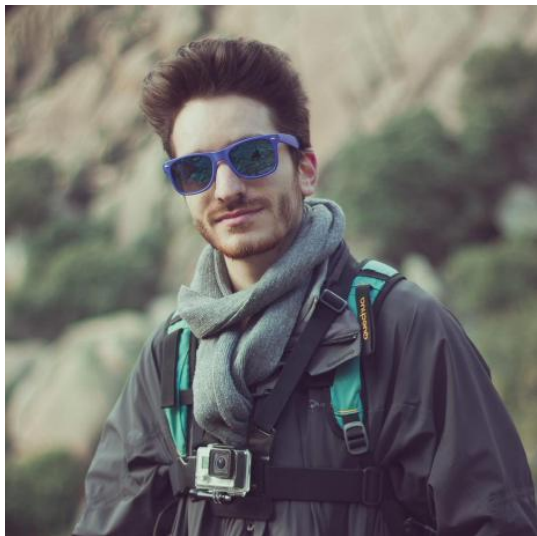




**OpenVidu**, una plataforma  
open source para añadir  
videoconferencia a tu  
aplicación web y móvil



# ¿Este quién es?



@fuentepab



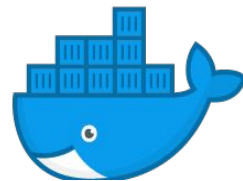
@pabloFuente



pablofuenteperez@gmail.com

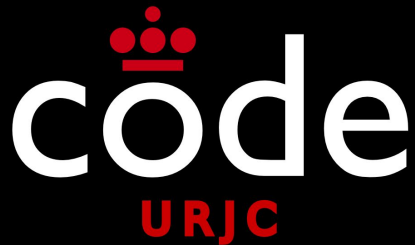


## Developer



TypeScript





## Consultancy / Training

Cloud Computing

Web Technologies

Extreme Programming

Testing / Git / Jenkins

Software Architecture

Concurrent Programming



# openVidu

Easy videoconference in your web or mobile app



## VoD

Bajo demanda:  
sin factor temporal



## Live streaming

500ms ~ 10 s



## Real time

< 500 ms



Web  RTC

¿Qué es



**WebRTC ?**

# WebRTC: Comunicación en tiempo real para la web



- Permite comunicar un browser directamente con otro, ***peer-to-peer***, sin que los datos tengan que pasar por el servidor (siempre que sea posible)
- Usa el **framework ICE** para **traspasar los NATs** y permitir una comunicación directa entre navegadores
- La baja latencia es ideal para transmitir **audio y vídeo**



# Can I use WebRTC?



También como librerías nativas para apps

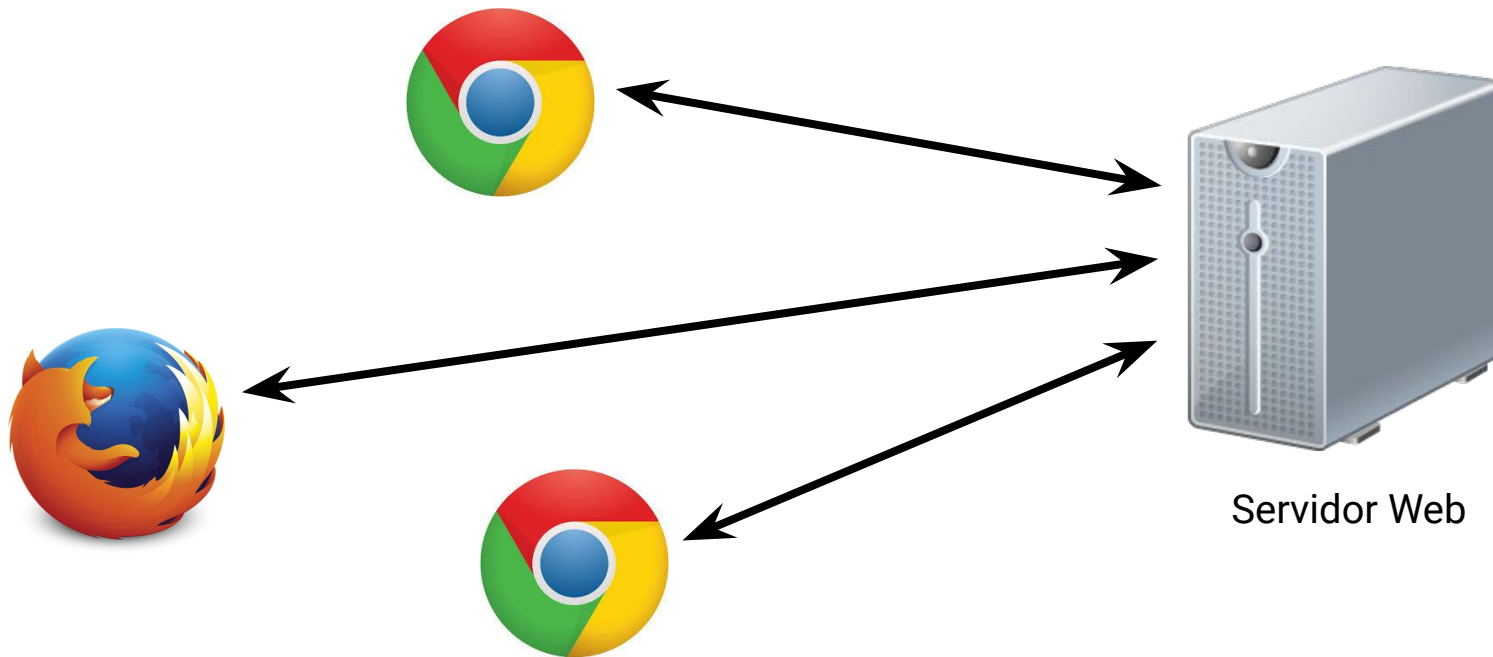


iOS

# ¿Cómo funciona?

1

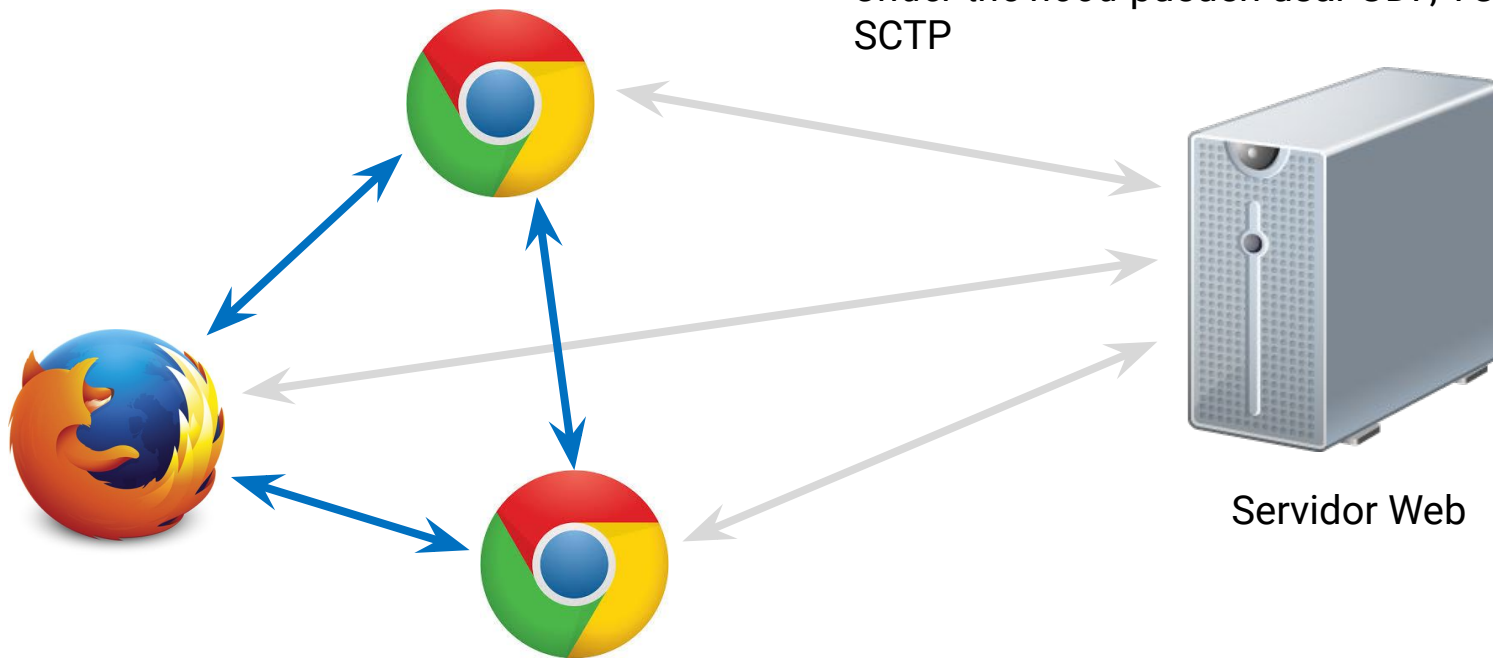
Los browsers se conocen conectándose a la misma aplicación web usando Websockets o REST



# ¿Cómo funciona?

2

Los browsers se conectan directamente entre sí mediante el protocolo WebRTC. *Under the hood* pueden usar UDP, TCP o SCTP



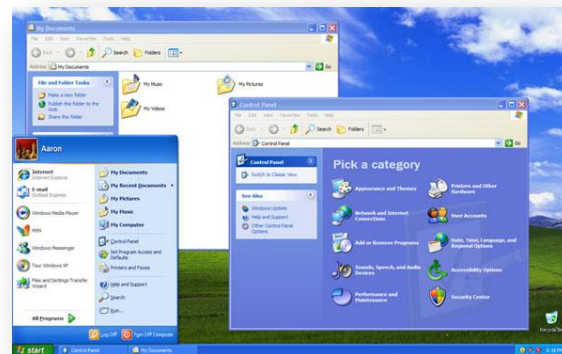
# Fuentes de audio y vídeo en WebRTC



Webcam



Micrófono



Compartición de  
Escritorio



Google Hangouts





GoToMeeting

<https://app.gotomeeting.com/organizer/index.html?meetingid=862097581>

David Williams Jenn Wilson Sue Miller Liz Davis

Q3 EARNINGS

\$15 K \$35 K \$50 K \$100 K \$150 K

Attendees

All All Invite

Liz Davis organizer

Chris Jones presenter

Kim Thomas

Sue Miller

David Williams

Mute David Williams

Make presenter

Make organizer

Excuse David Williams

Talking: Chris Jones

A screenshot of a GoToMeeting web interface. The top section shows four video thumbnails of participants: David Williams, Jenn Wilson, Sue Miller, and Liz Davis. Below this is a large donut chart titled "Q3 EARNINGS" with five segments of different colors (yellow, green, orange, red, blue) and labels: \$15 K, \$35 K, \$50 K, \$100 K, and \$150 K. On the right side, there is a sidebar with a list of attendees. The list includes Liz Davis (organizer), Chris Jones (presenter), Kim Thomas, Sue Miller, and David Williams. A context menu is open over David Williams, showing options: "Mute David Williams", "Make presenter", "Make organizer", and "Excuse David Williams". At the bottom right, a status bar indicates "Talking: Chris Jones".



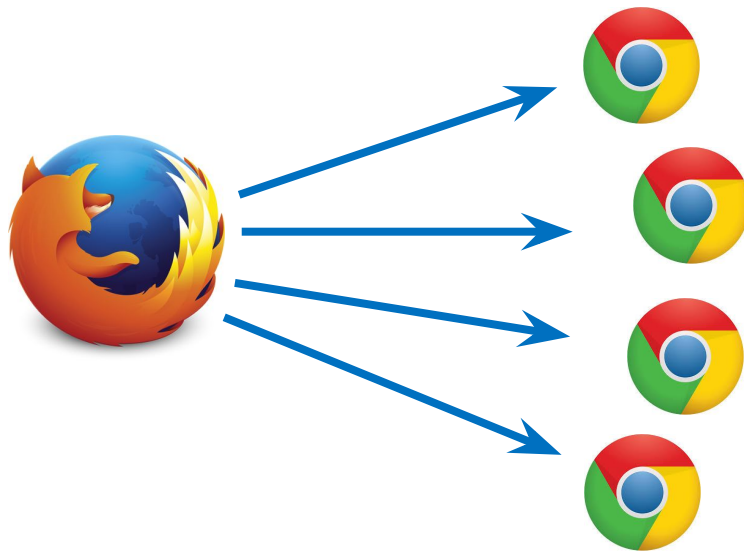


# Tipos de apps WebRTC

Videoconferencia /  
chat



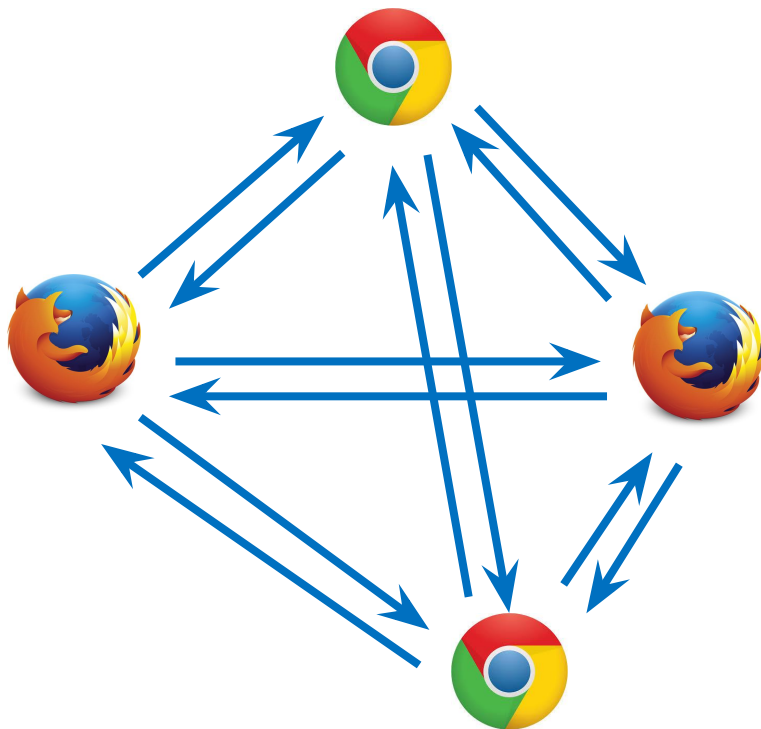
Emisión de media





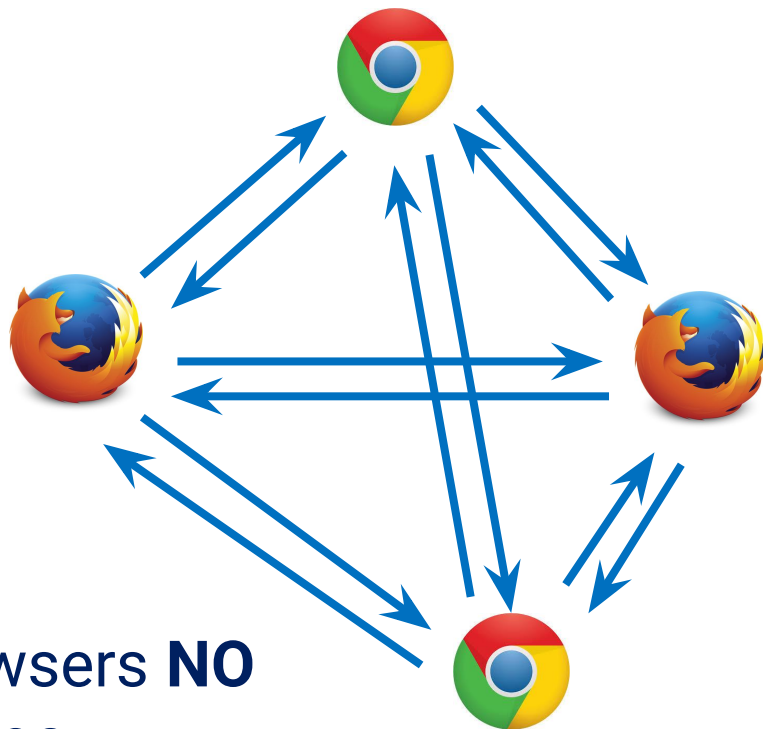
# Tipos de apps WebRTC

Llamadas en grupo



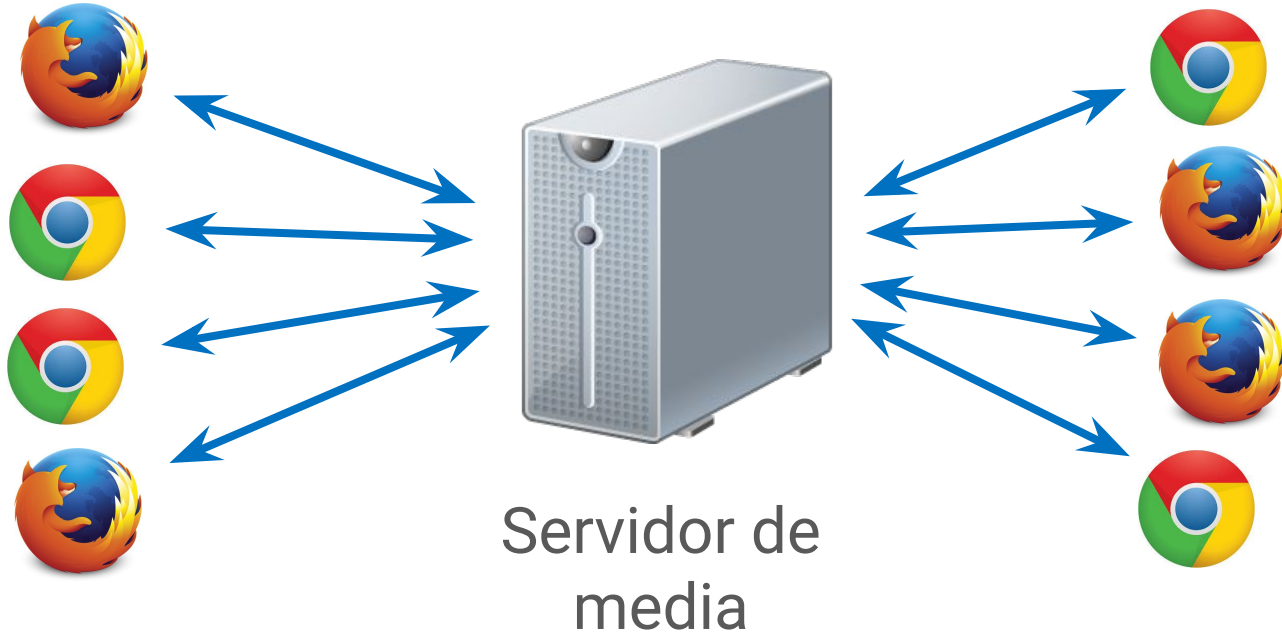
# Tipos de apps WebRTC

Llamadas en grupo



La conexión directa entre browsers **NO** es adecuada en todos los casos

# Servidores de media WebRTC



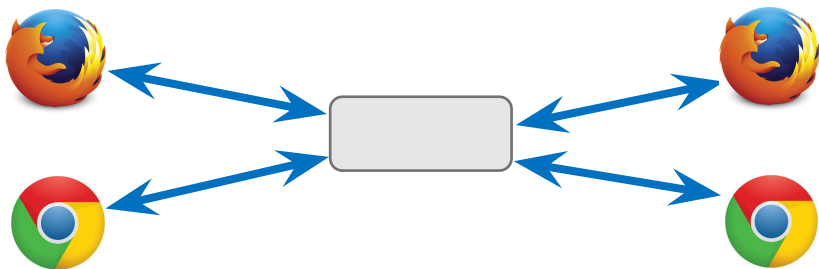
# ¿Qué suelen hacer los servidores de media?

Transcodificar



Reenviar

(Para llamadas en grupo o streaming de baja latencia)



Grabar



# Servidores de media WebRTC



Google Hangouts



appear.in



**GoTo**Meeting

Servicios de videoconferencia cerrados



# Servidores de media programables



As a Service



Servidores  
Comerciales



openVidu



Servidores  
Open Source

**¿y es muy complicado?**









openVidu

# ¿Qué es OpenVidu?

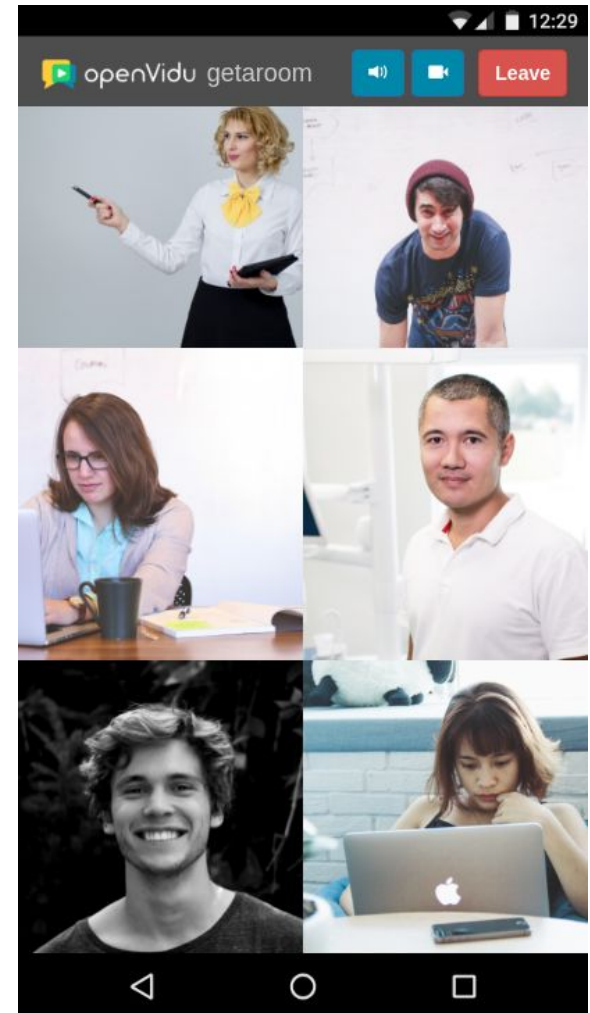
- **Plataforma** para desarrollo de apps WebRTC
- Modelo de programación **sencillo** y múltiples **ejemplos**
- Funcionalidades avanzadas (grabación, filtros, visión artificial...)
- **Open source** (Licencia Apache)

<https://openvidu.io/>



# Videoconferencia

- Modelo de desarrollo simplificado de aplicaciones de **videoconferencia en grupo**





openVidu

OpenVidu\_Call



Nick



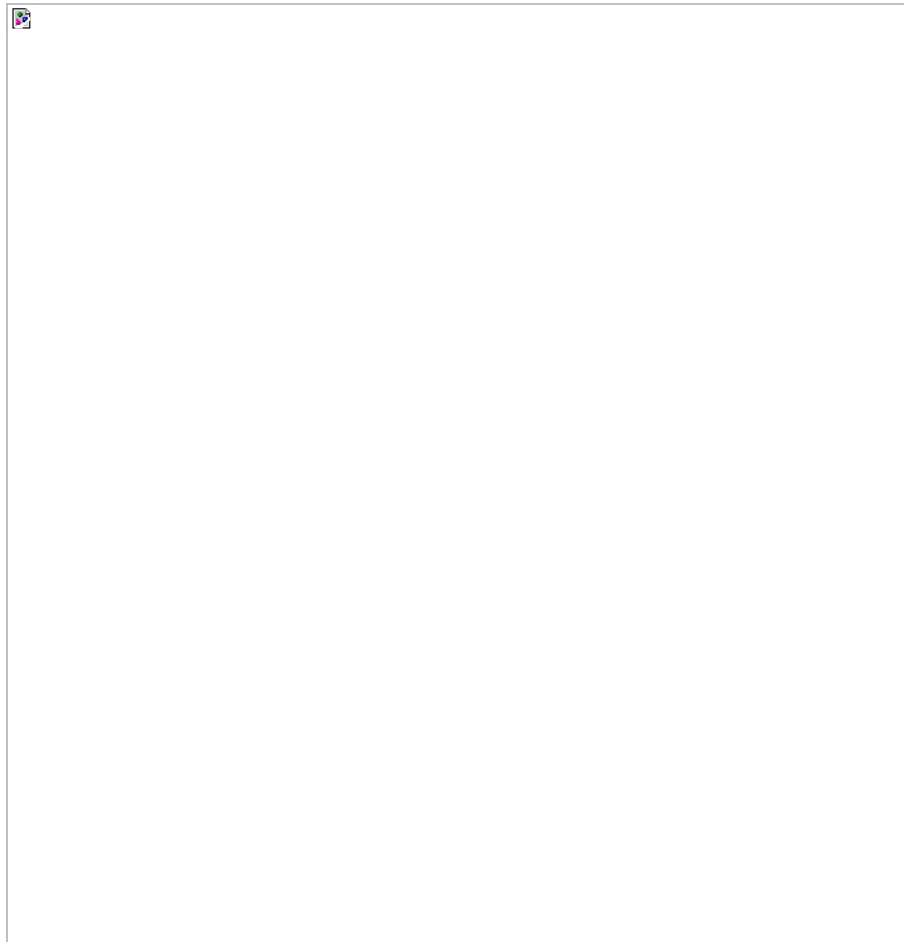
Lewis (edit)



# Tutoriales con múltiples tecnologías



# Arquitectura de una web app con OpenVidu



Server

App code  
(Backend)

App code  
(Frontend)

JavaScript  
Library



# Control de acceso y permisos

- **Backend**

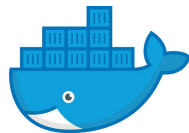
- Se conecta a la **API REST** de **OpenVidu Server**
- Decide qué sesiones existen y qué usuarios pueden conectarse a ella
- Decide qué rol tiene el usuario (PUBLISHER, SUBSCRIBER, MODERATOR)
- Genera token y se lo envía a frontend

- **Frontend**

- Usa el token en OpenVidu Browser para conectarse a la sesión

# Tutoriales / Demos

- Ejecutar servidor OpenVidu



```
docker run -p 4443:4443 --rm -e openvidu.secret=MY_SECRET openvidu/openvidu-server-kms:2.6.0
```

- Ejecutar la aplicación de demo



```
mvn clean package exec:java
```



```
ng serve --open
```



```
node server.js
```

## Initialize a new session and its events

```
OV = new OpenVidu();  
session = OV.initSession();  
  
session.on('streamCreated', function (event) {  
    session.subscribe(event.stream, 'subscriber');  
});
```

<https://openvidu.io/docs/tutorials/openvidu-insecure-js/>

# Vainilla JS

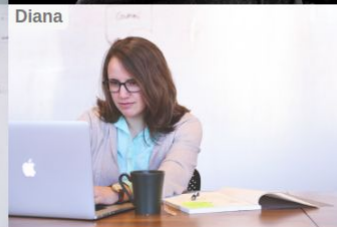


## Connect to the session using the token

```
getToken(mySessionId).then(token => {  
  
  session.connect(token)  
    .then(() => {  
      document.getElementById("session-header").innerText = mySessionId;  
      document.getElementById("join").style.display = "none";  
      document.getElementById("session").style.display = "block";  
  
      var publisher = OV.initPublisher("publisher");  
      session.publish(publisher);  
    })  
    .catch(error => {  
      console.log("There was an error connecting to the session:", error.code, error.message);  
    });  
});
```

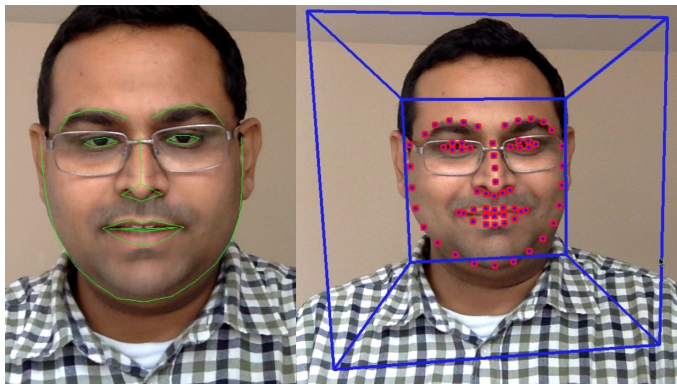
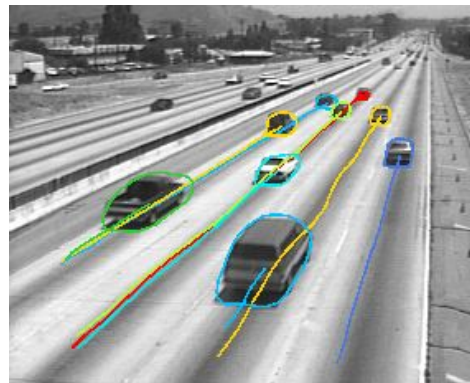
## My Session

Leave session



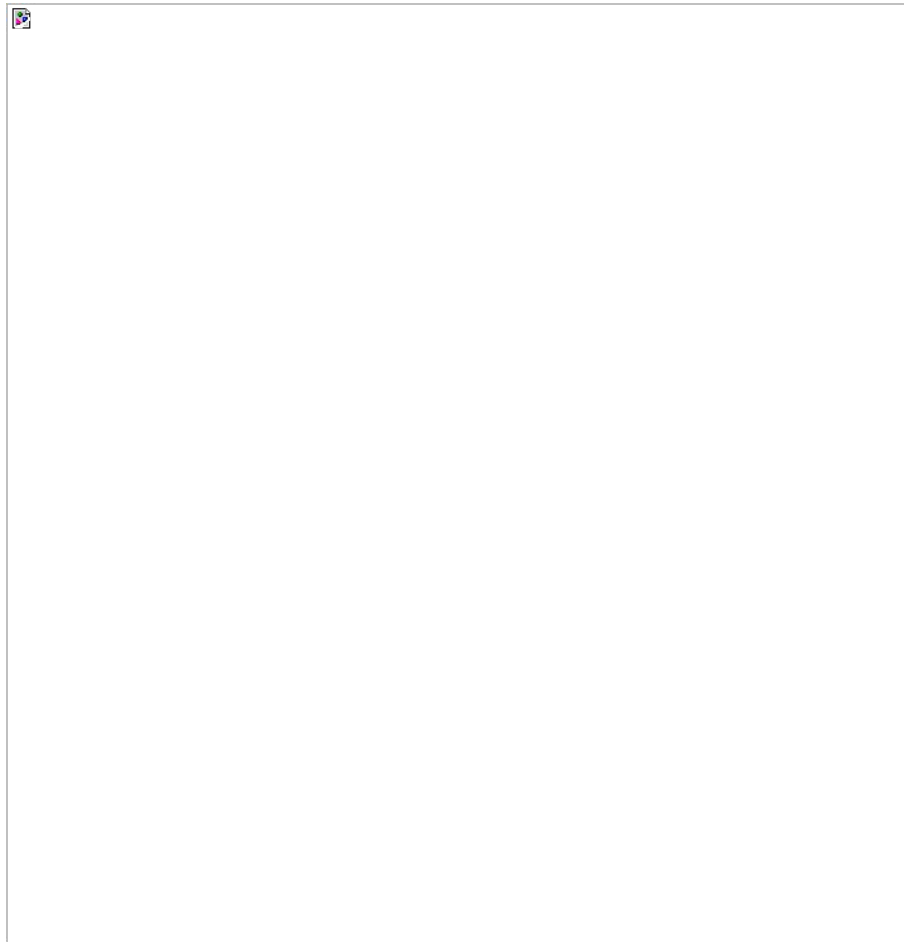
# Funcionalidades avanzadas

- Compartir **pantalla**
- **Grabación** de la sesión
  - Como se ve en el navegador
  - Diseños personalizados
- Filtros de **visión artificial**
- Detector de **voz**
- Conexión a la cámaras (p.e. **drone**)





# Arquitectura de una web app con OpenVidu



Server

App code  
(Backend)

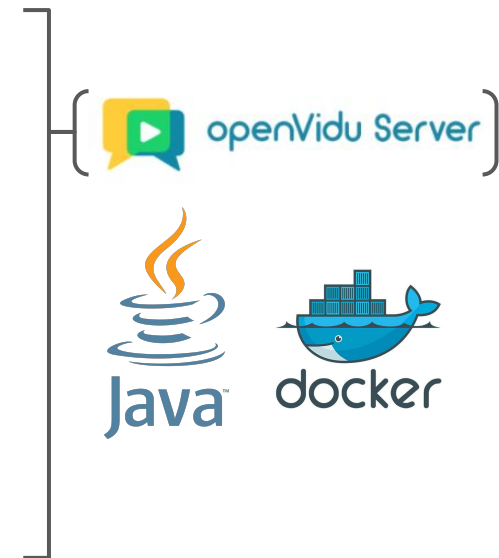
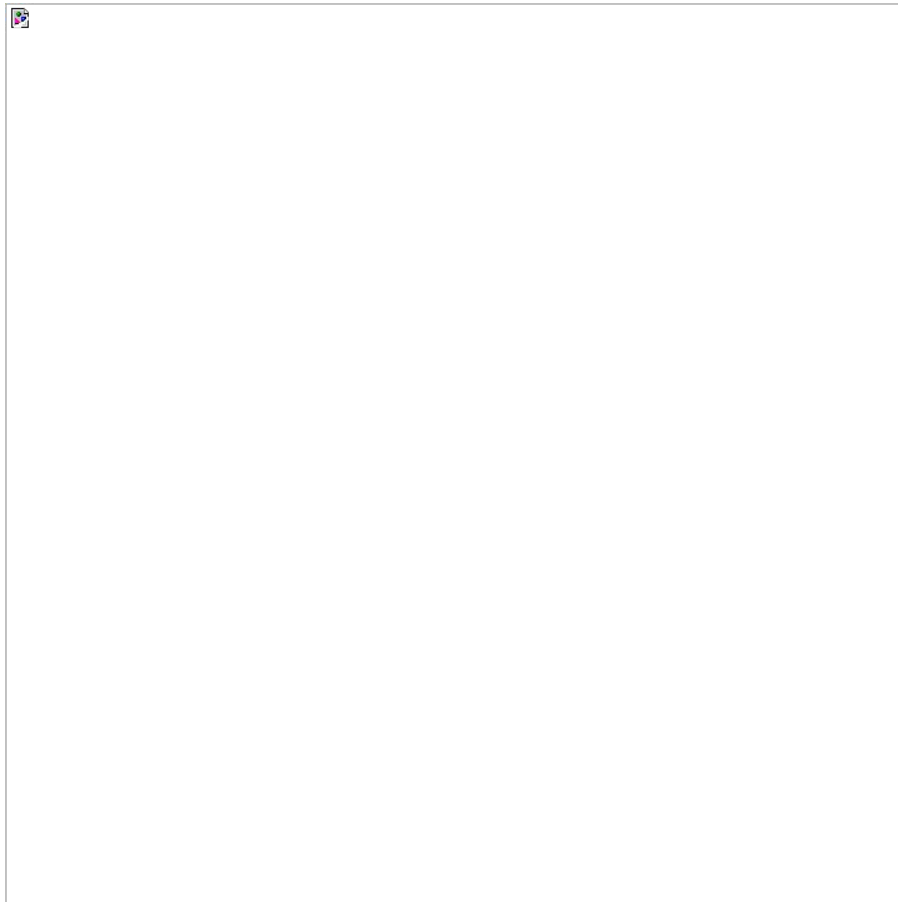
App code  
(Frontend)

JavaScript  
Library



Un poco más  
a fondo...







 openVidu Browser

TS

JS

 openVidu Server

  
Java

  
docker

  
KURENTO





Servidor de media **WebRTC**

Control de **bajo nivel** del procesamiento del media en **pipelines**

Controlable con un protocolo sobre **WebSockets**

**También desarrollado en la URJC!**

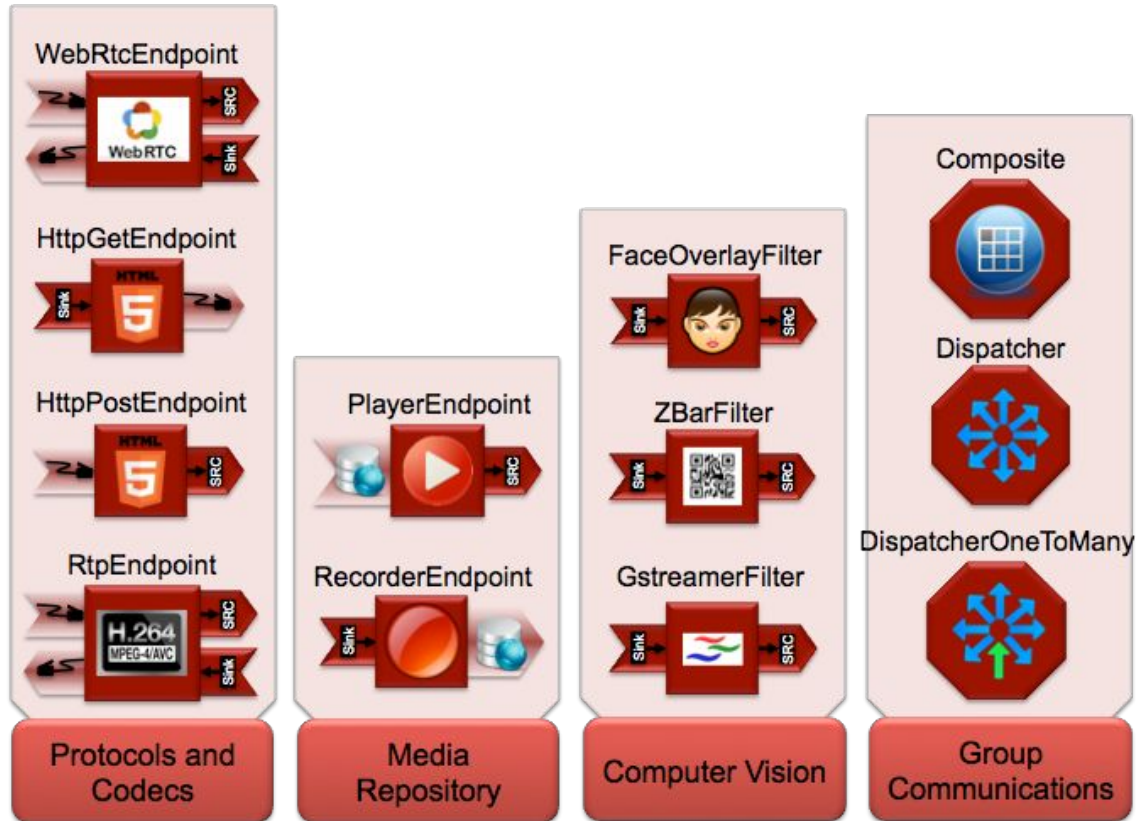
<http://www.kurento.org>



C/C++

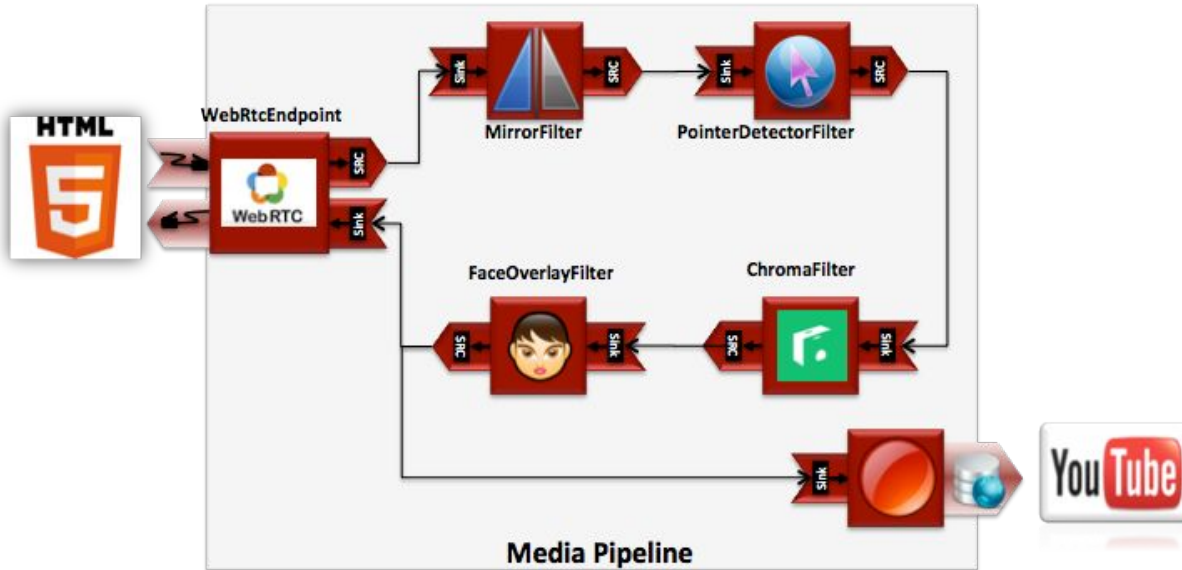
<http://www.kurento.org>

# Biblioteca de componentes predefinidos



# Media Pipeline

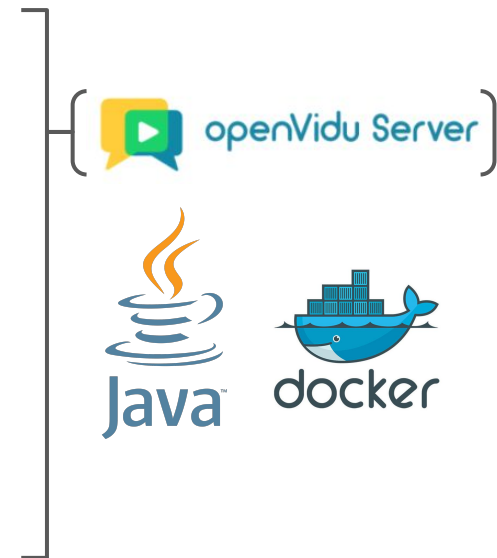
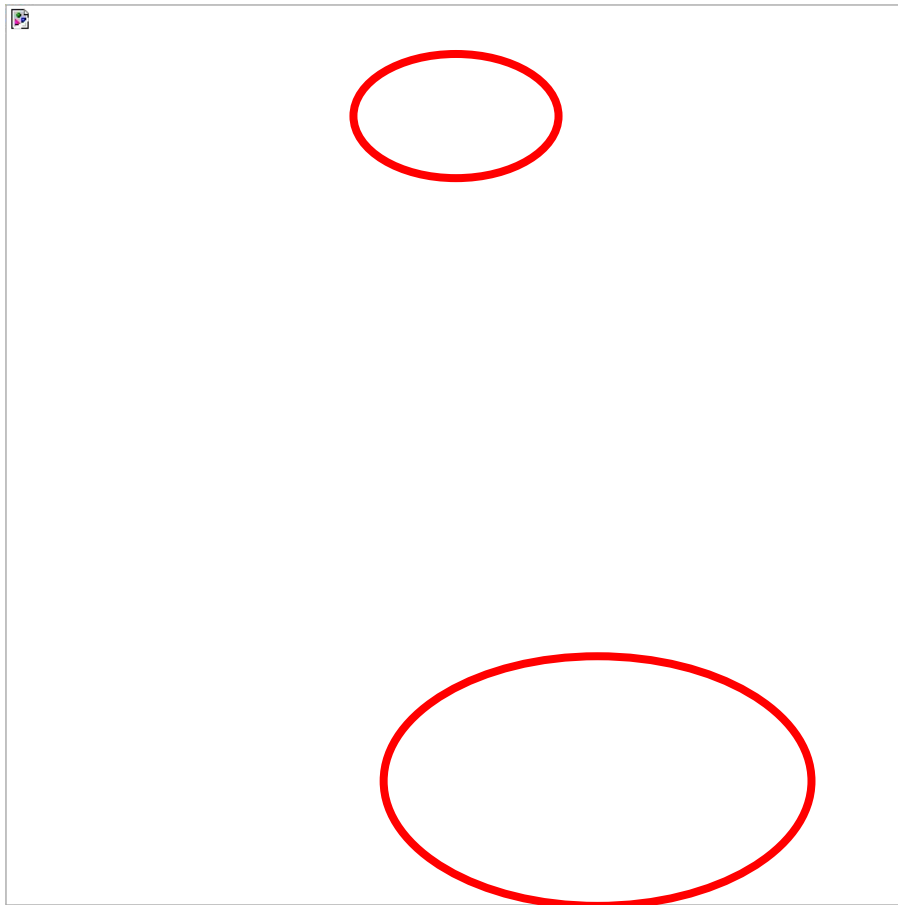
- Cadena de elementos que implementan la funcionalidad deseada





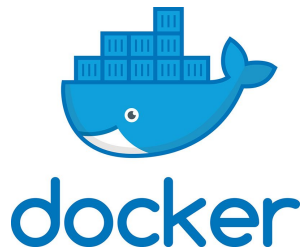






# OpenVidu Browser & Server

- Simplifica el modelo de programación sobre Kurento (o cualquier futuro servidor de media)
- Gestiona sesiones, permisos, negociación de media
- Despliegues sencillos sobre proveedores cloud (CloudFormation) o tu propia infraestructura (Ansible)
- Componentes front de alto nivel





@fuentepab



@pabloFuente



pablofuenteperez@gmail.com



# Gracias!!

