**Development of a browser based real-time peer-to-peer remote support application**

Assistive technology has been in use in factories for a few years now. Also known as remote support applications, they enable on-site personnel to repair malfunctions under support of experts, while they are connected via audio and video stream. For companies, this brings the big advantage that disruptions can be repaired much quicker, without the necessity of an expert having to be physically present.

Proprietary video chat applications come with a few disadvantages, though: The data flows over a third party server. Companies that deal with sensitive data might not want that, as they can never be sure that their data does not get into the wrong hands. Furthermore, the data always streaming over a server is automatically coming with a higher latency for the data transfer.

To eradicate these problems, Web Real-Time Communication (WebRTC) could be used instead of a proprietary video chat application. It enables users to communicate via audio and/or video stream directly in the browser without the need of proprietary software or external plugins. With WebRTC, a server is only needed so that users can discover each other and set up a connection. After that the data flow is going directly from user to user, or peer-to-peer, thus bringing lower network latency. Furthermore, data encryption is mandatory for all WebRTC components.

At the beginning of this thesis, the core components of WebRTC will be discussed, as well as the various possibilities it offers to developers. A concept will be created that describes the aspects and components that are necessary to develop a remote support application with WebRTC that works peer-to-peer directly through the web browser.

It should be possible for a user to call another user through the web browser and connect to them via audio/video stream. The called user can see what the calling user sees on his/her device and highlight any parts of the video screen with coloured geometrical shapes to help them find a solution for a problem. Additionally, the users should be able to communicate through a text-based chat on the same web page. To avoid too much complexity, users will not have to login to use the web page.

To demonstrate the practical functionality of the concept, a prototype will be developed. The prototype consists of a simple management server, whose sole purpose is that users can discover each other via a web page to start an audio/video call through WebRTC. After the call has been initiated via WebRTC, all data exchange and signaling happens directly peer-to-peer, without the management server.

To set up a peer-to-peer connection, the public IP address of both users in the call must be known. In a real life environment, this might be a problem if one or both of the users is located in a network that uses Network Address Translation (NAT). To solve this problem, WebRTC uses Session Traversal Utilities for NAT (STUN) and Traversal Using Relay NAT (TURN) servers to discover a user’s public IP address. This is not necessary if both users are located in the same network. For this prototype, it is assumed that both users are part of the same network.

The developed prototype will undergo tests regarding the quality of the audio/video stream connection on a mobile browser in an area with limited WLAN reception. Because WebRTC offers the possibility to pass constraints on the camera’s picture quality, different levels of picture quality will be assessed within these tests.

Because WebRTC is still under development, there are only a few web browsers that fully support it. Therefore, only Chrome, Firefox and their mobile versions will be used.

Conclusively, all insights and experiences from the prototype development as well as the test results will be discussed in detail.

For the future, the prototype could be extended by adding the possibility to set up a connection if one or both users are located in a network that uses NAT. This will require the use of STUN and TURN servers.

Another possible extension would be to enable more than two users to communicate at a time. A multipoint control unit (MCU) will be necessary to enable this functionality.