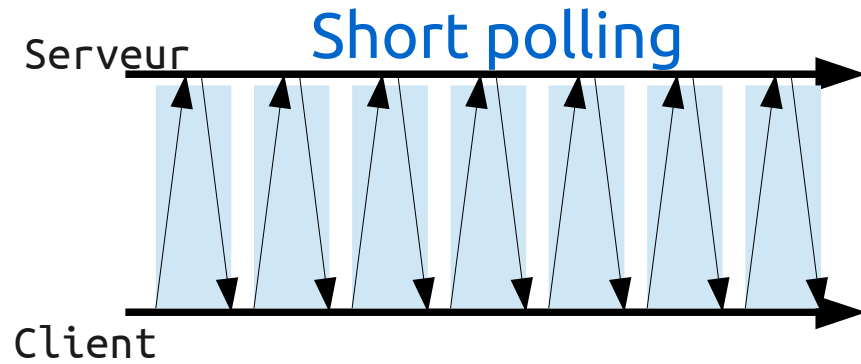


WEBSOCKET

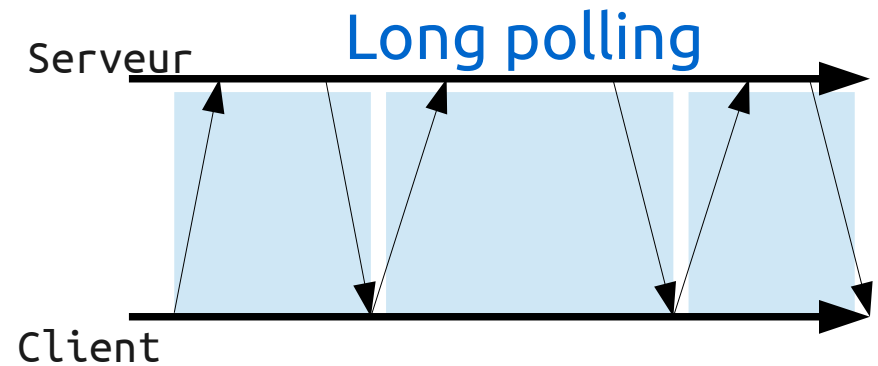
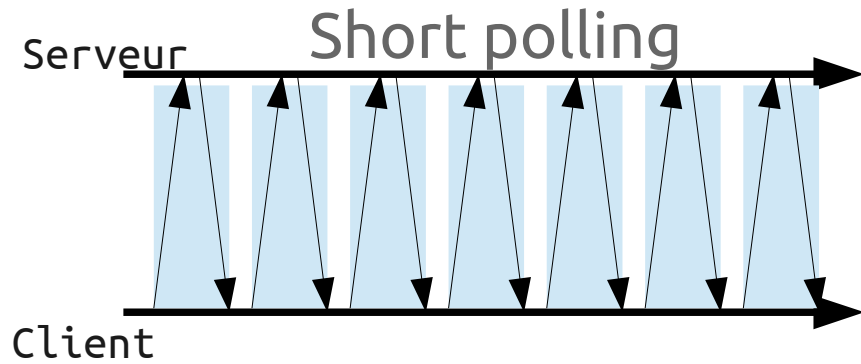
JRES 2013 Montpellier

Jérôme Bousquié
IUT de Rodez
Université de Toulouse 1 Capitole

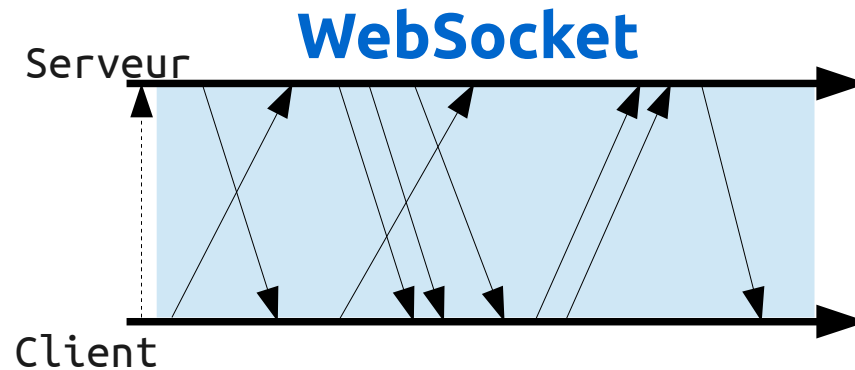
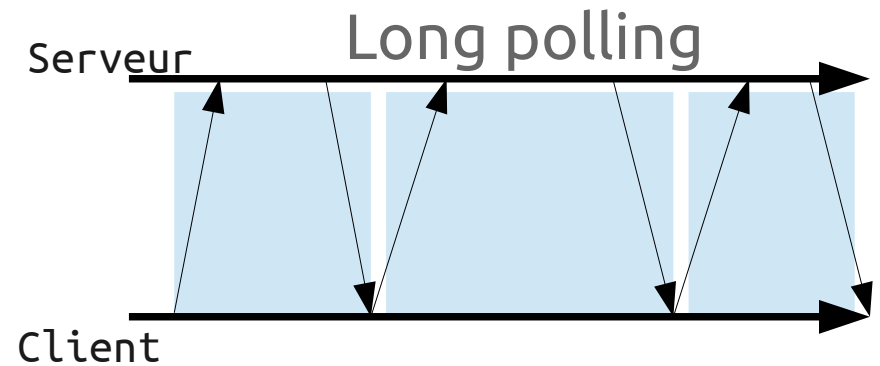
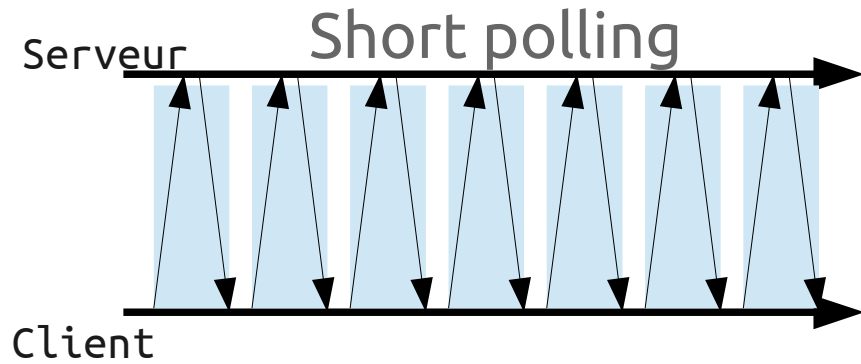
simuler « temps réel » ou « push » sur le web



simuler « temps réel » ou « push » sur le web



simuler « temps réel » ou « push » sur le web



Poignée de main de connexion

Requête handshake

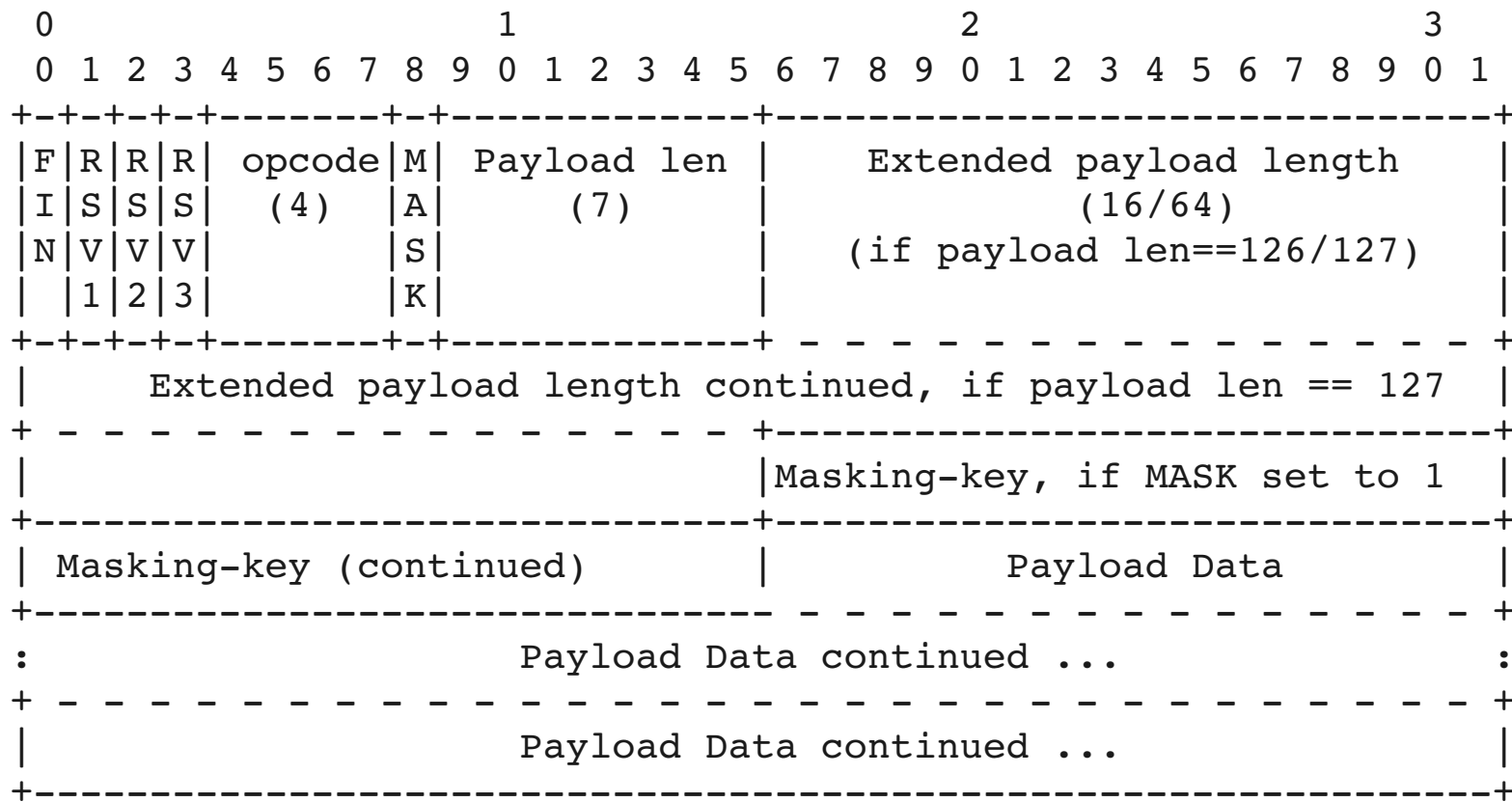
```
GET /socket HTTP/1.1  
Host: websocket.server.org  
Connection: Upgrade  
Upgrade: websocket  
Sec-WebSocket-Key: dGhIIHNhbXBsZSBub25jZQ==  
Origin: http://mon.siteweb.net  
Sec-WebSocket-Protocol: soap, xmpp, stomp  
Sec-WebSocket-Version: 13
```

Réponse handshake

```
HTTP/1.1 101 Switching Protocols  
Connection: Upgrade  
Upgrade: websocket  
Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+xOo=  
Sec-WebSocket-Protocol: xmpp
```

Trames typées :

texte, binaire ou contrôle



Sécurité

dans le browser : Same Origin Policy (RFC 6454)
à la connexion : Sec-WebSocket-Key dans le handshake
sur le flux : TLS wss://ws.server.net:443/path
dans le protocole : contrôle des trames

Proxies explicites ou transparents et firewalls

non chiffré : dépend du comportement du relais
http connect,
headers connection, upgrade,
inspection http, durée connexion

chiffré sur port 443 : passe toujours si https passe déjà

Côté client : API HTML5

```
<script type='text/javascript'>

// Connexion au serveur
var ws = new WebSocket('ws://ws.server.net');

// Réception d'un message du serveur
ws.onmessage = function(e) {
    console.log('message reçu : ' + e.data);
};

// Envoi d'un message texte
ws.send('Coucou serveur');

// Envoi de données binaires
var file =
    document.querySelector('input[type="file"]').files[0];
ws.send(file);
```


Côté serveur

Librairies

C/C++, C#, Objective-C, perl, python, ruby, php, etc

Serveurs

Jetty, Kaazing : Java

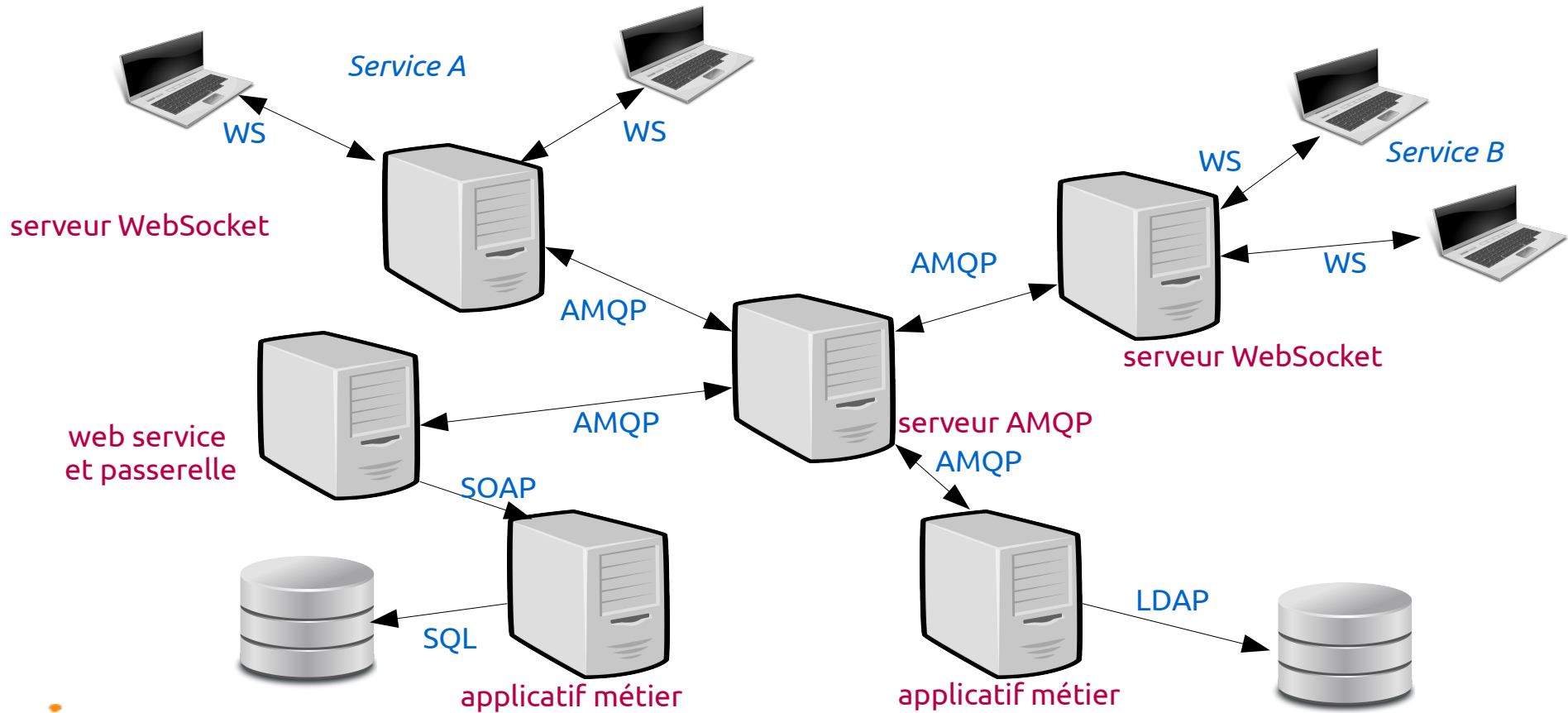
Vert.X : Java, Javascript, CoffeeScript, Ruby, Python, Groovy

Node.js : Javascript

```
var connection = request.accept(null, request.origin);

connection.on('message', function(msg) {
  if (msg.type === 'utf8') {
    var reponseTexte = traiteTexte(msg.utf8Data) ;
    connection.sendUTF(reponseTexte);      }
  else if (msg.type === 'binary') {
    var reponseBinaire = traiteBinaire(msg.binaryData) ;
    connection.sendBytes(reponseBinaire); }  });
```

Exemple d'architecture événementielle WebSocket + AMQP



Questions ?