

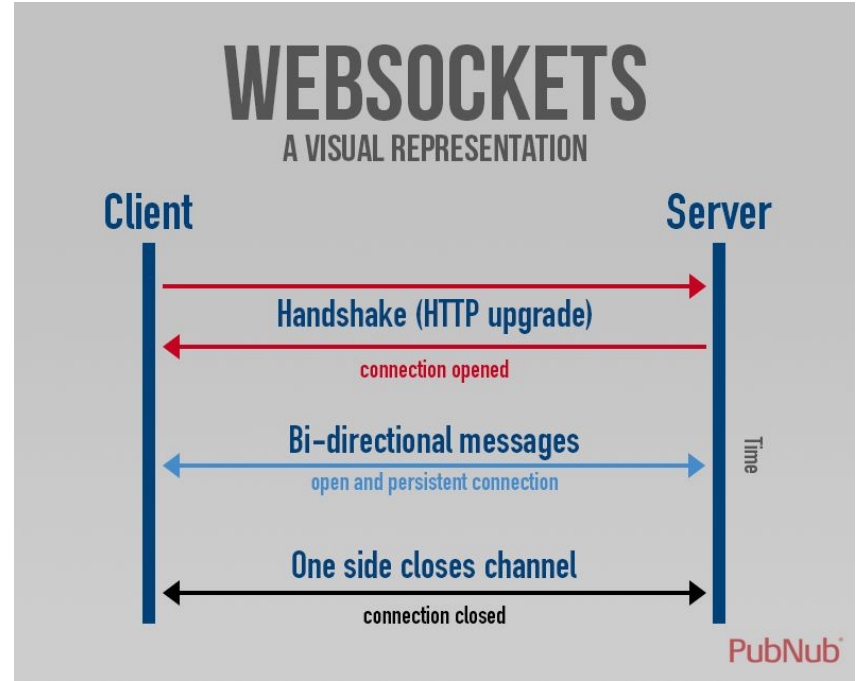
# Les WebSockets

Programmation sur le client



# Le protocole WebSocket

- Créer un canal de communication serveur/client
- Communication bilatérale sous forme de message
- Protocole basé sur le système d'événement
- Message texte



# Utilisation des WebSockets en Javascript

## Création d'un objet WebSocket

- Connexion sécurisée
- Connexion non sécurisée

```
var socket = new WebSocket("ws://localhost")
```

```
var socket = new WebSocket("wss://localhost")
```

L'objet dispose de 2 méthodes et 2 gestionnaire d'événements

- close()
- send()
- onopen = function(event)
- onmessage = function(event)

# Démonstration

## WebSockets



# Compatibilité

## Web Sockets - LS

## Bidirectional communication technology for web apps

Usage	% of all users	
Global	92.99% + 0.2%	= 93.18%
unprefixed:	92.99% + 0.16%	= 93.14%
France	93.78% + 0.11%	= 93.88%
unprefixed:	93.78% + 0.11%	= 93.88%

[illegible]

# Les avantages et les inconvénients

+	-
Communication bidirectionnelle	Peu de précision sur la nature du message
Jusqu'à 2go de données envoyées	Toute la réception dans onmessage
	Compatibilité anciens navigateurs

# SocketIO



Pleins de **méthode de communications**

- WebSocket
- Adobe Flash Socket      ⇒ Si Flash installé
- AJAX long polling      ⇒ Client demande en continu au serveur s'il y a des informations
- Forever Iframe      ⇒ Iframe invisible qui récupère les informations du serveur
- etc...



Le choix de la méthode est **AUTOMATIQUE**  
et dépend du **NAVIGATEUR**





- Internet Explorer 5.5+
- Safari 3+
- Google Chrome 4+
- Firefox 3+
- Opera 10.61+
- Safari sur iPhone et iPad
- Le navigateur Android





# Démonstration



SOCKET.IO

```
/* Configuration Websocket */
```

```
function sendAllClients(clients,type, msg){  
  for(let i=0; i < clients.length; i++) {  
    clients[i].send(JSON.stringify({type:type, content:msg}));  
  }  
}
```

```
const WebSocketServer = WS.Server;  
const wss = new WebSocketServer({server});
```

```
let clients = [];  
let messages = [];
```

```
wss.on('connection', function(ws){  
  clients.push(ws);  
  ws.send(JSON.stringify({type:'messages', content:messages}));  
  
  ws.on('message', function(msg){  
    msg = JSON.parse(msg);  
    let currentDate = new Date();  
    let msgDate = currentDate.getHours() + ':' + (currentDate.getMinutes()  
    msg = `<p>${msg.author}@${msgDate}----${msg.data}</p>`;  
    messages.push(msg);  
    sendAllClients(clients, 'message', msg);  
  });  
  
  ws.on('close', function(){  
    let index = clients.indexOf(ws);  
    clients.splice(index, 1);  
  });  
});
```

```
/* Configuration Socket.IO */
```

```
const io = require('socket.io')(server);  
let messages = [];
```

```
io.on('connection', function(socket){  
  socket.emit('messages', messages);  
  
  socket.on('message', function(content){  
    let currentDate = new Date();  
    let msgDate = currentDate.getHours() + ':' + (currentDate.getMinutes()  
    msg = `<p>${content.author}@${msgDate}----${content.data}</p>`;  
    messages.push(msg);  
    io.sockets.emit('message', msg);  
  });  
});
```



JohanSaltutti/websockets-starter

JohanSaltutti/socketio-starter

CIASIE-2018/Mysterium