



Sommaire

- Historique de la visioconférence sur internet
- WebRTC qu'est ce que c'est ?
- Protocoles et API pour communiquer en temps réel
 - API Javascript et html
 - WebSockets
 - DTLS
 - Media
 - Traversée de NAT
- WebRTC et RTC
- Retour sur déploiements
- Conclusion
- Demo ?

Historique

La visioconférence existe en principe depuis 1958 (exposition universelle).

1995 : 1ère application de la visioconférence entre le Cap et San Francisco

1995 : Etablissement des bases de la VOIP par Intel, Microsoft et RADVISION

2011 : 1ère ébauche du WebRTC

Outils de Visioconférence

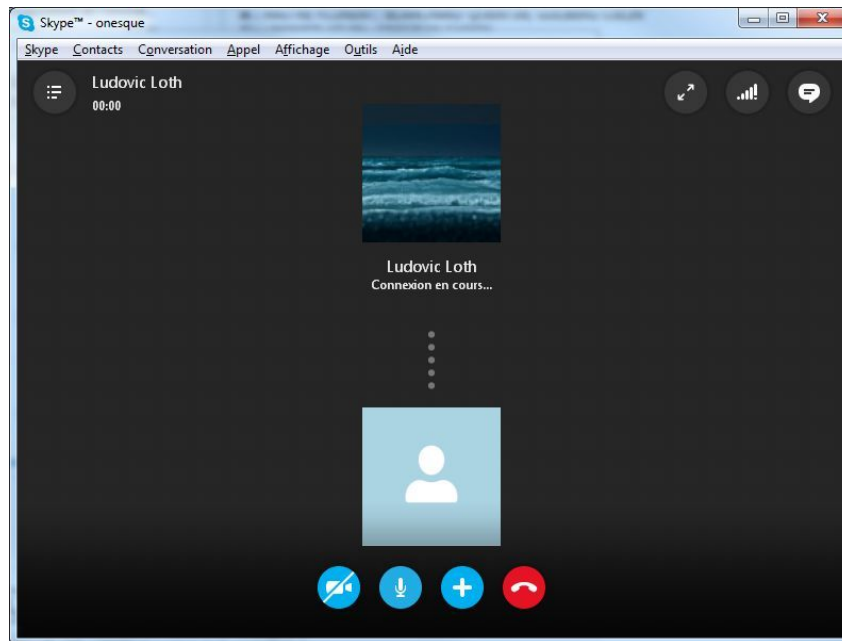
Historiquement, la visioconférence se faisait surtout grâce à des clients lourds :

-Solution CISCO



Outils de Visioconférence

-Solution Microsoft (MSN puis Skype)



Outils de Visioconférence

-A travers un navigateur avec des solutions lourdes (Flash, Java)



WebRTC Description

WebRTC : Web RealTime Communication

API Javascript = Interface de développement

En cours de développement au sein le W3C et de l'IETF

Pas de protocole de signalisation

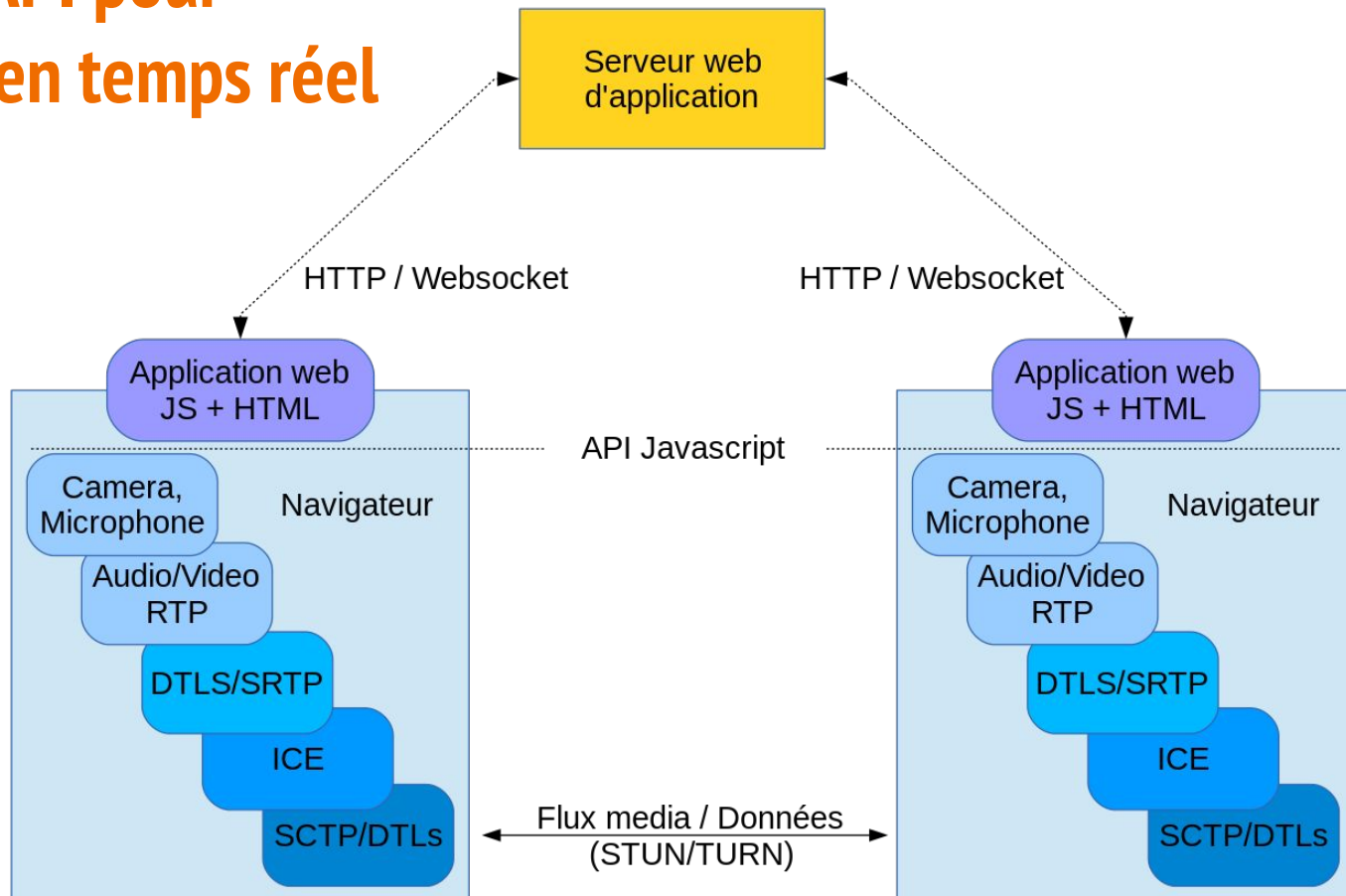
Soutenu par de grands acteurs du WEB et de l'informatique :

Standardisation en cours

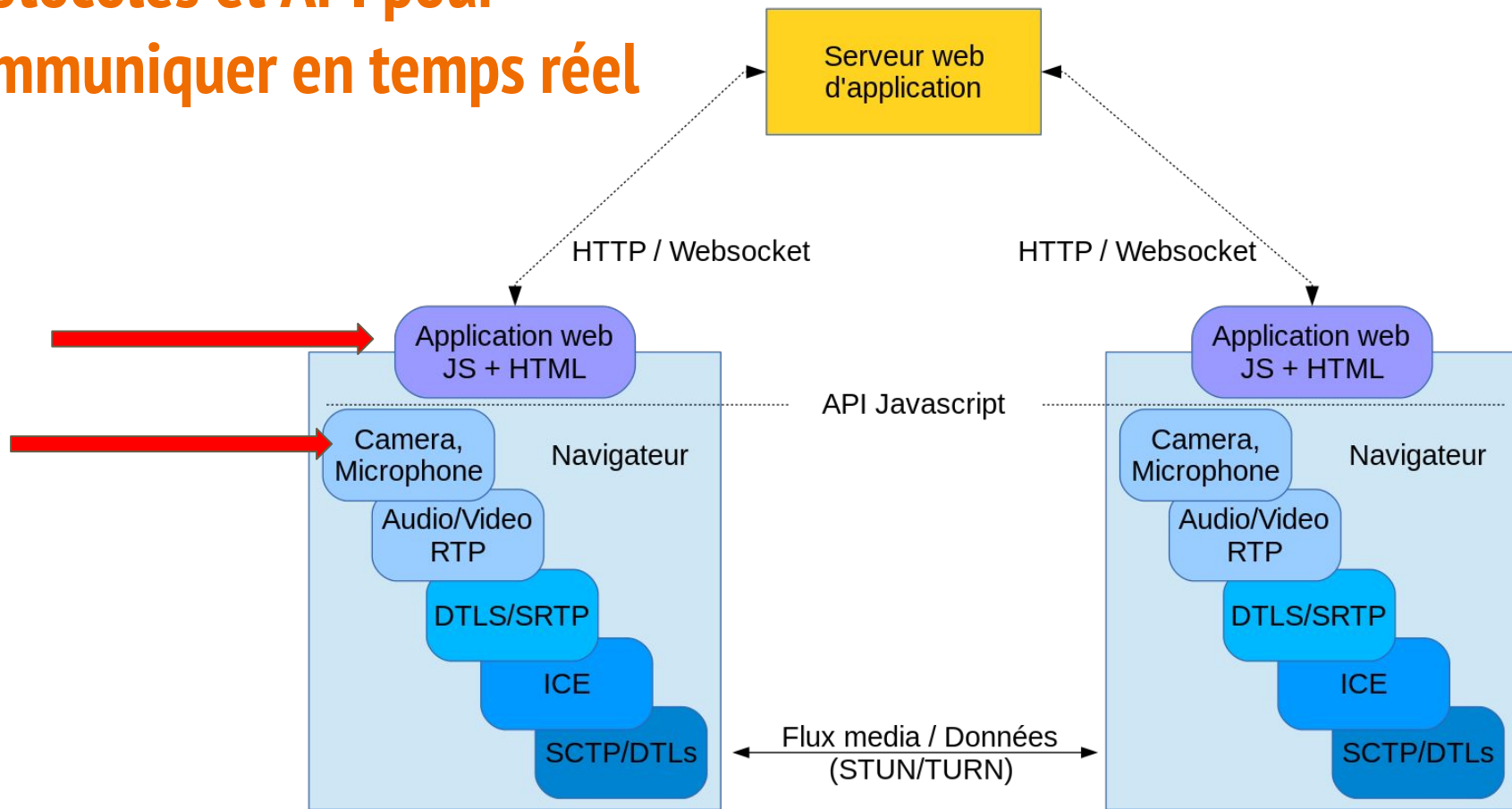
Supporté par les navigateurs récents

Google
Ericsson
Cisco
Plantronics
Microsoft
Mozilla Foundation
Apple
Orange
AT&T
Samsung
Avaya

Protocoles et API pour communiquer en temps réel



Protocoles et API pour communiquer en temps réel

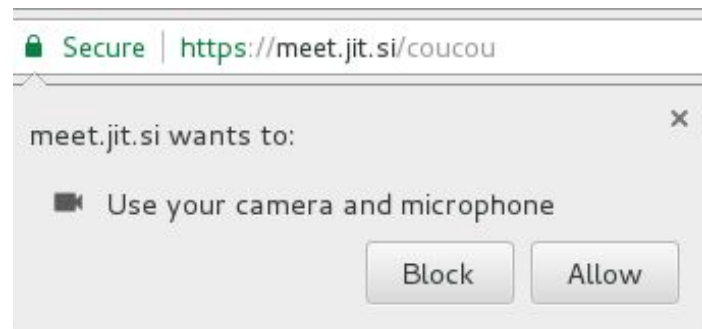
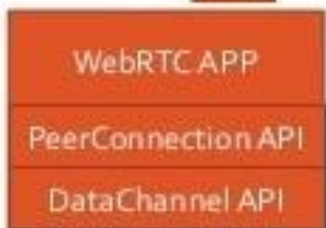


API implémenté dans les navigateurs

- API en javascript
- "Indépendant" du navigateur utilisé
- Grand nombre de librairies existantes

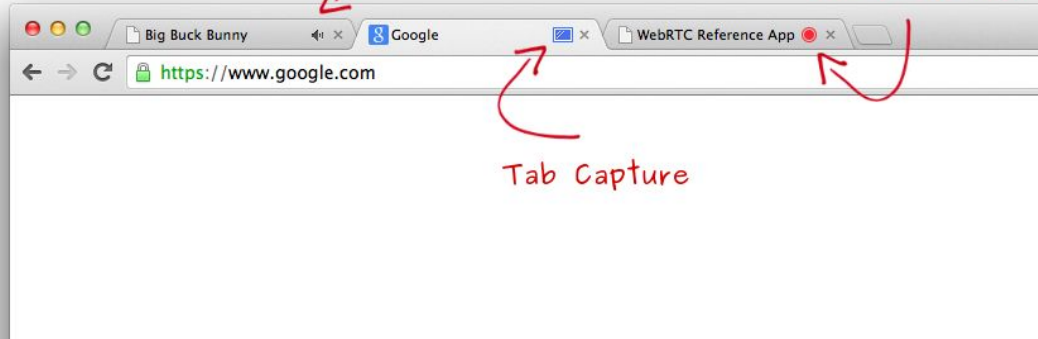


getUserMedia API

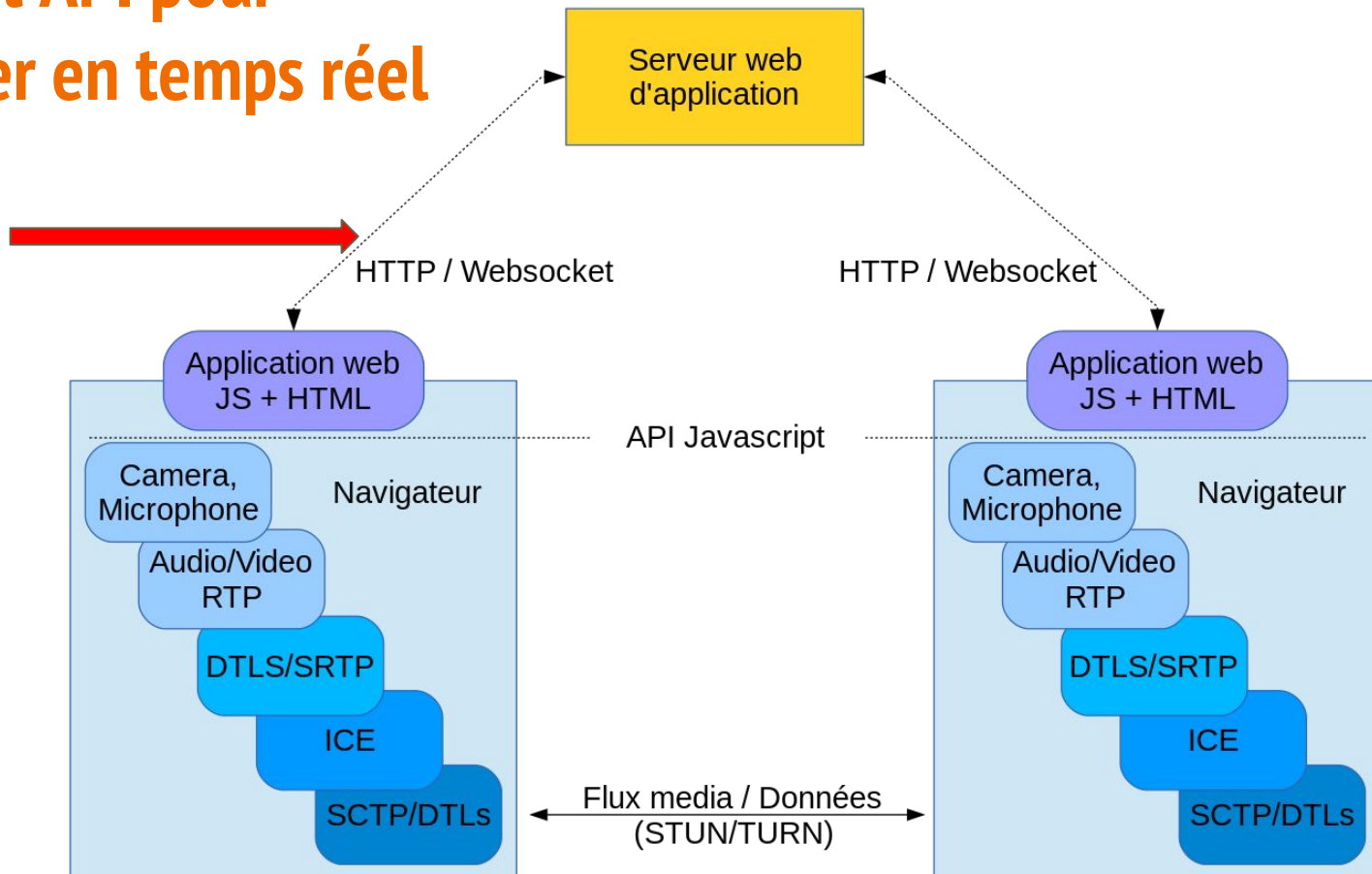


Audio Playback

Tab Recording



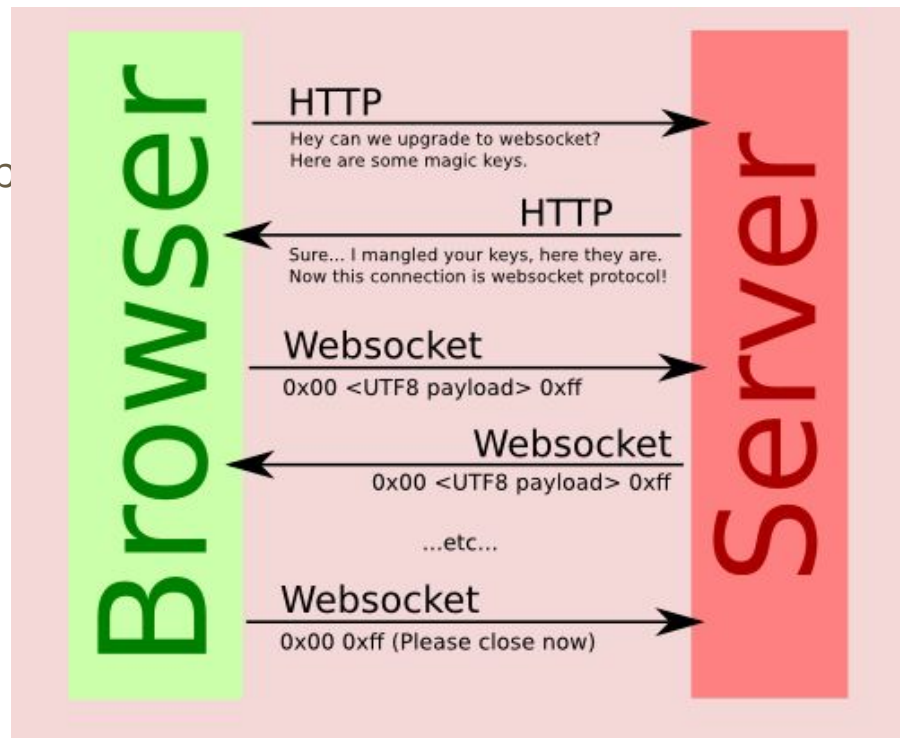
Protocoles et API pour communiquer en temps réel



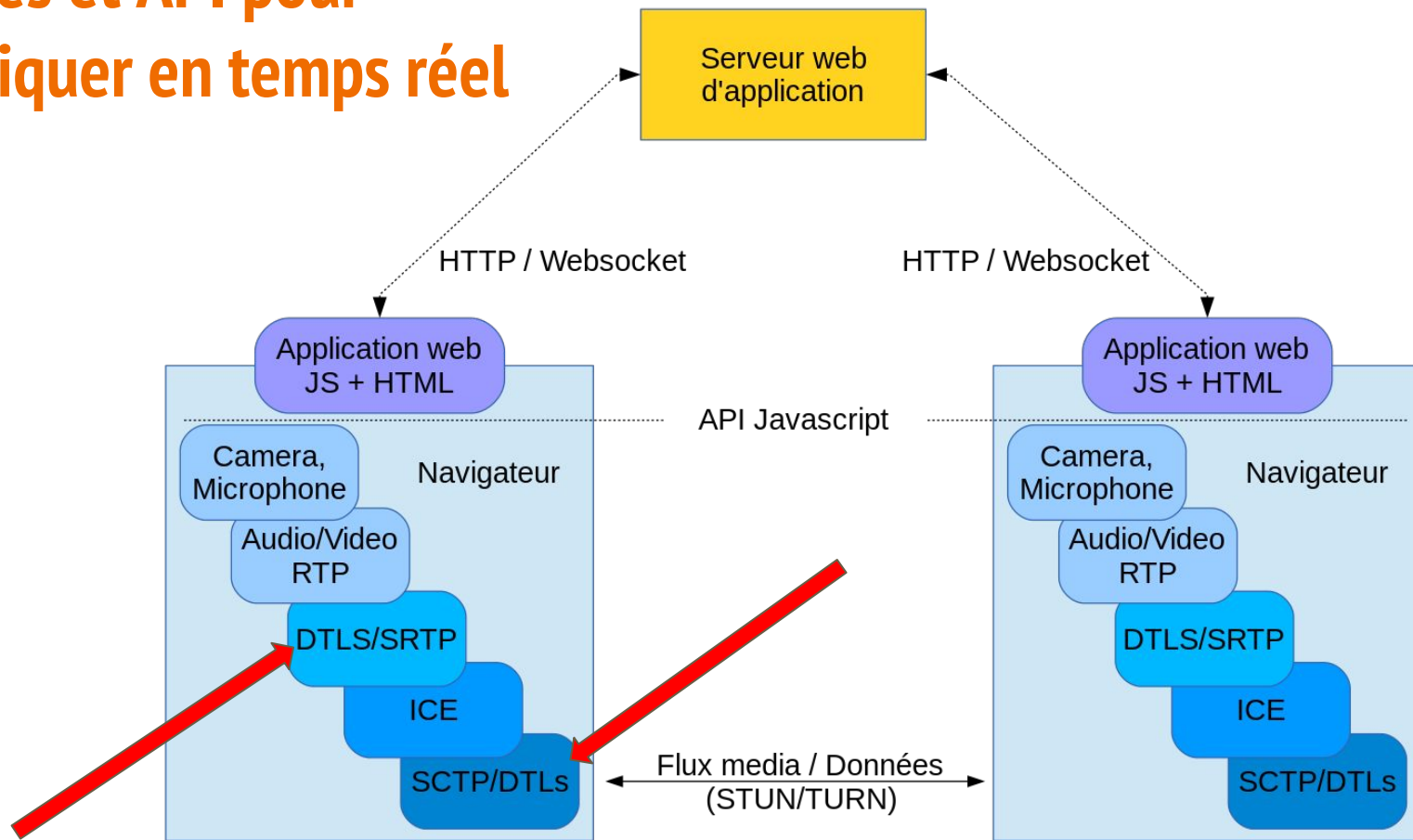
Transport de la signalisation : Websockets

Permet communications bidirectionnelles en http entre un client et le serveur

https obligatoire pour les flux de données depuis décembre 2015

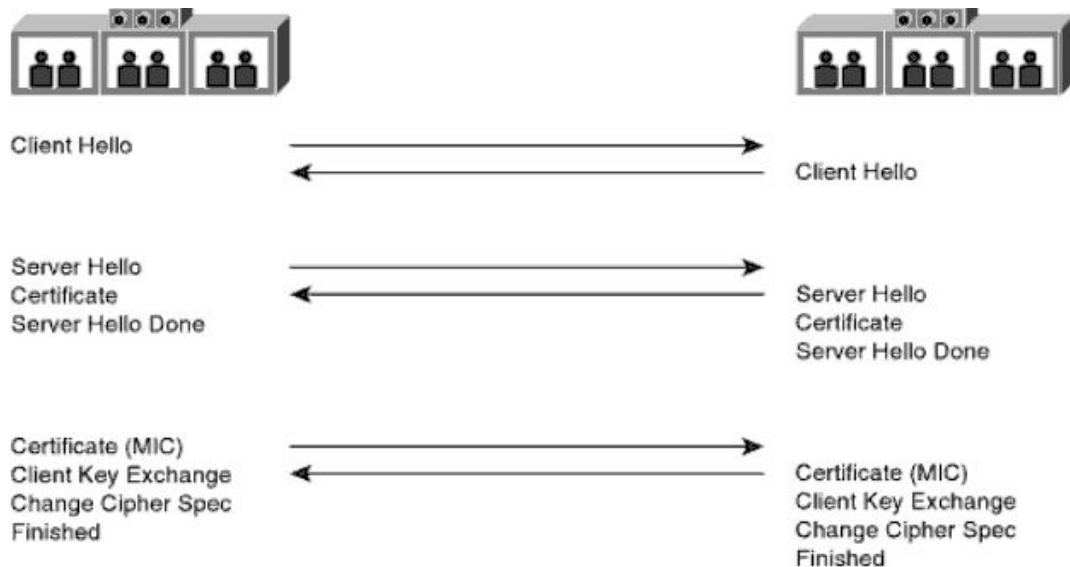


Protocoles et API pour communiquer en temps réel

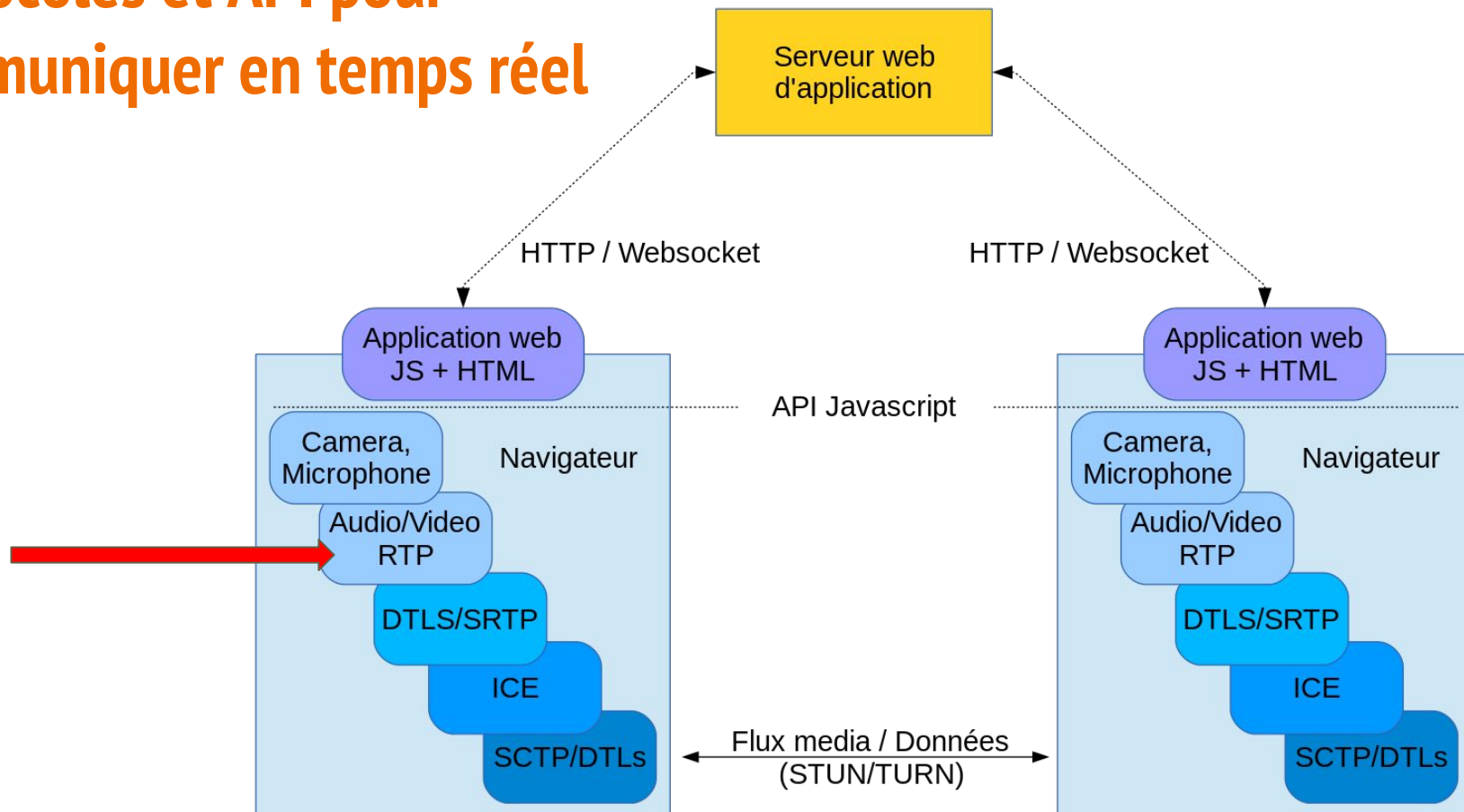


Chiffrement DTLS

Chaque endpoint peut se comporter comme client ou serveur



Protocoles et API pour communiquer en temps réel

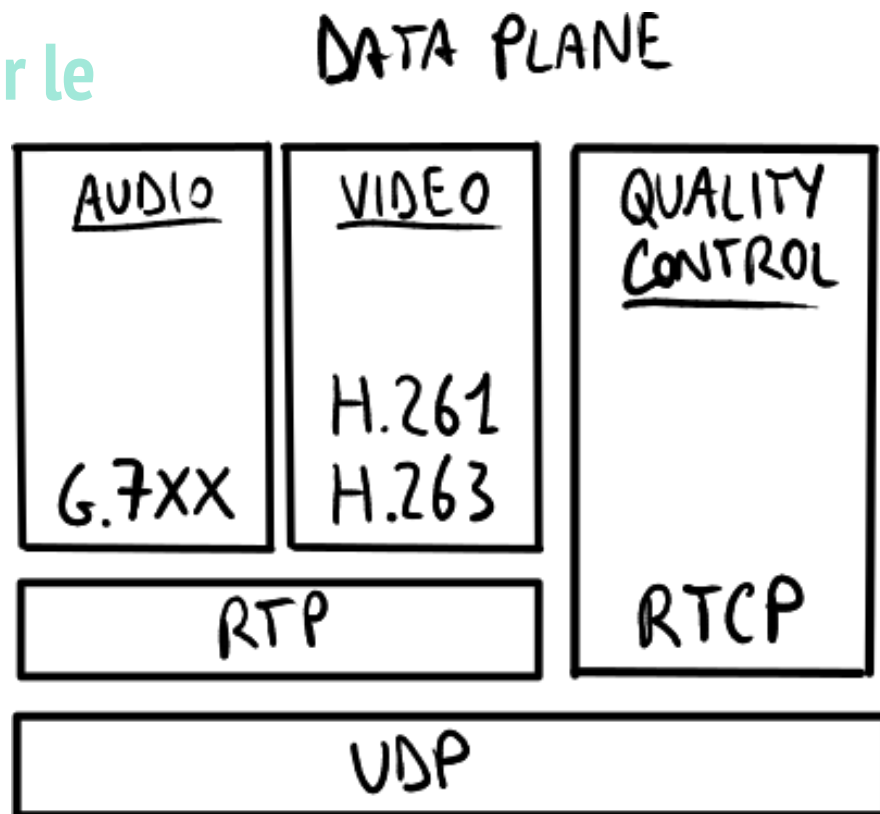


Pile de protocole standard pour le media (par rapport à la VOIP traditionnelle)

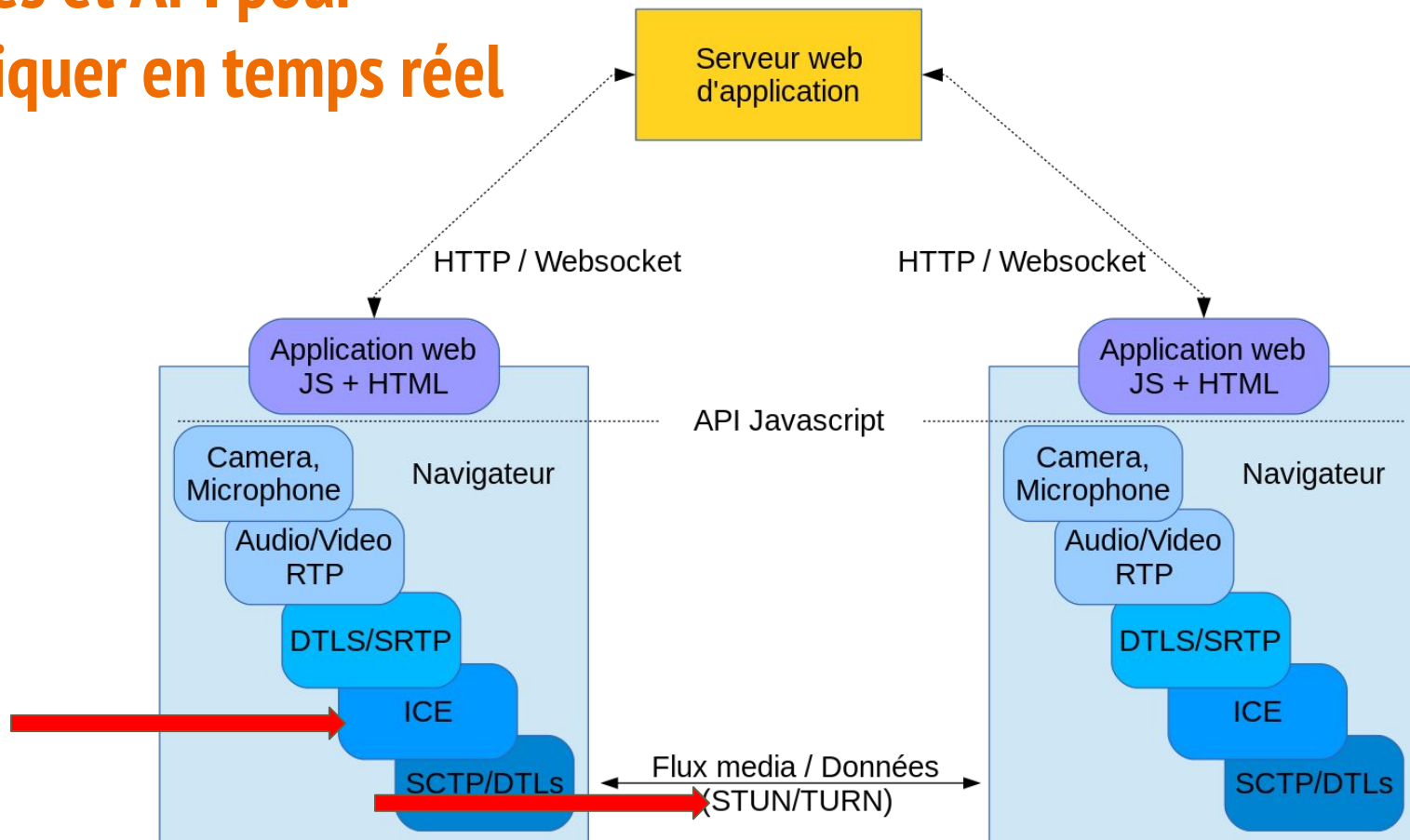
Comme vu avant chiffré

Audio : G.711, G.722, iLBC, OPUS

Vidéo : iSAC VP8 H264



Protocoles et API pour communiquer en temps réel



Traversée de NAT

Pourquoi : Internet / IPV4 / firewall

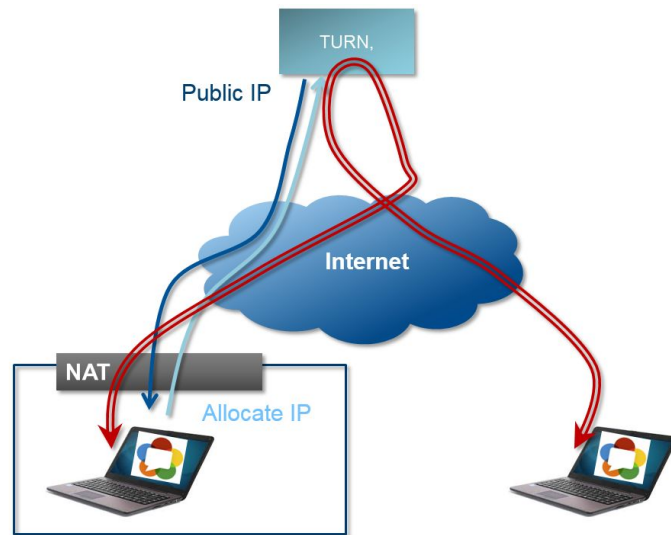
STUN → quelle est mon IP de ton point de vue
(Session Traversal Utilities for NAT)

TURN → peut relayer le flux media
(Traversal Using Relays around NAT)

STUN + TURN = ICE

Interactive Connectivity Establishment

Des abus de langage entre STUN et ICE



WebRTC vers RTC via SIP sur websocket

Session Initiation Protocol

Réseau Téléphonique Commuté



XIVO ACCESS

XIVO CONNECT

DÉCOUVRIR XIVO

Démo WEBRTC

Démo WEBRTC

Home / Démo WEBRTC

Bienvenue sur notre page démo

La technologie WEBRTC est intégrée nativement dans la solution xivo.solutions.

Nous vous proposons de la découvrir directement depuis notre site xivo.solutions et votre navigateur.

En cliquant sur le bouton "Dial" ci-dessous, vous pourrez lancer un appel type "RTC", sans aucune installation de plugin ni configuration.

Chargement de la page web



Aucun appel en cours

Enregistrement en cours

Etablissement de la websocket

▼ General

Request URL: wss://www.xivo.solutions/ws

Request Method: GET

Status Code: 🟢 101 Switching Protocols

▼ Response Headers [view source](#)

Connection: Upgrade

Sec-WebSocket-Accept: 8ho2QNOLtGVOLbBiATqB7Bh8bck=

Sec-WebSocket-Protocol: sip

Upgrade: WebSocket

▼ Request Headers [view source](#)

Accept-Encoding: gzip, deflate, sdch, br

Accept-Language: en-US,en;q=0.8

Cache-Control: no-cache

Connection: Upgrade

Cookie: pll_language=fr; PHPSESSID=qrtbsc3ldomec9ge8qpm7saj2258984.7.10.1488832463; __utmc=35058984; __utmz=35058984.1488

Host: www.xivo.solutions

Origin: https://www.xivo.solutions

Pragma: no-cache

Sec-WebSocket-Extensions: permessage-deflate; client_max_window

Sec-WebSocket-Key: IzVbF4jIimDbohWl9T9v4Q==

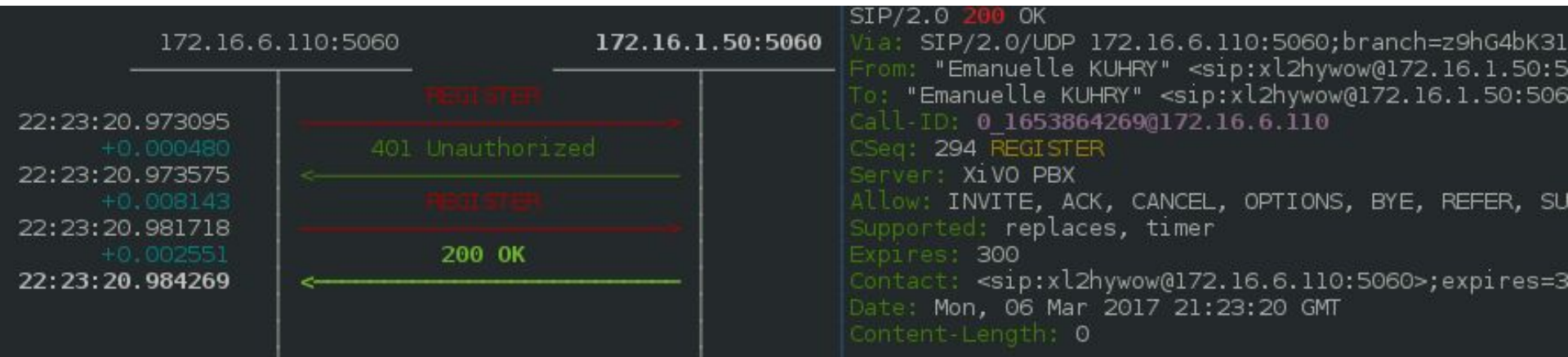
Sec-WebSocket-Protocol: sip

Sec-WebSocket-Version: 13

Upgrade: websocket

User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36

Enregistrement de la terminaison



Navigateur

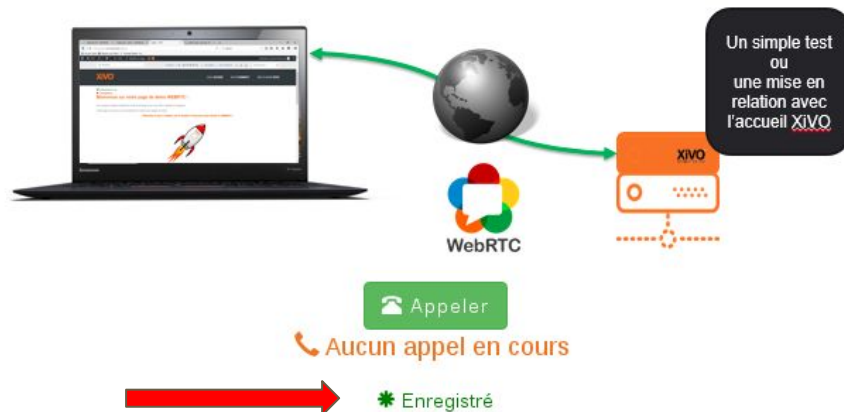
Serveur

Bienvenue sur notre page démo

La technologie WEBRTC est intégrée nativement dans la solution xivo.solutions.

Nous vous proposons de la découvrir directement depuis notre site xivo.solutions et votre navigateur.

En cliquant sur le bouton "Dial" ci-dessous, vous pourrez lancer un appel type "RTC", sans aucune installation de plugin ni configuration.



Bienvenue sur notre page démo

La technologie WEBRTC est intégrée nativement dans la solution xivo.solutions.

Nous vous proposons de la découvrir directement depuis notre site xivo.solutions et votre navigateur.

En cliquant sur le bouton "Dial" ci-dessous, vous pourrez lancer un appel type "RTC", sans aucune installation de plugin ni configuration.



INVITE → Candidats ICE

Réponse SDP → candidats ICE

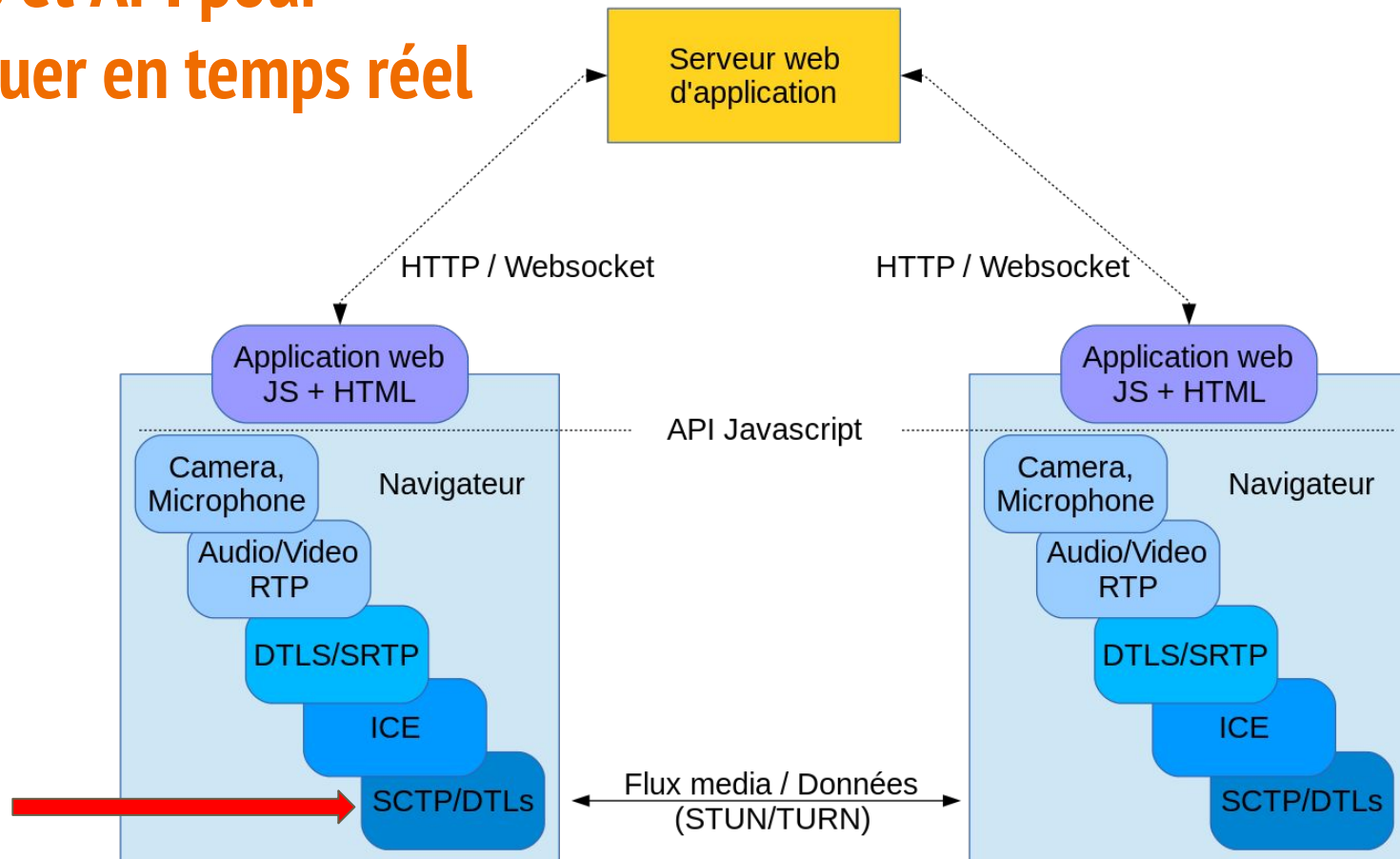


En communication
On envoie le Flux
Media

```
SIP/2.0 200 OK
Via: SIP/2.0/WS df7jaL23ls0d.invalid;branch=z9hG4bKT28PDG0I9qf6BCqNAwQpmldaL6b1Hqyy;received=x.x.x.x;rport=
From: "1060"<sip:1060@y.y.y.y>;tag=rf4ZEllaSI3suiZv9IqW
To: <sip:1061@y.y.y.y>;tag=as2641dea4
Call-ID: a18a9df3-8396-0692-bc8b-578c2715b651
CSeq: 53756 INVITE
Server: Asterisk PBX 11.7.0
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY, INFO, PUBLISH
Supported: replaces, timer
Contact: <sip:1061@y.y.y.y:5060;transport=WS>
Content-Type: application/sdp
Content-Length: 1099

v=0
o=root 1383765036 1383765036 IN IP4 y.y.y.y
s=Asterisk PBX 11.7.0
c=IN IP4 y.y.y.y
t=0 0
m=audio 18268 RTP/SAVPF 0 8 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=silenceSupp:off - - -
a=ptime:20
a=ice-ufraq:0192884b4f19f1773e601d8746d019a9
a=ice-pwd:4ecbdf147f46f2151231e79c1d050f14
a=candidate:Ha2f2df83 1 UDP 2130706431 y.y.y.y 18268 typ host
a=candidate:Hab0a297 1 UDP 2130706431 10.176.162.151 18268 typ host
a=candidate:Hc0a8a804 1 UDP 2130706431 192.168.168.4 18268 typ host
a=candidate:Sa2f2df83 1 UDP 1694498815 y.y.y.y 18268 typ srflx
a=candidate:Ha2f2df83 2 UDP 2130706430 y.y.y.y 18269 typ host
a=candidate:Hab0a297 2 UDP 2130706430 10.176.162.151 18269 typ host
a=candidate:Hc0a8a804 2 UDP 2130706430 192.168.168.4 18269 typ host
a=candidate:Sa2f2df83 2 UDP 1694498814 y.y.y.y 18269 typ srflx
a=connection:new
a=setup:active
a=fingerprint:SHA-256 4B:D0:F1:39:36:B7:B0:91:57:87:9F:1F:6E:CF:C4:37:D2:BB:C7:72:F9:DC:A4:CB:50:64:9E:0D:F
a=sendrecv
```

Protocoles et API pour communiquer en temps réel



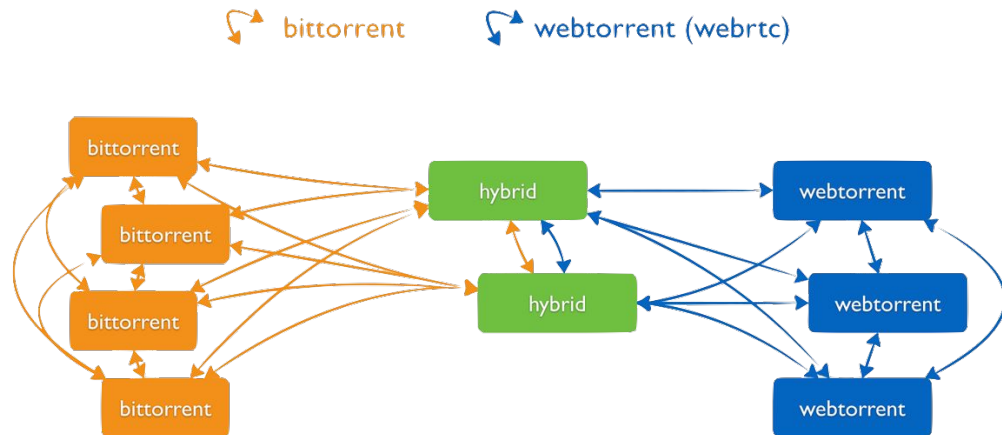
Echange de fichier / chat

Pas en TCP mais en SCTP (nouveau protocole années 2000)

Permet l'envoi de stream

Est pensé pour envoyer des données directement d'un navigateur à un autre

Application détournée : webtorrent



Retour sur Déploiement

Sujet complexe nécessite des compétences dans plusieurs domaines (réseau + téléphonie IP / web)

- Maîtriser le parc de navigateurs internet / gestion de QOS ?
- Évolue rapidement (encore un draft)

Problèmes rencontrés avec asterisk (chan_sip et res_rtp):

- Pas de réponse aux DTLS hello avant d'envoyer le flux RTP :
<https://issues.asterisk.org/jira/browse/ASTERISK-25614>
- Implémentation de RTCP-MUX : <https://webrtcstandards.info/rtcp-mux/>
<https://issues.asterisk.org/jira/browse/ASTERISK-26732>
- Trop d'interfaces réseau → latence ICE

Conclusion

Court un exposé de 10 min sur le webRTC

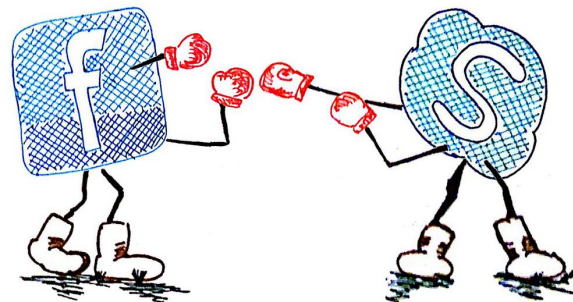
Aperçu car pas abordé exactement tous les protocoles utilisés

Projet innovant et dynamique

Processus de décision public et ouvert

Pensé pour être chiffré + décentralisé

Solution la plus pertinente pour faire du RTC à l'heure actuelle sur internet



Démo

<https://www.xivo.solutions/webrtc/>

https://meet.jit.si/coucou_m2gpi

Questions ?

