

# **OATS: An Architectural Design and Implementation for Telepresence Robots**

Swapnil Kalhapure  
Department of Computer Engineering,  
A.I.S.S.M.S's, College of Engineering,  
Pune, India

## **ABSTRACT**

Robotic Telepresence systems focus majorly on the concept of telecommunication and remote presence with independent mobility. Recently a number of Telepresence robots have been introduced in the market, each with some unique feature and show-off factor. While these systems promise good user experience, they are too expensive for the general public to use them. This paper proposes an innovative approach to build a commercial robotic Telepresence system called OATS. OATS stands for “Open Access Telepresence Systems”, which is an architecture primarily based on open source platforms like, WebRTC and MQTT. The uniqueness of the architecture is that, the Telepresence robot can use any mobile smartphone running an OATS client application, as the brain of the robot, reducing the need of expensive, onboard electronics and processing power. The smartphone connects to a robotic platform that enables the robot to move. Using this robot, a person can remotely interact using any smartphone or laptop. This OATS client connects to a server which provides services required for peer to peer videoconferencing and machine communication between the clients. A user can login to one of this Telepresence robot and pilot it in a remote area, from any part of the world. This architecture not only reduces the cost of the Telepresence robot, but the use of smartphone platform also opens up whole new possibilities for development of next generation of telepresence robots. In this paper, we briefly discuss the OATS architecture, its deployment strategies and its advantages over existing systems.

## **General Terms**

Virtual-Presence, Client-Server architecture, robotics, communication, Real Time Communication, Intel Edison, Authentication

## **Keywords**

Telepresence, WebRTC, MQTT, audio, visual, Telerobotics, Teleoperations.

## **1. INTRODUCTION**

Recent advances in information and communication technologies offer a variety of methods for exchange, out of which Telepresence systems offer a possibility of feeling present in remote location. Although this concept is not new in today's fairly advanced technology, but the commercialization of the technology is still at its very initial stages. Telepresence system is an area that includes human-machine interface, communication technologies and robotics. The system provides a way of communication between users and distant people or a remote environment. Basically a Telepresence robot is a system which integrates video conferencing equipment into a mobile robotics platform that is remotely piloted. Such robots have displays, microphones, speakers for streaming video and audio between two users.

Also the use of a mobile robotic platform enables the system to navigate in a remote area using control-interface provided by the smartphone application to a user at distant place. In the era of global communication, Telepresence technology can prove to be a boon for worldwide communications and many IT organization and industries have started using such systems. Most Commercial robotic Telepresence system in the market today, are focused on building systems that use state of the art technologies for providing a good user experience. However the extremely high cost of such systems tends to outweigh the advantages offered.

This paper discusses the design, implementation and deployment of a Telepresence system based on a proposed architecture called OATS or “Open Access Telepresence System”. This architecture is completely based on open source technologies and affordable off-the-shelf hardware. The architecture is proposed for developing a scalable commercial Telepresence system that can be mass-manufactured and easily deployed on the existing internet infrastructure. The paper is organized as follows: section 2 provides a description of the related work done in recent past. Section 3 will discuss a general overview of the OATS architecture. The sections 4 and 5 describe the client and server architecture for OATS. Section 6 states the advantages and future scope of this architecture.

## **2. RELATED WORK**

The application of Telepresence has been receiving significant attention from the robotics community in the recent years. The concept of Telepresence was first proposed by Marvin Minsky, the founder of MIT's artificial Intelligence laboratory, in the OMNI magazine in 1980 where he mentioned various needs and applications. In an effort to create mutually immersive communication using Telepresence robots, HP built its early version of Surrogate in 2002, which percept audio and video of being in a remote location. Aided by improved technologies in network communication, Cisco launched Cisco Telepresence video conferencing system in 2006. The system provided high definition video and realistic sound displays, allowing remotely connected users to feel as if they were in the same room.

In the recent years more and more Telepresence products have been pushed in the market. Following the development of Telepresence, various people and organization started making semi-automated Telepresence systems that included various modifications such as person following system, automation of audio-visual attention and overcoming various occlusions.

Akansel Cosgun et al.[1] presented a method for building an autonomous mobile robot to follow a person where there is an interaction between the robot and human while following. The remote user connects to the robot via Wi-Fi and communicates with others using Skype. Nadia Magnenat et

al.[2] discussed the various forms of Telepresence, how they are related and different from each other and how autonomy takes place in Telepresence.

Sina Radmard et al. [3] presented a blockage handling system, applied to Telepresence platforms, that not only reduces the control burden on the operator but also ensures the continuity of the visual interaction. Katherine M. Tsui et al. [4] focused on how a Telepresence robot operator, the people with the robot, and the robot itself collaborate so that the operator reaches the intended destination. Keng Peng Tee et al. [5] presented an audio-visual attention control system to establish focus of attention for a pan-tilt Telepresence robot. Their system is able to detect attentions automatically and reliably in a Telepresence scenario.

The backbone of all the above mentioned Telepresence systems includes video conferencing capabilities. Most of these mentioned systems use Skype, Face-time or some other video-conferencing tools that follow the client-server model for their transfer of audio-video streams. Since there is usage of servers for handling of data, these systems have a single point of failure i.e. the servers used for video-conferencing. To overcome many such issues and lessen the dependency on servers, Google released an open source project for browser-based real-time communication known as WebRTC in May 2011. Since the need of servers is very minimal, user experience is not dependent on servers anymore.

Since its birth in 2011 various developments have been made in making WebRTC available to everyone. Sergej Alekseev et al. [6] presented an approach for analyzing the behavior of WebRTC-based applications, typically being used for direct browser-to-browser communication. Alan Johnston et al. [7] illustrated and discussed a number of issues that are specific to WebRTC enterprise usage. Deploying multimedia communication based on WebRTC API for more than peer to peer browser-based connection is not inherently supported and creates new challenges in terms of media stream distribution as well as membership and conferencing operation control overhead. Many conference models presented by Wajdi Elleuch[8] can support multi-party communication between end- points over an IP networks.

Jukka K. Nurminen et al. [9] reviews whether the new HTML5 and WebRTC standards are fit for P2P video streaming, evaluates the performance challenges and proposes solutions for them. They provide preliminary experimental data on WebRTC which measures the performance of such a system in a laboratory environment.

Now looking at the opportunity of building smarter and more scalable Telepresence robotic systems, this paper has suggested a new methodology to implement Telepresence systems using open source technologies, to make the system more affordable and useful.

### 3. OVERVIEW

The primary objective of designing the OATS architecture is to develop a scalable and affordable Telepresence system for the general public. In any typical robotic Telepresence scenario a user remotely drives a robot using a laptop or browser, which is capable of videoconferencing and moving around in a remote location. The OATS Architecture is basically divided into two parts: the server and the smartphone application based clients, that is either the robot or the remote controller.

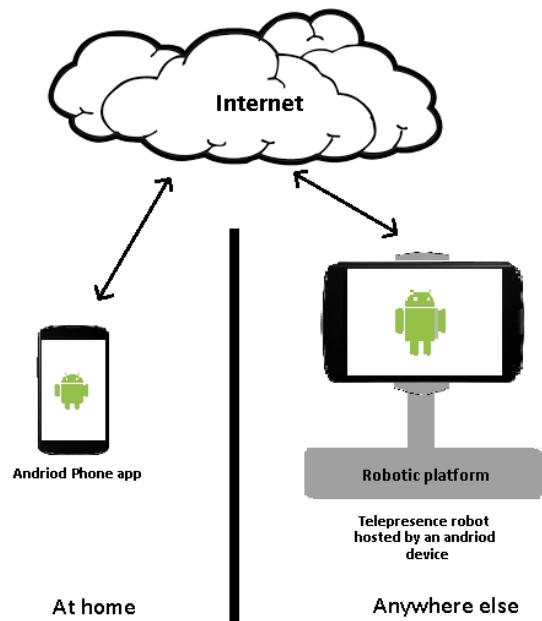


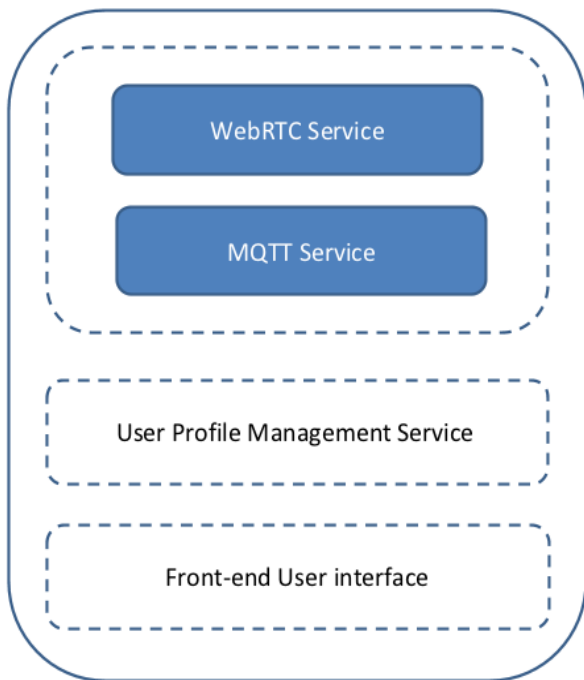
Fig. 1 General OATS Telepresence scenario

So basically, a smartphone device acts as brain for a Telepresence robot and another smartphone/tablet/laptop is used as the remote controller to control the robot. The smartphone application first connects to the server, then depending on if it's connected to a robotic platform, either controls the robotic platform or provides an interface to interact with the robotic platform. The OATS client architecture divides the robot into two parts which are the brain and body. The robotic client that is basically a smartphone or tablet over a robotic platform is running an OATS application. This application uses the smartphone camera and microphone, which are usually already optimized for hands free video calling, for video conferencing. The smartphone also connects to the hardware of the robotic platform via Bluetooth/ WiFi, which is the simplest wireless connectivity option for a smartphone. Using Bluetooth/WiFi the smartphone application can control the robotic platform to do a desired task. The robotic platform houses a built-in power supply unit, motor driver microcontrollers, Bluetooth module that enables the robot to move in the surroundings and communicate via Bluetooth/ WiFi RF link .At the other end, another smartphone application, which connects to the OATS server, acts as a controller for the robot that has a user interface which displays the video conference feed, as well as the drive-control interface. So in a complete scenario that can be seen in figure [1] two smartphone application are connected to each other via an OATS server. One smartphone client application provides a video-audio conferencing and control interface, while the other smartphone client is connected to a robotic platform. The OATS architecture allows the developer and manufacturers to design and deploy customized versions of applications and robotic hardware. In the next two sections a brief description of the OATS server and client architecture is discussed.

### 4. OATS SERVER ARCHITECTURE

The OATS server is an important key for the OATS architecture. It is responsible for hosting two important services for the telerobotics system. One of the services is a WebRTC based video conferencing application that provides a peer to peer communication between devices. WebRTC is basically an open source project, that supports browser-to-

browser applications for voice calling, video chat, and P2P file sharing without the need of either internal or external plugins. This service is only required for the initial call setup phase where the clients share each other's internet address.



**Fig 2. OATS server architecture**

Once the call setup phase is completed, the devices can directly share audio-video feeds with each other. Another advantage of such service is that the server is free from the burden of managing client audio-video feeds. The other service provided by the OATS server is basically a MQTT broker service. MQTT (Message Queue telemetry transport) protocol service enables machine to machine communication. In this case it provides machine to machine communication between the smartphone clients. This is required to communicate the hardware control signals from one device to another in real time. The broker basically enables smartphone clients to communicate by publishing and subscribing to communication channels. We will now see each module of the OATS server in detail.

### **4.1 User Profiling**

The server is not only responsible for providing telerobotics services but it is also used for user profile management. Basically, this module is responsible for keeping a profile record of the Telepresence robots as well as the users that use them. This centralized approach helps them to keep track on the Telepresence systems and to disengage certain services in case of malicious access. Though the service is peer-to-peer the OATS server is still in control of the communication channel and can control the channel use.

### **4.2 WebRTC**

WebRTC as discussed early is an open source project that enables peer-to-peer communication between browsers. The smartphone application uses Web View, which is a view of the browser inside your application that gives it the ability to share application data and perform teleconferencing peer-to-peer (P2P). Historically RTC has been corporate and complex, but with the introduction of WebRTC, we have an open standard for real-time, plugin-free video and audio

communication. In case of OATS architecture, it provides a gateway for audio-video conferencing between clients enabling them to share audio-video stream with each other.

### **4.3 MQTT**

Another major aspect of robotic Telepresence systems is machine to machine communication that is addressed by MQTT. It is a machine to machine protocol that communicates control signals from one smartphone client to another smartphone client in real-time. It provides flexibility in case of multiple connections for Telepresence robots. It is ideal for these system as it is small in size, has minimized data packets and has efficient distribution of data over to other clients

## 5. OATS CLIENT ARCHITECTURE

The OATS client is basically a smartphone application that can work either as the “brain” of the Telepresence robot or as the remote controller for users. For the sake of understanding let’s call the smartphone client that connects to robotics platform as robot client and other device that is used as the controller as user client. In a typical scenario one user client and one robot client would be connected to each other via the OATS server. One applications is connected to a robotic platform that forms the robotic Telepresence device whereas the other application is used as a remote controller for the robot at a distant place. These clients use WebRTC based call setup service provided by OATS server to initiate peer-to-peer video conferencing with each other. This allows two-way audio-video communication between the clients. As most smartphone devices are already designed for video conferencing, so instead of embedding independent microphones, screens etc. on the robot, we utilize the hardware of the smartphone itself.

The devices also use the MQTT broker service to send data to each other. Basically the data is control signals used to pilot the robot and the status signal of the robot client. The control signals are for the robot client sent by the user client to pilot the robot remotely. Whereas the status signal are sent from the robot client, to notify the robot's position, battery status and other details to the user client. As any commercial smartphone, it provides no functionality to interface in external hardware. The OATS client uses Bluetooth to connect with the hardware of the robotic platform. The message sent via the MQTT communication link are communicated via the Bluetooth to the robotic platform which pilots the robot with smartphone. The robotic platform on the other hand is the peripheral connecting device that connects with the smartphone. Figure-3. Shows the typical architecture of the Telepresence robot which is the combination of the smartphone and the robotic platform. This extra piece of hardware (robotic platform) has a power supply unit, motor controlling mechanism and an IMU (Interial measurement Unit) sensor to capture angular velocity and linear acceleration data. This data is used to stop any activity to avoid damage caused to itself.

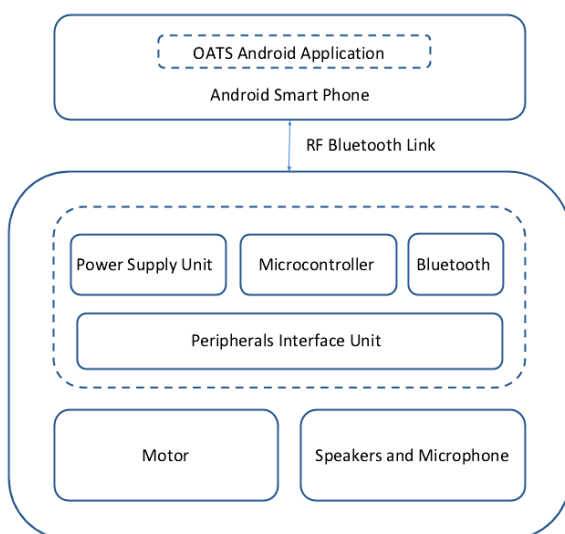


Fig 3. OATS client architecture

## 6. ADVANTAGES

This architecture ensures minimalistic hardware requirement, while achieving every distinct feature of a Telepresence robot. Using OATS server-client architecture telerobotics platforms can be built for surveillance, medical robots etc. An implementation of the above architecture can prove helpful in the field of Internet of things. The use of open source hardware will help hardware-hobbyist collaborate and connect their hardware developments, which will enable them to extend its capacity to new level. The architecture can also help develop robots that can control IOT-enabled house appliances, making it not only a home automation enabled robot but also a home security robot. Our further interests are focused on making this system autonomous and more innovative by adding new features like semi-autonomous navigation and a facial tracking system to make it more interactive.

## 7. CONCLUSION

With further advanced implementation with encryption and security the same system can be deployed for commercial use. When deployed the end user will only have to bear hardware fees and a minimal fees for use of OATS service for getting a fully functional robotic Telepresence device. This architecture not only makes the Telepresence system affordable but also opens a gateway for new applications of telerobotics with the use of the smartphone platform.

## 8. ACKNOWLEDGMENT

This work was supported by the Department of Computer Engineering, AISSMS College of Engineering, Pune.A special thanks to Adesh Atole (Sinhgad Academy of Engineering) for his helpful suggestion.

## 9. REFERENCES

- [1] Nadia Magnenat Thalmann , Zerrin Yumak , Aryel Beck "Autonomous Virtual Humans and Social Robots in Telepresence", 2014 IEEE 16th International Workshop on Multimedia Signal Processing (MMSp), Jakarta, Indonesia, Sep. 22-24, 2014
- [2] Sergej Alekseev, Christian von Harscher, Marco Schindler, "Finite State Machine based Flow Analysis for WebRTC Applications", 978-1-4799-4233-6/14/ 2014 IEEE
- [3] Florian Rhinow, Pablo Porto Veloso, Carlos Puyelo, Stephen Barrett, Eamonn O Nuallain, "P2P Live Video Streaming in WebRTC", 978-1-4799-3351-8/14/ 2014 IEEE
- [4] Katherine M. Tsui, Adam Norton, Daniel J. Brooks, Eric McCann, Mikhail S. Medvedev, and Holly A. Yanco "Audio-Visual Attention Control of a Pan-Tilt Telepresence Robot", 2013 13th International Conference on Control, Automation and Systems (ICCAS 2013) in Kimdaejung Convention Center, Gwangju, Korea Oct. 20-23, 2013.
- [5] Akansel Cosgun, Dinei A. Florencio and Henrik I. Christensen "Autonomous Person Following for Telepresence Robots", 2013 IEEE International Conference on Robotics and Automation (ICRA) Karlsruhe, Germany, May 6-10, 2013
- [6] Alan Johnston, John Yoakum, and Kundan Singh, Avaya Inc, "Taking on WebRTC in an Enterprise", 0163-6804/13/ 2013 Communications Magazine April 2013

- [7] Wajdi Elleuch, “Models for Multimedia Conference between Browsers based on WebRTC ”, 2013 Sixth International Workshop on Selected Topics in Mobile and Wireless Computing, 978-1-4799-0428-0/13/ 2013 IEEE
- [8] Jukka K. Nurminen, Antony J. R. Meyn, Eetu Jalonen, Yrjo Raivio and Ra’ul Garc’ia Marrero, “P2P Media Streaming with HTML5 and WebRTC”, 978-1-4799-0056-5/13/ 2013 IEEE
- [9] Sina Radmard and Elizabeth A. Croft "Overcoming Occlusions in Semi-Autonomous Telepresence Systems", 978-1-4799-2722-7/13/ 2013 IEEE Science, 1989