

UNIVERSITÉ  
**CÔTE D'AZUR**

# **PROJET TER COURS À DISTANCE**

**M1 MIAGE**

**ANNÉE 2020 – 2021**

# SOMMAIRE

Introduction

Peer to Peer

Api et Bibliothèques utilisées

Conclusion

# INTRODUCTION



# TER

Le TER (Travail d'Étude et de Recherche) est un stage sous la direction d'un encadrant universitaire ou industriel qui s'effectue par groupe de 2 à 4 étudiants (ingénierie) ou seul (recherche). Il sanctionne la fin du Master 1 et s'étend sur environ 3-4 mois (1 jour par semaine)

Une plateforme de cours "live" à distance qui repose sur une approche la moins centralisée possible



Ce stage, encadré par M.Menez est réalisé par 3 étudiants :  
Benabdelkrim Mohamed, Elarays Said et M'hamed Lotfi



PEER TO PEER

# DÉFINITION

- Peer to Peer en français « d'égal à égal » ou « pair à pair », c'est un modèle de réseau informatique proche du modèle client-serveur mais où chaque client est aussi un serveur.
- Système d'échange direct de ressources entre machines connectée en réseau
- Se distingue fondamentalement de l'architecture client / serveur

# AVANTAGES :

- **La répartition des charges:**

Les P2P ne posent pas le problème de congestion des réseaux qui peuvent se produire autour d'un serveur central devant répondre à de très nombreuses demandes

- **Capacité de stockage:**

Chaque nœud ne possède qu'une infime partie des données du réseau, qu'il partage avec les autres ordinateurs.

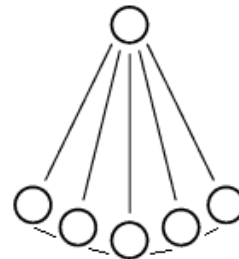
- **Extensibilité:**

Le P2P est très facile d'utilisation, c'est donc simple de rajouter de nouveaux pairs.

# LE PEER TO PEER : ARCHITECTURES

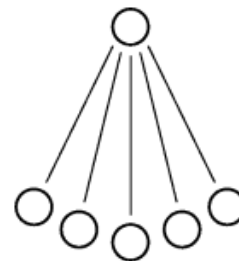
## Architecture centralisée :

- Mise en relation à partir du serveur
- Puis liaison directe entre les clients



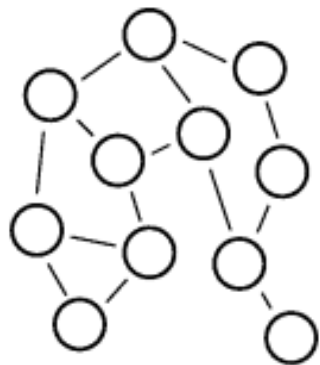
## Architecture client / serveur :

- Ce n'est PAS un réseau Peer to Peer
- Ici tout passe par le serveur
- Aucune liaison directe entre les clients





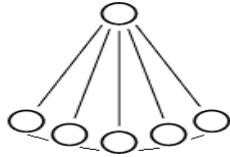
# LE PEER TO PEER : ARCHITECTURES



## Architecture décentralisée :

- Égalité entre tous les utilisateurs
- Liaisons établies de proche en proche
- Requêtes transférées et relayées
- Fichiers transférés directement du demandeur au donneur
- Réseau en perpétuelle mutation

# LE PEER TO PEER : COMPARATIF



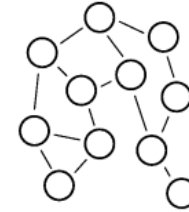
## ► P2P centralisé

### ☐ **Avantage :**

- Efficacité des recherches, facilité d'utilisation

### ☐ **Inconvénients :**

- Complètement tributaire du serveur central
- Aucun anonymat n'est garanti
- Possibilité pour le serveur central de créer des fichiers clients et des profils d'utilisateurs pour les revendre à des compagnies spécialisées



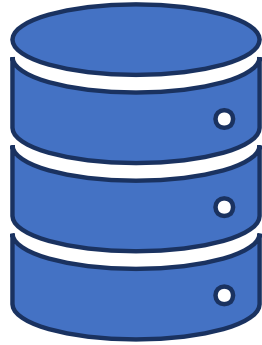
## ► P2P décentralisé

### ☐ **Avantages :**

- Grande souplesse, grande robustesse
- Anonymat (relatif) assuré car il n'y a pas de serveur qui stocke des données sur les utilisateurs

### ☐ **Inconvénients :**

- Problèmes au niveau de la bande passante
- Anonymat => risques de piratage et d'échange de données illégales



# API ET BIBLIOTHÈQUES

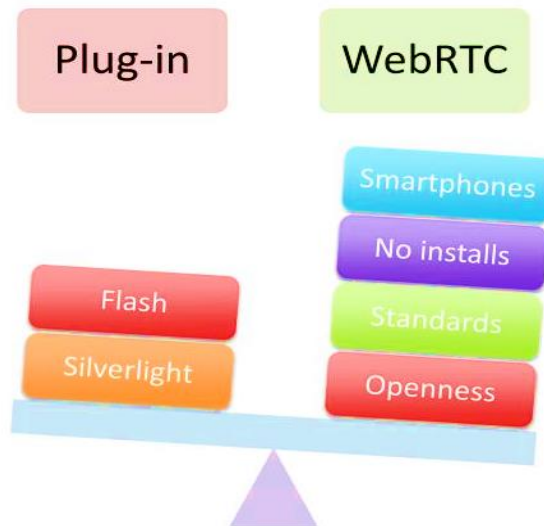


## DÉFINITION

WebRTC est une collection de standards, de protocoles et d'API JavaScript permettant le partage navigateur à navigateur de contenu vidéo, audio ou encore de données.

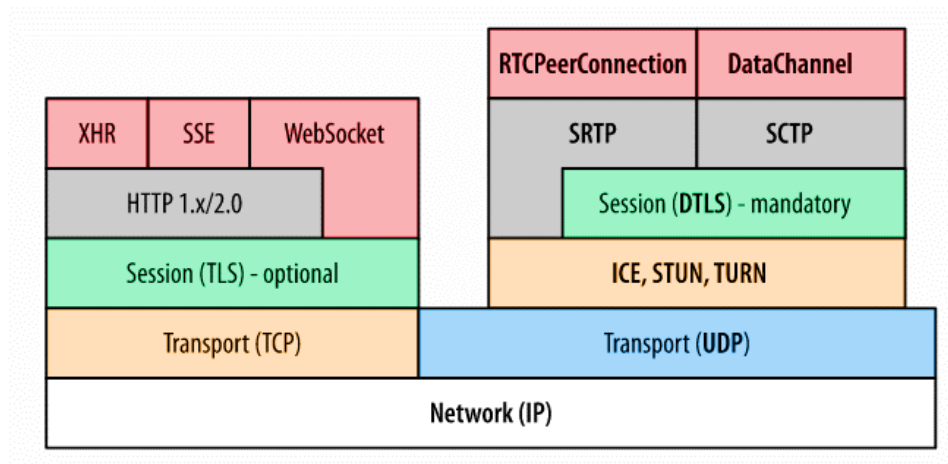
WebRTC pour Web Real-Time Communication permet, comme son nom l'indique de rendre possible les communications multimédias en temps réel dans un contexte Web en évacuant la nécessité d'une installation préalable.

## Why WebRTC



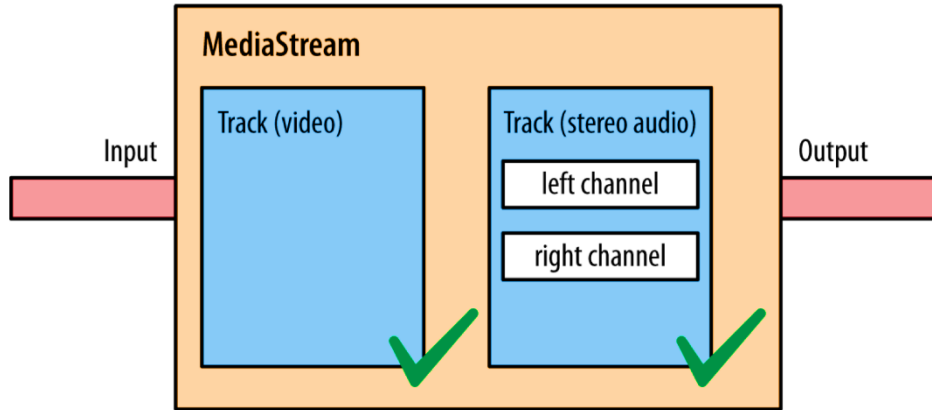
# TRANSPORT ET PROTOCOLES

WebRTC se distingue d'abord par le choix du protocole de transport qui le soutient. En effet, WebRTC a déterminé UDP comme couche de transport qui se prête particulièrement bien au contexte temps-réel où la latence devient critique. Si UDP n'offre aucune garantie par rapport à TCP (qui garantit l'arrivée des paquets, mais aussi l'ordre de ceux-ci), il se prête beaucoup mieux aux flux multimédias en temps réel qui peuvent généralement supporter la perte de certaines données (codecs audio et vidéo).



# API : MEDIASTREAM

L'objet `MediaStream` permet des fonctionnalités (accéder au contenu audio/vidéo capturé par l'appareil hôte) de téléconférence simplifié par la méthode `getUserMedia`, mais encapsule aussi de puissants moteurs de traitement audio/vidéo internes permettant d'en améliorer la qualité (encodage, décodage, synchronisation, gestion des fluctuations de la bande passante, dissimulation des paquets perdus, réduction des bruits dans la bande audio, amélioration de l'image dans le flux vidéo, etc.) Notons que la majorité des navigateurs vont forcer l'utilisation de HTTPS dans la page avant de permettre l'utilisation de `getUserMedia`.

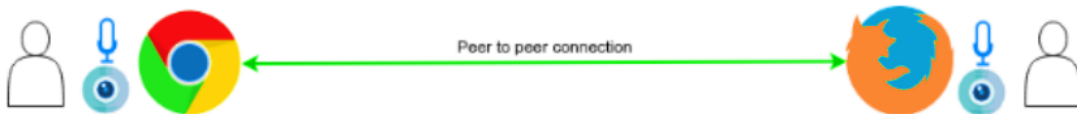


# RTCPeerConnection

L'API RTCPeerConnection est responsable de la communication des contenus audio et vidéo et devient donc fondamentale dans le cas qui nous intéresse.

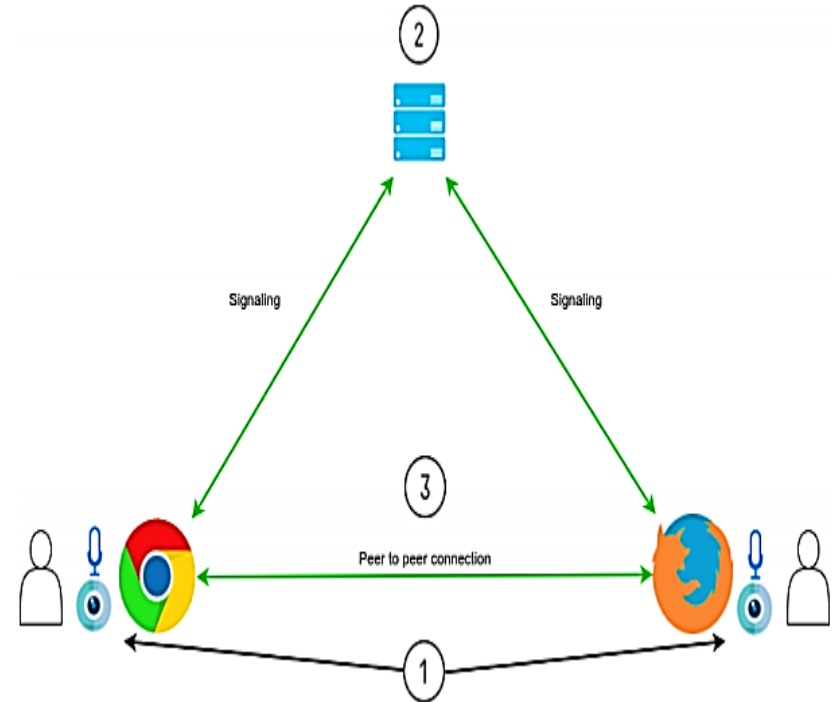
C'est par cette interface que deux pairs vont apprendre à communiquer ensemble et vont établir un tunnel de partage de contenu multimédia (par SRTP).

RTCPeerConnection s'occupe des étapes d'initialisation de la connexion entre les pairs, la gestion de la session, l'envoi des flux multimédias et l'état de la communication.



# EXEMPLE D'INITIALISATION DE LA COMMUNICATION

1. Le navigateur obtient l'accès au matériel d'enregistrement comme la caméra ou le microphone.
2. Chaque participant s'identifie et échange l'information nécessaire à l'établissement d'une connexion avec les autres participants.
3. Chaque participant peut maintenant partager directement son contenu multimédia avec les autres participants et la communication peut alors commencer.



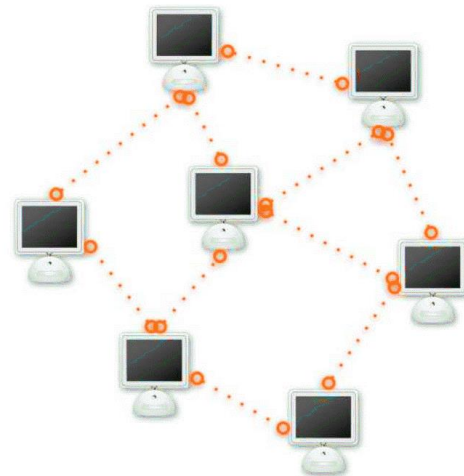


# LA NATURE PLURIPARTITE D'UNE VIDÉOCONFÉRENCE

Jusqu'à maintenant nous avons vu comment connecter un pair à un autre, mais qu'arrive-t-il lorsque que nous voulons connecter plus de participants comme dans le cas d'une vidéoconférence typique? Il existe en fait trois approches pour déployer une application de vidéoconférence pluripartite: Mesh, MCU et SFU.

- **L'approche Mesh**

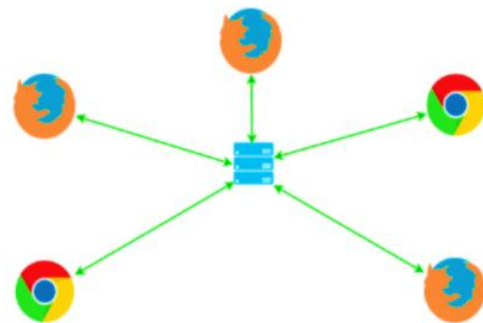
Mesh est en quelque sorte l'approche naïve, à savoir que chaque participant va envoyer son média à chacun des autres participants. Cette approche échoue rapidement au delà de 3 participants en termes de bande passante et de nombre de connexions. Un participant devra téléverser son média à chacun des autres participants et télécharger le média de chacun de ceux-ci.



# LA NATURE PLURIPARTITE D'UNE VIDÉOCONFÉRENCE

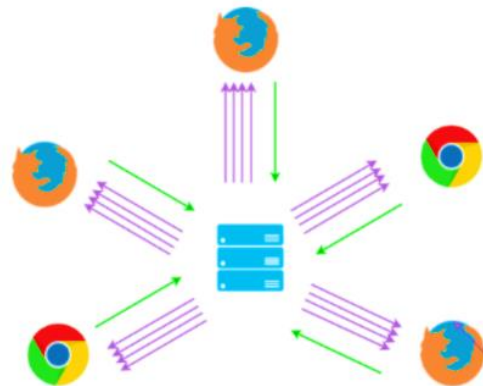
## ■ Le modèle MCU

différents participants vont tous communiquer avec une unité centrale (un serveur) qui va s'occuper de composer les flux en un seul flux audio-vidéo. C'est définitivement la meilleure approche en terme de charge sur le réseau et de nombre de connexions. Cette approche limite à une connexion entrante et sortante entre le client et le serveur, peu importe le nombre de participants.



## ■ L'approche SFU

SFU (pour Selective Forwarding Unit) propose que chaque participant envoie son média à un serveur central qui s'occupe de le rediriger (*router*) aux autres participants si ceux-ci en font la demande. On limite ainsi à un le nombre de téléversement par rapport à Mesh ( $n$  participants), mais on garde le nombre de téléchargements de flux égal au nombre de participants.





**CONCLUSION**

MERCI !

