



## COMET, l'autre direction du web

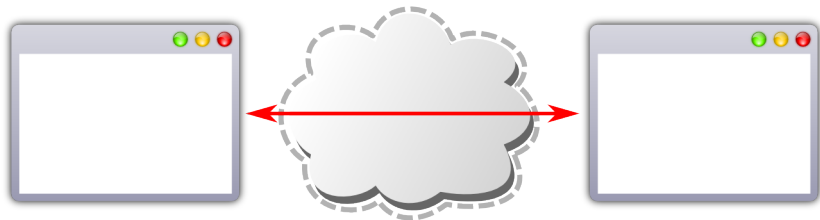
Viktor Horvath, Alixen  
viktor@fsfe.org

OSDC 2010

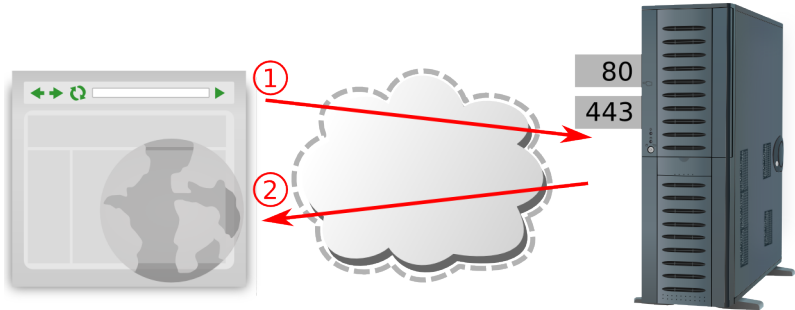
# Plan

- ① Motivation
- ② COMET
- ③ Implémentations
- ④ Démonstration

# Applications conventionnelles



# Applications web



## Motivation pour le design pattern COMET

- 1 On veut que le **serveur** puisse initier la communication...

## Motivation pour le design pattern COMET

- ① On veut que le **serveur** puisse initier la communication...
- ② de manière **efficace**...

## Motivation pour le design pattern COMET

- ① On veut que le **serveur** puisse initier la communication...
- ② de manière **efficace**...
- ③ avec les navigateurs **d'aujourd'hui**.

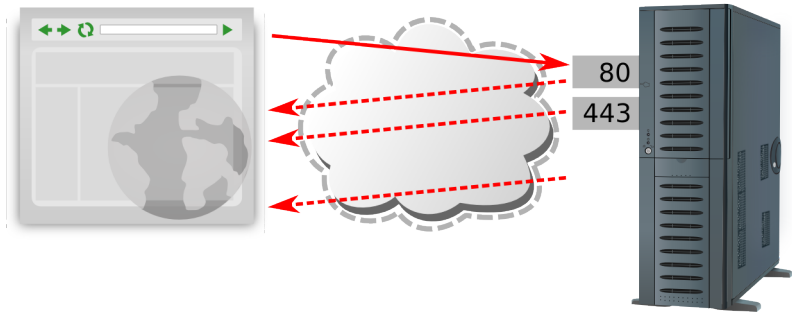
## Motivation pour le design pattern COMET

- ① On veut que le **serveur** puisse initier la communication...
- ② de manière **efficace**...
- ③ avec les navigateurs **d'aujourd'hui**.

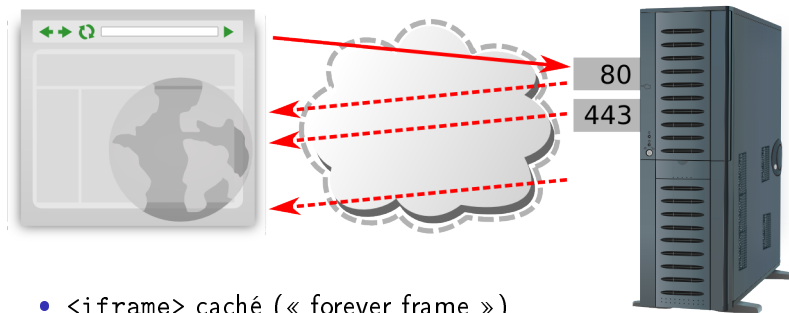
Établissement des connexions permanentes,  
utilisable à tout moment par le serveur



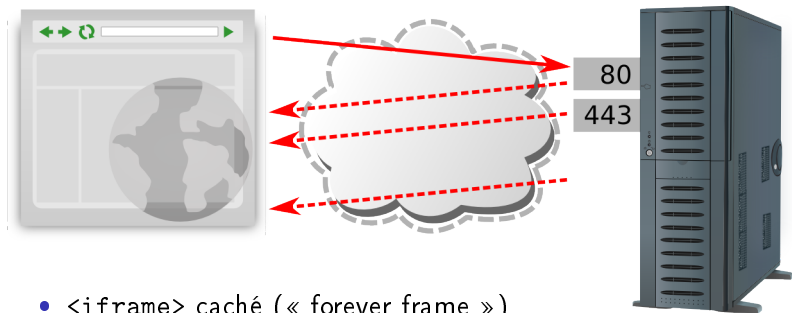
## Mode « streaming », connexions persistantes HTTP 1.1



## Mode « streaming », connexions persistantes HTTP 1.1

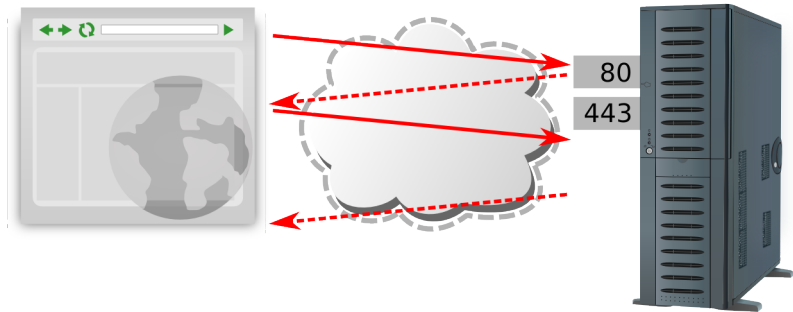


## Mode « streaming », connexions persistantes HTTP 1.1

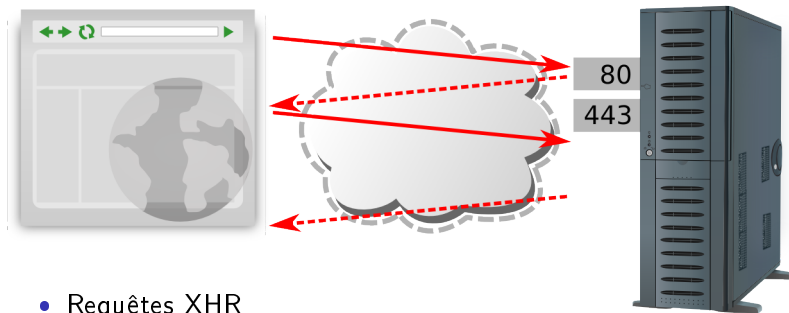


- `<iframe>` caché (« forever frame »)
- Gecko : réponse XHR multipart

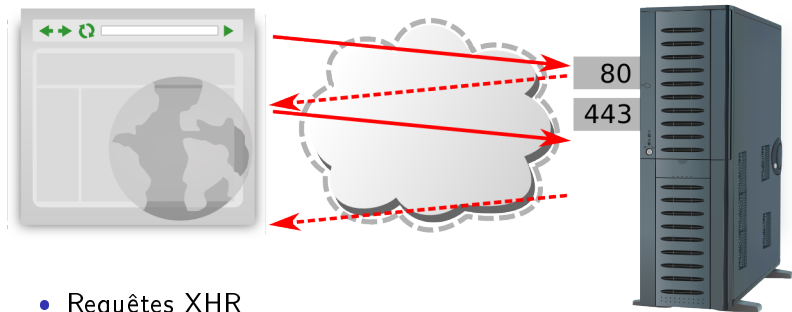
## Mode « long-polling »



## Mode « long-polling »



## Mode « long-polling »



- Requêtes XHR
- Tags `<script>`

## Protocoles

- Orienté vers **canaux nommés** de communication (publish/subscribe) :

## Protocoles

- Orienté vers **canaux nommés** de communication (publish/subscribe) :
  - Le protocole Bayeux



# Protocoles

- Orienté vers **canaux nommés** de communication (publish/subscribe) :
  - Le protocole Bayeux
  - Protocoles propres à une solution serveur/client

## Protocoles

- Orienté vers **canaux nommés** de communication (publish/subscribe) :
  - Le protocole Bayeux
  - Protocoles propres à une solution serveur/client
- Simulation des **sockets** : BOSH

## Protocoles

- Orienté vers **canaux nommés** de communication (publish/subscribe) :
  - Le protocole Bayeux
  - Protocoles propres à une solution serveur/client
- Simulation des **sockets** : BOSH
- Vers les **bus de messages** : AMQP

## Alternatives

- HTML 5 : WebSockets, server-side events

## Alternatives

- HTML 5 : WebSockets, server-side events
- Adobe Flash, Silverlight ou autres plug-ins

## Serveurs

<i>Nom</i>	<i>Protocole</i>	<i>Technique</i>	<i>écrit en</i>
Jetty	Bayeux	LP WS	Java
Meteor	[propre]	LP STR SSE	Perl
Orbited	Sockets, message bus	LP STR SSE WS	Python
Ajax Push Engine	[propre]	LP STR SSE WS	C
ErlyComet	Bayeux	LP WS	Erlang
nginx HTTP Push	Basic HTTP Push Relay Protocol	LP	C
em-websocket	EventMachine	WS	Ruby

... et encore plus !

LP = long polling, STR = streaming, SSE = Server-Sent Events, WS = WebSockets

## Liens

Article Wikipedia . . . . . [http://en.wikipedia.org/wiki/Comet\\_\(programming\)](http://en.wikipedia.org/wiki/Comet_(programming))  
 AJAX Design Patterns : « HTTP Streaming »  
 . . . . . [http://ajaxpatterns.org/HTTP\\_Streaming](http://ajaxpatterns.org/HTTP_Streaming)  
 Tutoriel en BD et serveur/client sur . <http://www.ape-project.org>  
 Nouvelles sur COMET . . . . . <http://cometdaily.com>  
 Comparaison des serveurs . . . . . <http://cometdaily.com/maturity.html>  
 Protocole Bayeux, CometD implementations . . . <http://cometd.org>  
 nginx HTTP Push avec client Ruby  
 . . . <http://www.igvita.com/2009/10/21/nginx-comet-low-latency-server-push>  
 COMET en Java (Jetty et Tomcat)  
 . . . <http://www.javaworld.com/javaworld/jw-03-2008/jw-03-asynchhttp.html>  
 HTML 5 : Server-Side Events et WebSockets  
 . . . . . <http://www.java.net/author/gregor-roth>

## Démonstration



`http://comet.steckenpferde.de:8080`

`http://comet.steckenpferde.de`

Chat démo, à regarder avec l'analyse du trafic réseau :

- Firefox + Firebug : version Dojo
- Chromium + Web Developer Tools : version jQuery