ZENNOUCHE Douria
RIVOAL Alice

ViseoConferenceGoogleVR

ZENNOUCHE Douria
RIVOAL Alice

OUTLOOK:

ZENNOUCHE Douria
RIVOAL Alice

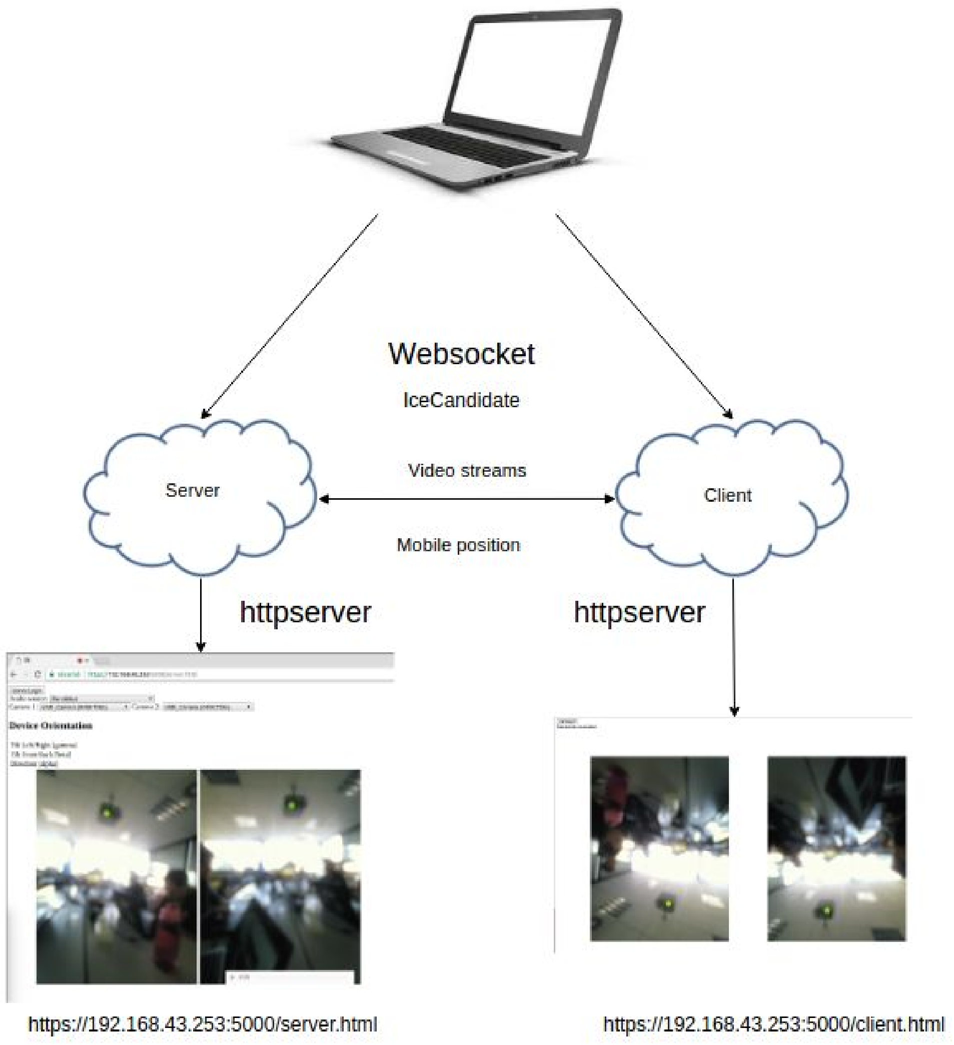
● Introduction:

Our product is a software that makes the link between two video and the user mobile phone. This software includes a set of functionalities.

First of all it makes the connexion between the phone and camera (what we call in computer science the client and the server). Then the user is supposed to be able to explore the space in which the robot is in by moving his head in whichever direction he wants. This kind of motion will create a request that will be sent to the robot and make the camera move in the same direction as the user's head.

● Used Technologies:

Architecture:



For this project we used many technologies.

For the video part:

Chrome: a browser that supports all the technologies and methods needed for webRTC.
WebRTC : a technology that allows a real time communication between two devices. It is nowadays used in many apps such as skype.
Websocket: webRTC itself uses websockets at first. When a client reaches a server to give it its IPadress it uses websocket.
STUN protocol: it is also used in webRTC at the second stage of the connection to make the two clients reach each other for the first time.
RTCpeerConnection: the clients are iceCandidates and once they are connected to each other via STUN protocol they make an RTCpeerConnection to be finally able to send streams back and forth from client1 to client2 and vice versa.

For the cameras motion part:

on the phone(client):

android listeners: to notify any position changes of the phone.

on the computer (server):

cylon : that allow to pilot the cameras by reaching the arduino directly
arduino: a software that can directly control the robot’s motors.
gort : used to program Arduinos

For the security part:

https: This protocol allows to make a secure connection in that only tho machines having the same certificate can receive the data.

WebRTC

WebRTC (**Web Real-Time Communication**) is a collection of protocols communications and application programming interfaces that enables real-time communication using peer-to-peer connections.

First, Websocket using port 5568 is created to connect the two clients (named client and server). Client applications need to traverse NAT gateways and as part of this process, the WebRTC APIs use STUN servers to get the IP address of your computer.

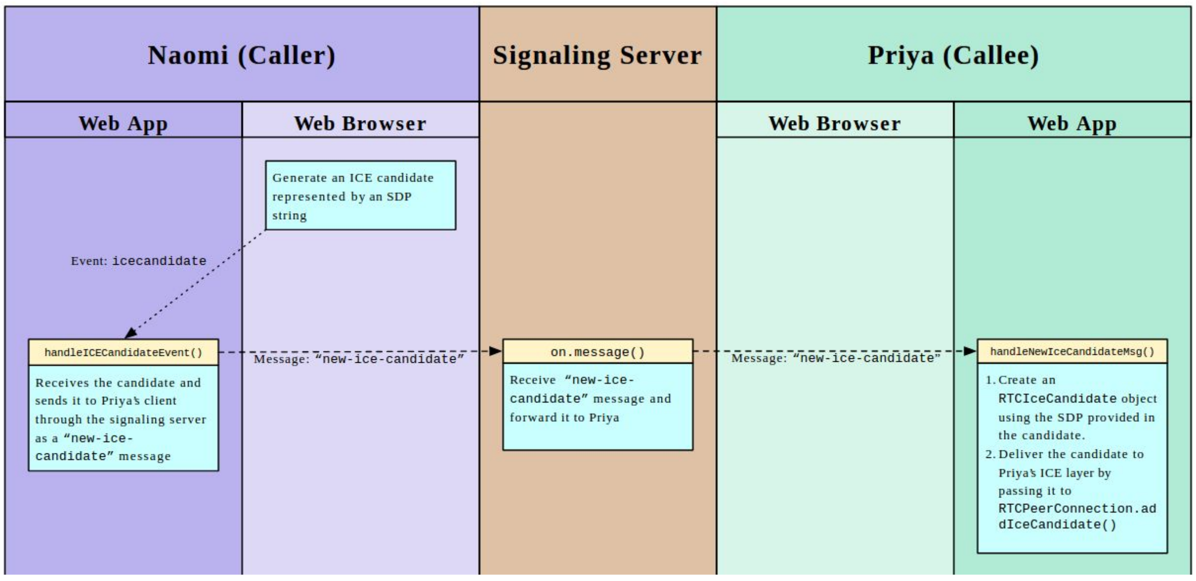
This connection will be used to send all the streams. JSON.stringifyis transforms the streams into strings to transport them as IceCandidate.

ZENNOUCHE Douria
RIVOAL Alice

1. Server creates an RTCPeerConnection object.
2. Server creates an offer (an SDP session description) with the RTCPeerConnection *createOffer()* method.
3. Server calls *setLocalDescription()* with his offer.
4. Server stringnifies the offer and uses a signaling mechanism to send it to Client.
5. Client calls *setRemoteDescription()* with Server's offer, so that her RTCPeerConnection knows about Server's setup.
6. Client calls *createAnswer()*, and the success callback for this is passed a local session description: Client's answer.
7. Client sets her answer as the local description by calling *setLocalDescription()*.
8. Client then uses the signaling mechanism to send her stringified answer back to Server.
9. Server sets Client's answer as the remote session description using *setRemoteDescription()*.

Server and Client need to exchange network information. The expression 'finding candidates' refers to the process of finding network interfaces and ports using the ICE framework.

1. Server creates an RTCPeerConnection object with an *onicecandidate* handler.
2. The handler is called when network candidates become available.
3. In the handler, Server sends stringified candidate data to Client, via their signaling channel.
4. When Client gets a candidate message from Server, she calls *addIceCandidate()*, to add the candidate to the remote peer description.


Naomi (Caller)
Signaling Server
Priya (Callee)
Web App
Web Browser
Web Browser
Web App
Generate an ICE candidate represented by an SDP string
Event: icecandidate
handleICECandidateEvent()
Receives the candidate and sends it to Priya's client through the signaling server as a "new-ice-candidate" message
Message: "new-ice-candidate"
on.message()
Receive "new-ice-candidate" message and forward it to Priya
Message: "new-ice-candidate"
handleNewIceCandidateMsg()
1. Create an RTCIceCandidate object using the SDP provided in the candidate.
2. Deliver the candidate to Priya's ICE layer by passing it to RTCPeerConnection.addIceCandidate()

About the security:

We use https protocol which means a secured connection. Because of that, a certificate is needed to make display the video on the client's side. For that you need to create a folder named ssl in folder in which the server is in. Then run the bash script available in websocket_server.

createDeviceCRT.bash needs your public IPadress as an argument. then you run the second script : createRootCA.bash

Then a certificate is generated in the folder ssl that needs to be copied in cs folder, because httpserver needs it as well. Finally both devices (client and server) must contain this certificate in their chrome. In the phone's case just download it and it will be taken care of by it system. but for the computer(server), this certificate needs to be added to the chrome's certificates. To do so follow these steps:

1. download the certificate
2. go to parametres
3. click on Afficher les paramètres avancés…
4. Then look for the https field
5. click on gerer les certificats
6. and in vos certificats click on import and import your certificate

Cylon / Arduino / Gort

To control the camera and position them correctly we use the servomoteur that is controlled by an arduino uno.

Install arduino to recognize the card arduino uno plugged into the computer. Cylon.js allows to program the behaviour of the servo motors using javascript and run it with nodejs.

Gort makes the connection between the running code with cylon and the serial port to which we plugged the arduino. The code is executed in the adequate card.
Unfortunately, cylon can't be directly ran in an html. There are few solutions to overcome this difficulty .

1. Make another connection and use socket to update the position data.
2. Use browserify to be able to run directly from an html a cylon program.

Personally, we think the second solution is better because we already have a lot of connections and the htmls that take the video and the arduino are running on the same computer.

● Results

The first part of the project is done: the two videos are displayed on the mobile phone but the latency can be very important depending on the quality of the network.

All the mobile phone's position data is sent to the server but the script that can run the arduino's program can't access it.

● How to run our program:

In order to work correctly, this software has to be run with google Chrome, because some other browsers such as firefox are not compatible with webRTC in that it does not have all the methods it needs to run correctly .

First of all download all the files and scripts needed and add a new security certificate in your browser (cf About the security).
Secondly use node in order to run the two servers that will make the connection between the two clients. The first one is called websocket.js and can be found in the folder websocket_server. websocket.js can identify the websockets requests and use a special protocol (in our case STUN protocol) to make the link between the two clients and enable them to communicate directly with each other. The second one is called httpserver.js, it can be found in cs folder. This last script allows all kind of smart devices to connect directly to a

computer using it IP public address and a Port number (5000 in our case). This is how we run our program on the phone.

Then type https://(ip address):5000/client.html on the phone's url and https://(ip address):5000/server.html on the url's device into which the cameras are plugged in. because we use a secured connection you'll need to generate a certificate to allow the transition of flux between the two apparatuses (this will be explained in the next section).

After, connect first the client (the phone) and then the server(the other device) then if everything went well a message saying connection established should be displayed in both interfaces.

• Encountered difficulties:

In the first place when we first forked the last year's project from github, Their code was incomprehensible in that we were not familiar at all with javascript and even less with websockets and webRTC functions. on top of that the comments were in chinese which made our task even harder.

Once meeting with Zilong Zhao and Hammouti Guillaume we understood the outlook and the general organisation of their project, and that helped a lot. But when it came to modify their program to make it our own things it took us ages to know what to change and which part to keep.

We wanted to simplify the architecture in that the code would run on only one machine instead of using two which were more logic to us. But soon a security matter arise, leading us to secure our software and generate certificates.

WebRTC is a very dynamic technologie so it gets updates frequently which obliged us to review all of the methods used in our software because most of them were depressed.

We could only work on one machine because only one of our machines supported the 16 th version of ubuntu which was compatible with node. while the other machine supported 14 th version of ubuntu and so the node version it has was absolutely not compatible with the used technologies, as a result it was impossible to make local tests on it.

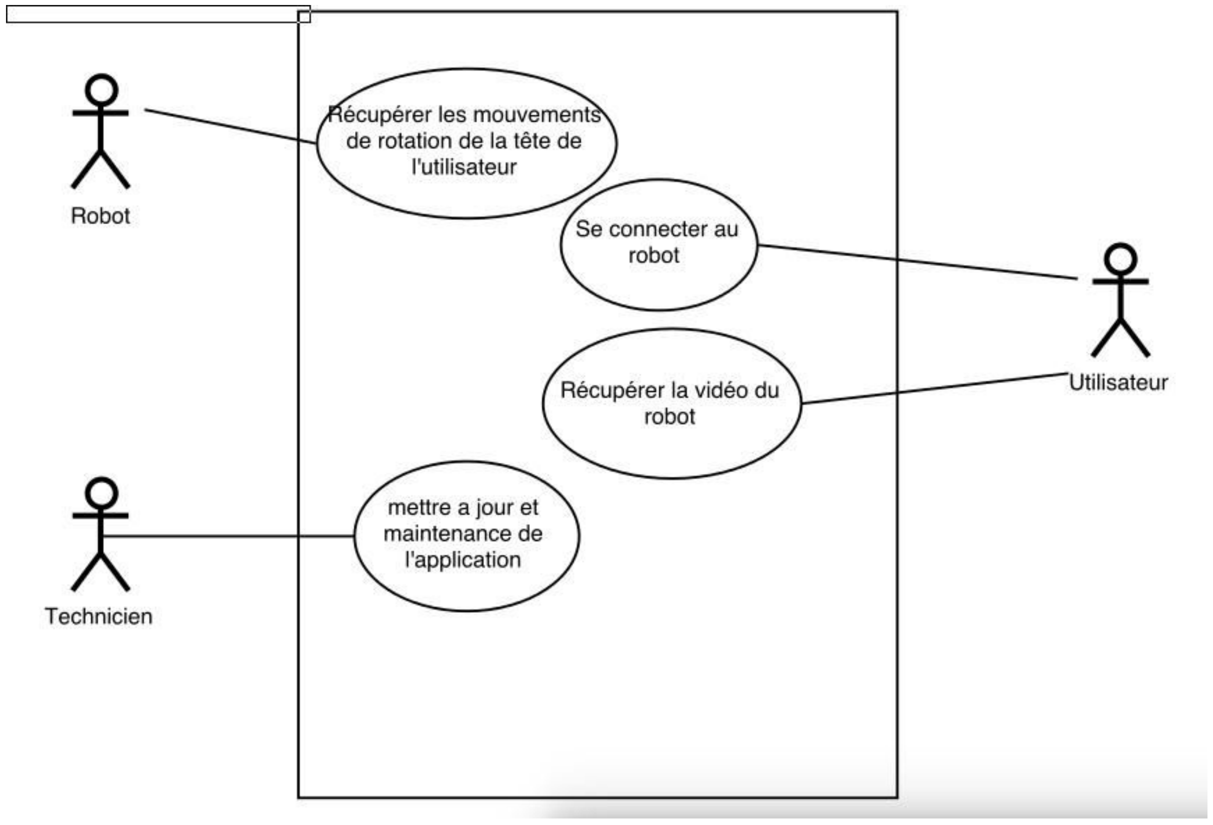
• Conclusion:

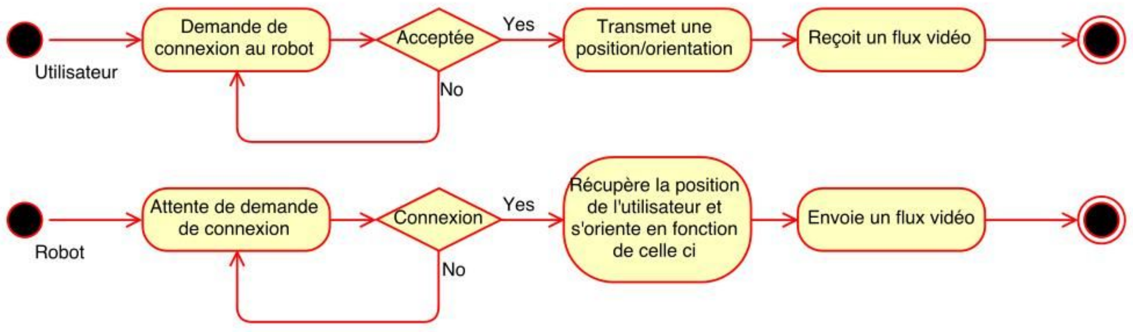
This project was very interesting in that we discovered many technologies like WebRTC and RTCPeerConnection which are still getting developed.
The application of this project can be very various. It was designed in the first place to help disabled children that stays in the hospital 24/7 to attend school classes and visit museums as any other kid of their age.

ZENNOUCHE Douria
RIVOAL Alice

- **Annex:**

<u>UML</u>


Robot
Récupérer les mouvements de rotation de la tête de l'utilisateur
Se connecter au robot
Utilisateur
Récupérer la vidéo du robot
Technicien
mettre a jour et maintenance de l'application


Utilisateur
Demande de connexion au robot
Acceptée
Yes
No
Transmet une position/orientation
Reçoit un flux vidéo
Robot
Attente de demande de connexion
Connexion
Yes
No
Récupère la position de l'utilisateur et s'oriente en fonction de celle ci
Envoie un flux vidéo