAT (TURN) services.

III. WEBRTC ARCHITECTURE

WebRTC brings a new dimension to video relay and/or video calling because of the fact that it runs in most browsers and mobile phones, making it platform-independent and easy to implement. It provides access to audio and video functionality for P2P conversations and data sharing without the need to download additional software. WebRTC is essentially a collection of APIs and protocols, which is undergoing API standardization [22][23] by the

World Wide Web Consortium (W3C) [24] and the Internet Engineering Task Force (IETF) [25].

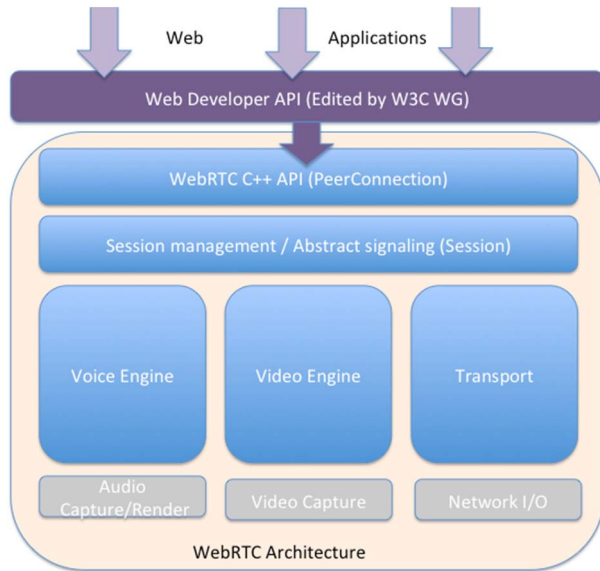


Fig 3: Overall WebRTC architecture diagram

The overall WebRTC architecture as retrieved from the WebRTC (www.webrtc.org) website is shown in Fig 3. WebRTC consists of three main components [13][26][22] namely:

MediaStream API - allows the browser to access the local input and output devices, such as microphones and cameras. MediaStream is responsible for setting up a media stream to devices, by creating a MediaStreamTrack for audio or video per device.

RTCPeerConnection API – sets up the secure peer-to-peer connection between WebRTC enabled browsers and peers through the Secure Real-Time Transport Protocol (SRTP). RTCPeerConnection handles signaling and NAT traversal for peers that are situated behind firewalls, and network address translation (NAT). Signaling is the process of setting up, controlling, and terminating a communication session. Peers need to be connected through signaling by passing Session Description Protocol (SDP) between one another. SDP contains information about peers, such as location and ports, codec support and data transmission method [27]. After successful RTCPeerConnection establishment between peers, data can be exchanged via the RTCDataChannel. To assist with connection of peers, RTCPeerConnection uses Interactive Connectivity Establishment (ICE) server URLs to the peer. ICE find the best possible path to connect two peers and uses Session Traversal Utilities for NAT (STUN) and Traversal Using Relays around NAT (TURN) servers for address translation of peers.

RTCDataChannel API – allows for a secure bi-directional data channel via Stream Control Transmission Protocol (SCTP) encapsulated in Datagram Transport Layer Security (DTLS). The data channel allows for the reliable and unreliable exchange of data.

IV. VRS PROTOTYPE ARCHITECTURE

Video relay service is an application that allows a three-way conversation between a Deaf person and a hearing person via a remote sign language interpreter (SLI). Our intended scenario is for the Deaf and hearing persons to be co-located. The sign language conversation between the Deaf person and SLI is enabled using a 50/50 split screen on any WebRTC enabled web browser on computer or mobile phone. The prototype described in this paper differs from those discussed in the related work section based on the following: (1) it is not a commercial application, developed using open source components, (2) it is platform independent, and can run on any computer with any operating system with a WebRTC enabled web-browser, like Firefox or Chrome, (3) will run on most mobile devices running the Android operating system, and (4) will provide 50/50 split screen as requested by the Deaf community.

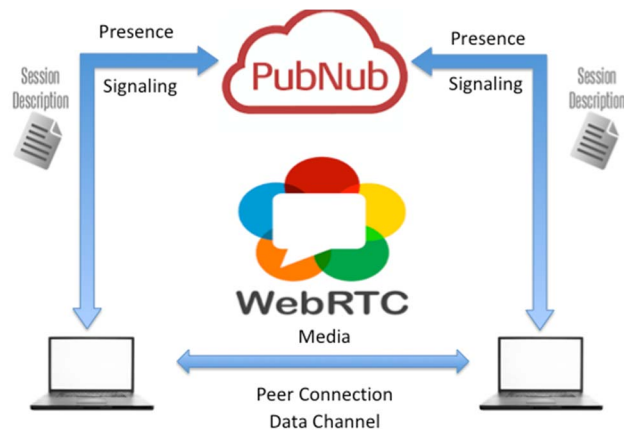


Fig 4: WebRTC and PubNub Signaling

As discussed in the previous section, WebRTC allows access to browser (peer) internal input/output devices such as microphones and cameras to make video calls. It will establish a secure connection between browsers, and securely exchange data between browsers. Signaling does not form part of the WebRTC architecture and a separate signaling sever is required to coordinate communication and metadata needs to be sent between browsers.

PubNub provides signaling as well as additional functionality as shown in Fig 4, such as Presence, e.g. when a peer A wants to establish a connection with peer B, peer A needs to know whether peer B is online or offline before establishing a connection. Presence also informs peer A on what device peer B is using and whether peer B is available for connection establishment. Storage and Playback is another added functionality of PubNub, and allows for the storage of messages e.g. chat conversation histories, which is stored for playback at another time.

WebRTC Video Relay Requirements

1. A WebRTC server, the portal through which clients will connect.

2. PubNub signaling server to exchange session description protocol information, to keep the communication active until it needs to be terminated.
3. WebRTC enabled clients to test the communication.

WebRTC Video Relay Implementation

There are various examples of WebRTC implementations. This paper focuses on extending the WebRTC implementation called SimpleRTC with JavaScript [28].

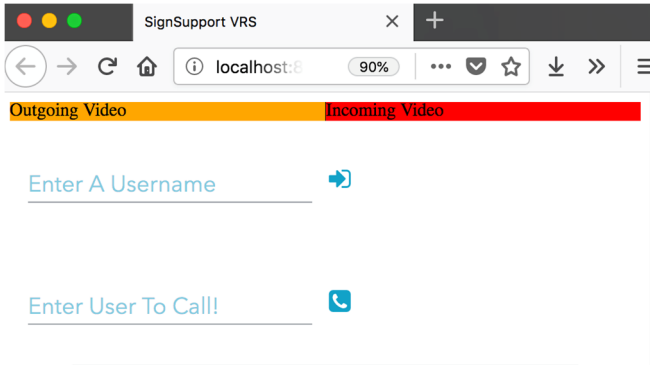


Fig 5: Video Relay Client Portal

The WebRTC client portal is an HTML page comprised of:

1. two HTML <div> elements that contain the incoming and outgoing videos, which represent the two users in the video relay,
2. a login field for the client login, and
3. a calling field to initiate a call to another online client.

After the username is entered and a connection is established, the browser (WebRTC DataStream) will request permission to access the multimedia devices, see Fig 5. In Figure 6 the client browser is shown requesting access to the local camera and microphone as per the function of the DataStream component of WebRTC using the code below. Once the user has granted camera and microphone access, the video is placed in the outgoing div container defined earlier in the code. This allows the user to see himself as well as the other person involved in the video relay.

```
var ctrl = window.ctrl = CONTROLLER(phone);
ctrl.ready(function(){
  form.username.style.background="#55ff5b";
  form.login_submit.hidden="true";
  ctrl.addLocalStream(vid_thumb);
  addLog("Logged in as " + form.username.value);
});
```

The user logs into the portal, and connects to the PubNub network using secure keys, generated by the PubNub system. The keys are called publish and subscribe keys, and are used for application authorization and to setup a channel through which users can communicate. The underlying

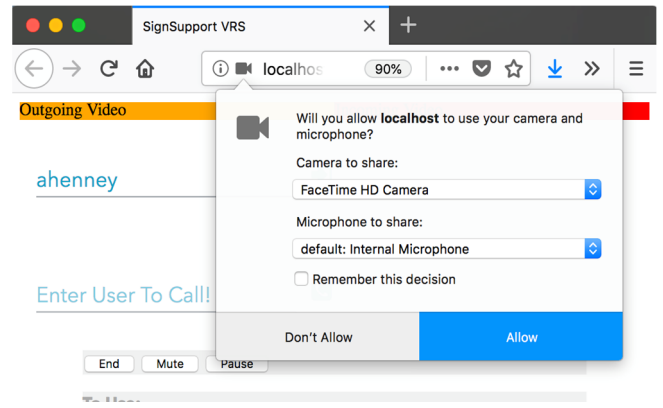


Figure 6: DataStream request access to media devices

WebRTC and PubNub APIs take care of the connection to PubNub to setup signaling and exchange SDP messages. Fig 7 displays the video relay web portal after a user has logged in and the camera feed is placed in the outgoing div container.

Once both parties are logged in and present on the PubNub system with their Unique User Identification (UUID) for the specific communication channel, they are able to call another user. If a user is not logged into the PubNub presence function, it will respond with a message that the user is not logged into the portal and therefore not online. The code snippet below describes how a call is made:

```
function makeCall(form){
  if (!window.phone) alert("Login First!");
  var num = form.number.value;
  if (phone.number()===num) return false; // No calling yourself!
  ctrl.isOnline(num, function(isOn){
    if (isOn) ctrl.dial(num);
    else alert("User is Offline");
  });
  return false;
}
```

If, however, both users are logged into the portal, any user can initiate a call to another. Once connection is established the user will be connected via a peer-to-peer connection. Both users will be able to see themselves and each other.

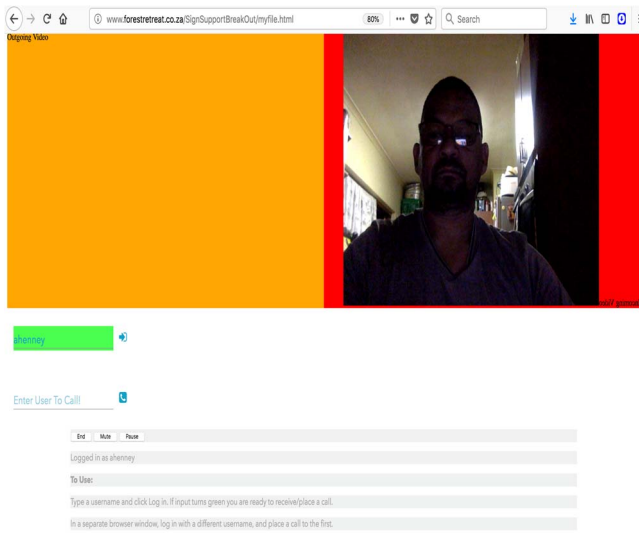


Fig 7: User logged in Video Relay Portal

V. CONCLUSION AND FUTURE WORK

In conclusion, this paper discussed the need for a video relay application for Deaf people in South Africa. We presented literature stating the importance of a video relay and access to an interpreter by Deaf people in South Africa. We presented a brief overview of both commercially available VRS solutions available. We discussed the most commonly used social video chat applications used by Deaf people in the Western Cape. A limitation to this paper is that testing the prototype with the Deaf community was not implemented as Deaf community participation is based on availability of Deaf participants, which is arranged by the Deaf organizations we collaborate with, however testing should be implemented in Q2 of 2019. This will allow for their input and feedback regarding the prototype. In comparison to the applications covered in the related work section, the prototype will be free, will feature the 50/50 split screen and platform independent, as Firefox and Chrome are of the most widely used web browsers on PC and mobile architectures. Furthermore this research can be used to lobby mobile network providers to allow free or heavily discounted connectivity to WebRTC websites for video relay, to reduce the cost the data consumed when Deaf people access it.

REFERENCES

- [1] D. Reeves and B. Kokoruwe, "Communication and communication support in primary care: A survey of deaf patients," *Audiol. Med.*, vol. 3, no. 2, pp. 95–107, 2005.
- [2] D. B. Scheier, "Barriers to health care for people with hearing loss: a review of the literature," *J. New York State Nurses Assoc.*, vol. 40, no. 1, pp. 4–10, 2009.
- [3] J. Kritzinger, M. Schneider, L. Swartz, and S. H. Braathen, "'I just answer 'yes' to everything they say': Access to health care for deaf people in Worcester, South Africa and the politics of exclusion," *Patient Educ. Couns.*, vol. 94, no. 3, pp. 379–383, 2014.
- [4] H. J. Haricharan, M. Heap, F. Coomans, and L. London, "Can we talk about the right to healthcare without language? A critique of key international human rights law, drawing on the experiences of a Deaf woman in Cape Town, South Africa," *Disabil. Soc.*, vol. 28, no. 1, pp. 54–66, 2013.
- [5] D. Federation, S. Africa, N. Road, and R. P. Bag, "POLICY ON THE Provision and Regulation of South African Sign Language Interpreters," no. June 2009, pp. 1–46, 2011.
- [6] T. Zulu, M. Heap, and E. Sinanovic, "PLOS ONE The Cost and Utilisation Patterns of a Pilot South African Sign Language Interpreter Service for Primary Health Care Services in South Africa," *PLoS One*, vol. 12, no. 12, pp. 1–11, 2017.
- [7] M. Glaser and W. D. Tucker, "Telecommunications bridging between {D}eaf and hearing users in {S}outh {A}frica," in *Proc. CVHI, Granada, Spain*, 2004.
- [8] P. Chinthorn, M. Glaser, A. Freudenthal, and W. Tucker 4, "Mobile Communication Tools for a South African Deaf Patient in a Pharmacy Context," *Conf. Proc.*, pp. 978–1, 2012.
- [9] M. B. Motlhabi, M. Glaser, M. Parker, and W. D. Tucker, "SignSupport: A Limited Communication Domain Mobile Aid for a Deaf patient at the Pharmacy," *South. African Telecommun. Networks Appl. Conf.*, pp. 173–178, 2013.
- [10] F. Buttussi, L. Chittaro, E. Carchietti, and M. Coppo, "Using mobile devices to support communication between emergency medical responders and deaf people," *Proc. 12th Int. Conf. Hum. Comput. Interact. with Mob. devices Serv. - MobileHCI '10*, p. 7, 2010.
- [11] C. Jennings, "WebRTC: Real Time Communications for the Web," no. May, 2015.
- [12] J. Jang-Jaccard, S. Nepal, B. Celler, and B. Yan, "WebRTC-based video conferencing service for telehealth," *Computing*, vol. 98, no. 1–2, pp. 169–193, 2016.
- [13] B. Sredojevic, D. Samardzija, and D. Posarac, "WebRTC technology overview and signaling solution design and implementation," *2015 38th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2015 - Proc.*, pp. 1006–1009, 2015.
- [14] Qwerty, "The Digital Landscape in South Africa 2017," *Digital Statistics in South Africa*, 2017. [Online]. Available: <http://qwertydigital.co.za/wp-content/uploads/2017/08/Digital-Statistics-in-South-Africa-2017-Report.pdf>.
- [15] A. J. Henney and W. D. Tucker, "Information Protection in Video Relay Services," in *Proceedings of the 4th Annual Symposium on Computing for Development*, 2013, p. 24:1–24:2.
- [16] M. Friedner, "Understanding and Not-Understanding: What Do Epistemologies and Ontologies Do in Deaf Worlds?," *Sign Lang. Stud.*, vol. 16, no. 2, pp. 184–203, 2016.
- [17] C. Communications, "What is Video Relay Service?" [Online]. Available: <https://www.convorelay.com/vrs>.
- [18] J. J. Tran, J. Kim, J. Chon, E. A. Riskin, R. E. Ladner, and J. O. Wobbrock, "Evaluating quality and comprehension of real-time sign language video on mobile phones," in *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, 2011, pp. 115–122.
- [19] R. Balaji and G. Naveen, "Secure data transmission using video Steganography," in *Electro/Information Technology (EIT), 2011 IEEE International Conference on*, 2011, pp. 1–5.
- [20] J. Kim, J. J. Tran, T. W. Johnson, R. Ladner, E. Riskin, and J. O. Wobbrock, "Effect of MobileASL on communication among deaf users," in *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems - CHI EA '11*, 2011, p. 2185.
- [21] M. Antunes, C. Silva, and J. Barranca, "A Telemedicine Application Using WebRTC," *Procedia Comput. Sci.*, vol. 100, pp. 414–420, 2016.
- [22] D. Sam, *Getting Started with WebRTC - HTML5 Rocks*. 2014.
- [23] C. Fan, "Research on Development and Evaluation of WebRTC Signaling based on XMPP," no. June, p. 118, 2017.
- [24] WebRTC, "WebRTC." [Online]. Available: <https://webrtc.org/>.
- [25] W. W. W. Consortium, "WebRTC 1.0: Real-time Communication Between Browsers." [Online]. Available: <http://w3c.github.io/webrtc-pc/>.
- [26] S. El Jaouhari *et al.*, "Toward a Smart Health-care Architecture Using WebRTC and WoT To cite this version: HAL Id: hal-01503434 Toward a Smart Health-care Architecture Using WebRTC and WoT," 2017.
- [27] PubNub, "What is WebRTC?" [Online]. Available: <https://www.pubnub.com/blog/2014-01-14-what-is-webrtc/>.
- [28] K. Gleason, "SimpleRTC." [Online]. Available: <http://kevingleason.me/SimpleRTC/>.
- [29]