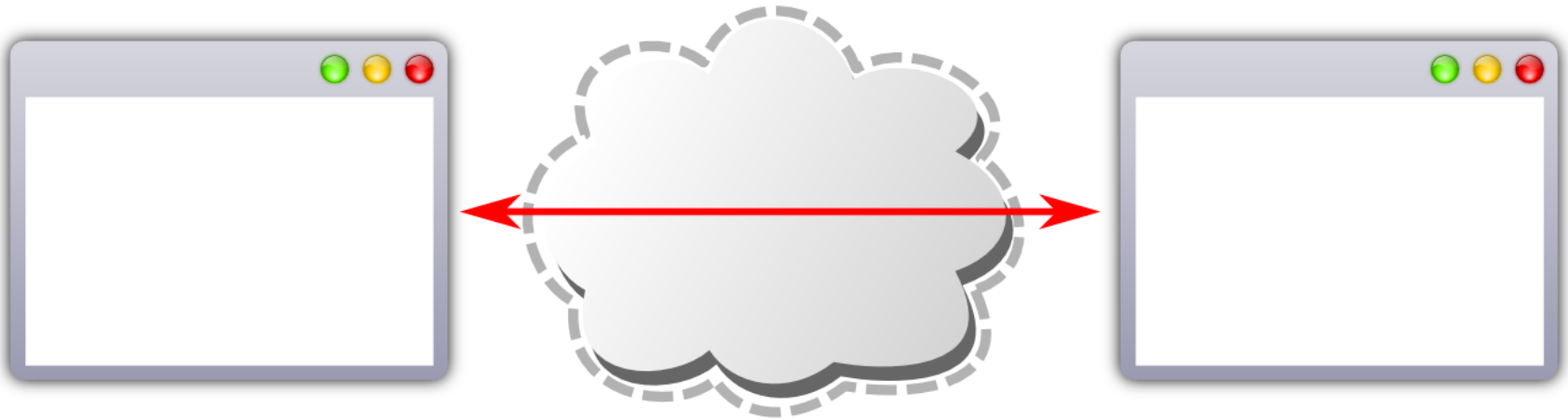
COMET, l'autre direction du web

Viktor Horvath, Alixen
viktor@fsfe.org

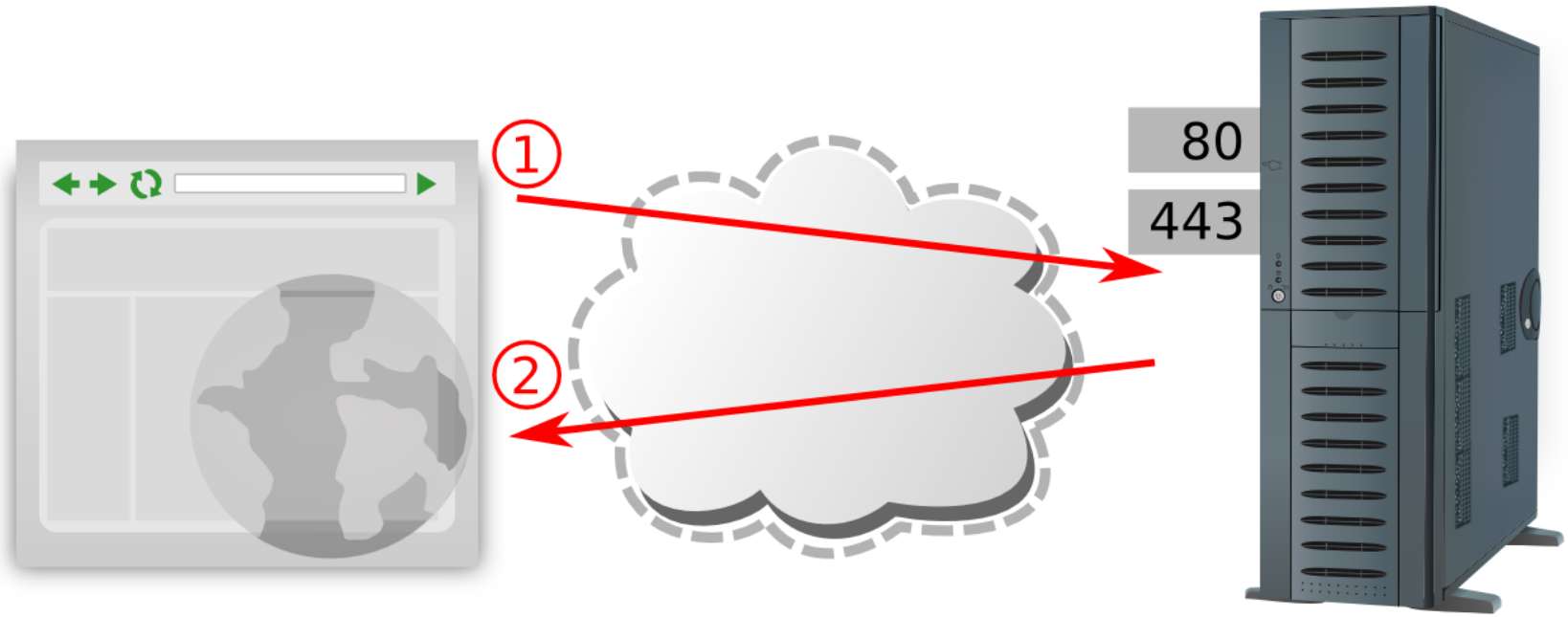
27 novembre 2010

Plan

Applications conventionnelles



Applications web

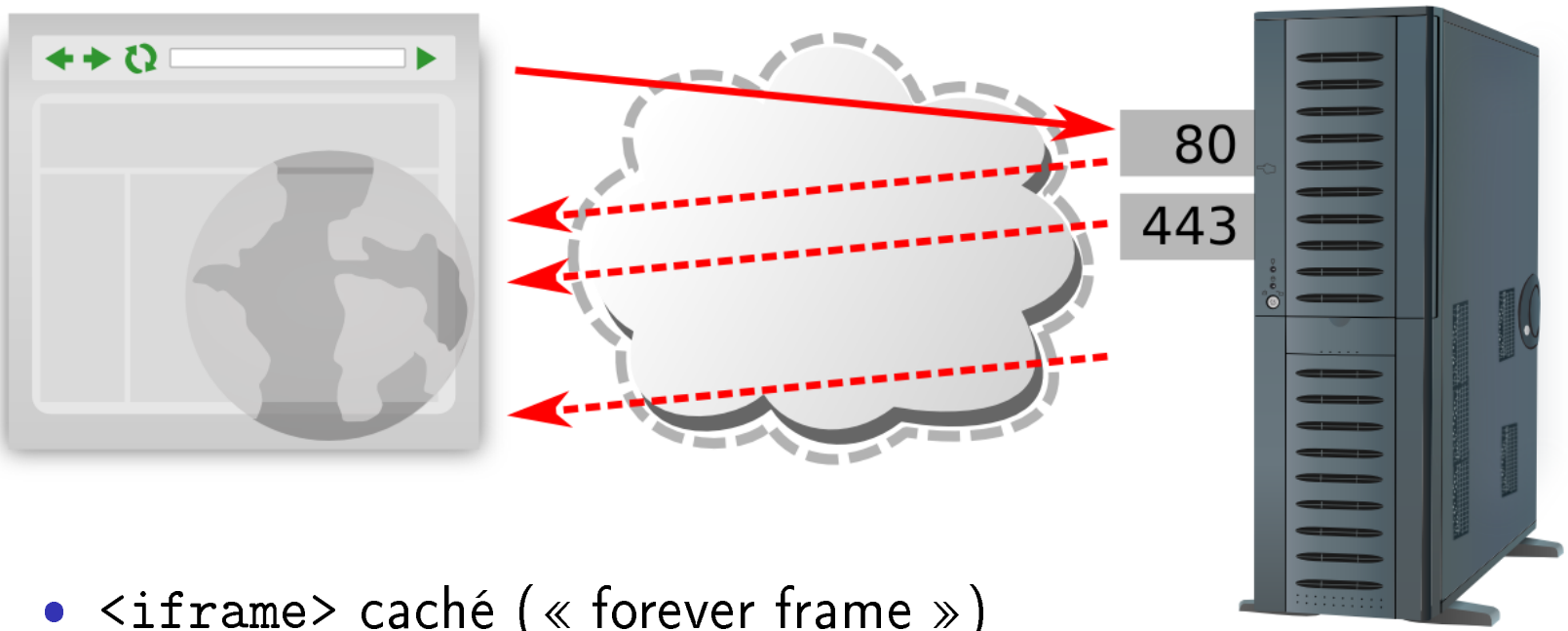


Motivation pour le design pattern COMET

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

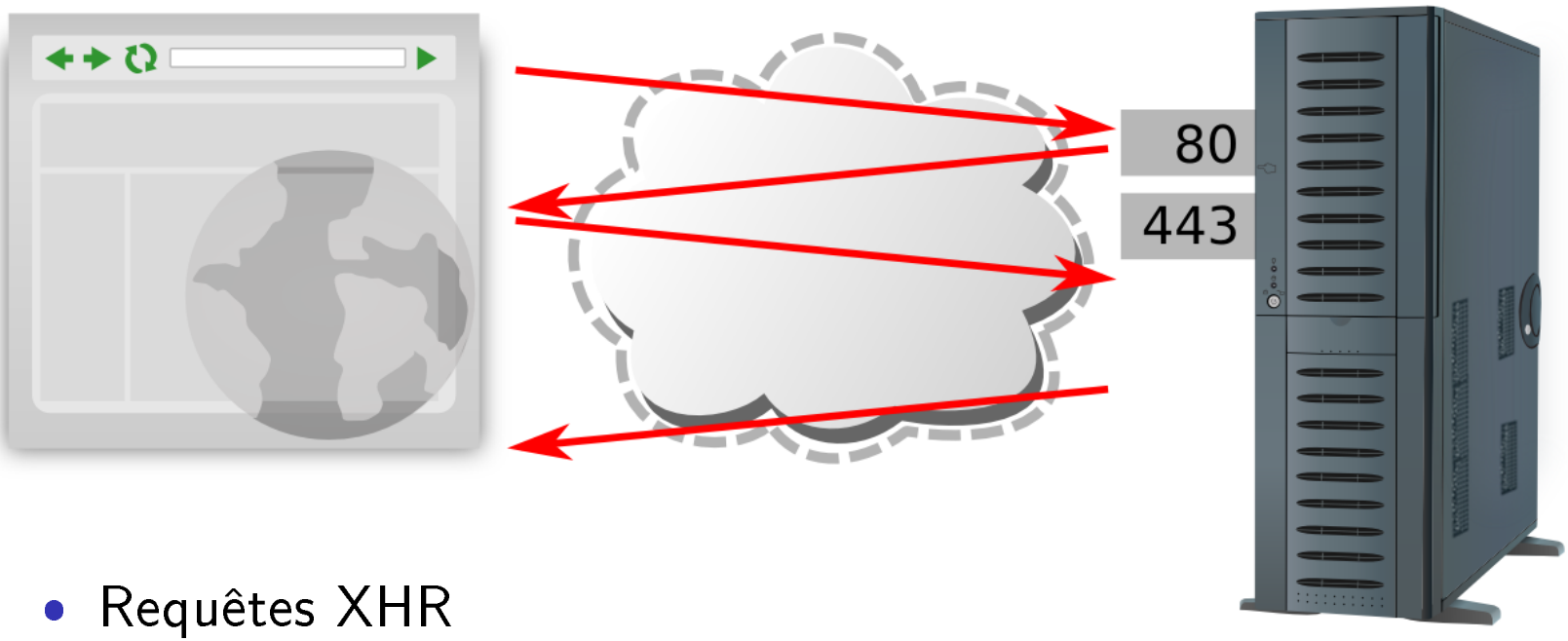
Établissement des connexions permanentes,
utilisable à tout moment par le serveur

Mode « streaming », connexions persistantes HTTP 1.1



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

Mode « long-polling »



- Requêtes XHR
- Tags `<script>`

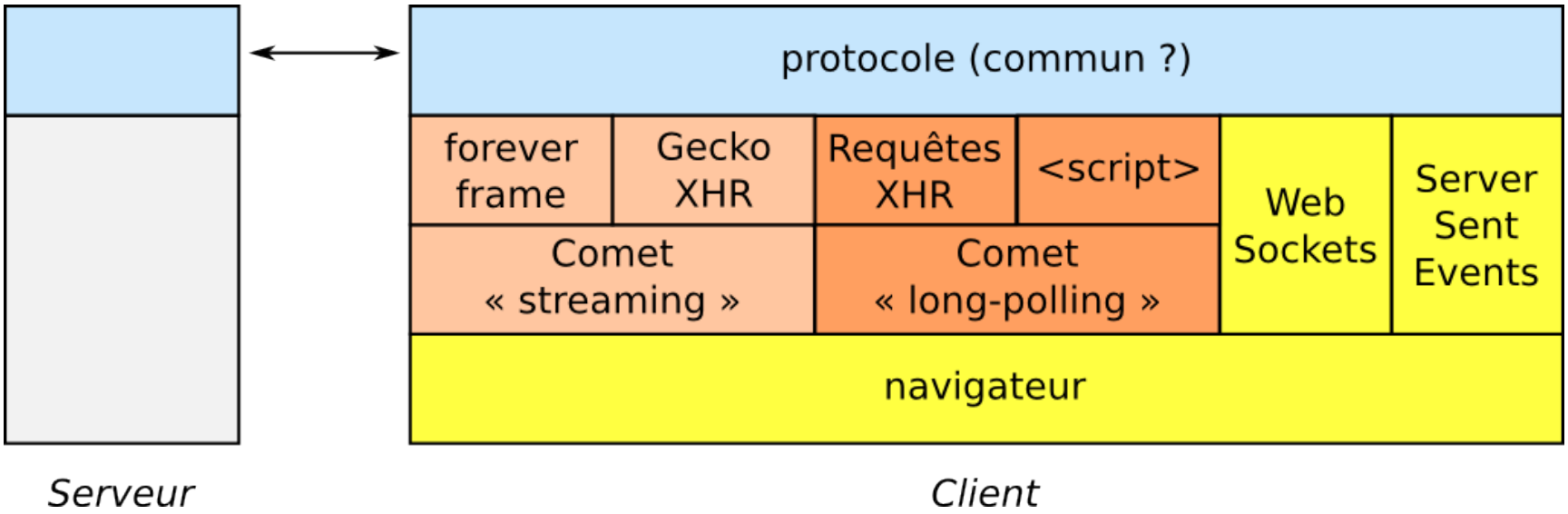
Alternatives : HTML 5

- WebSockets
  - Comme un socket TCP, avec le modèle de sécurité de HTTP
  - `ws://` et `wss://`
  - Besoin de mettre à jour proxies et firewalls
- Server–Side Events

Autres Alternatives

- Adobe Flash, Silverlight ou autres plug–ins
- Java ?

Architecture



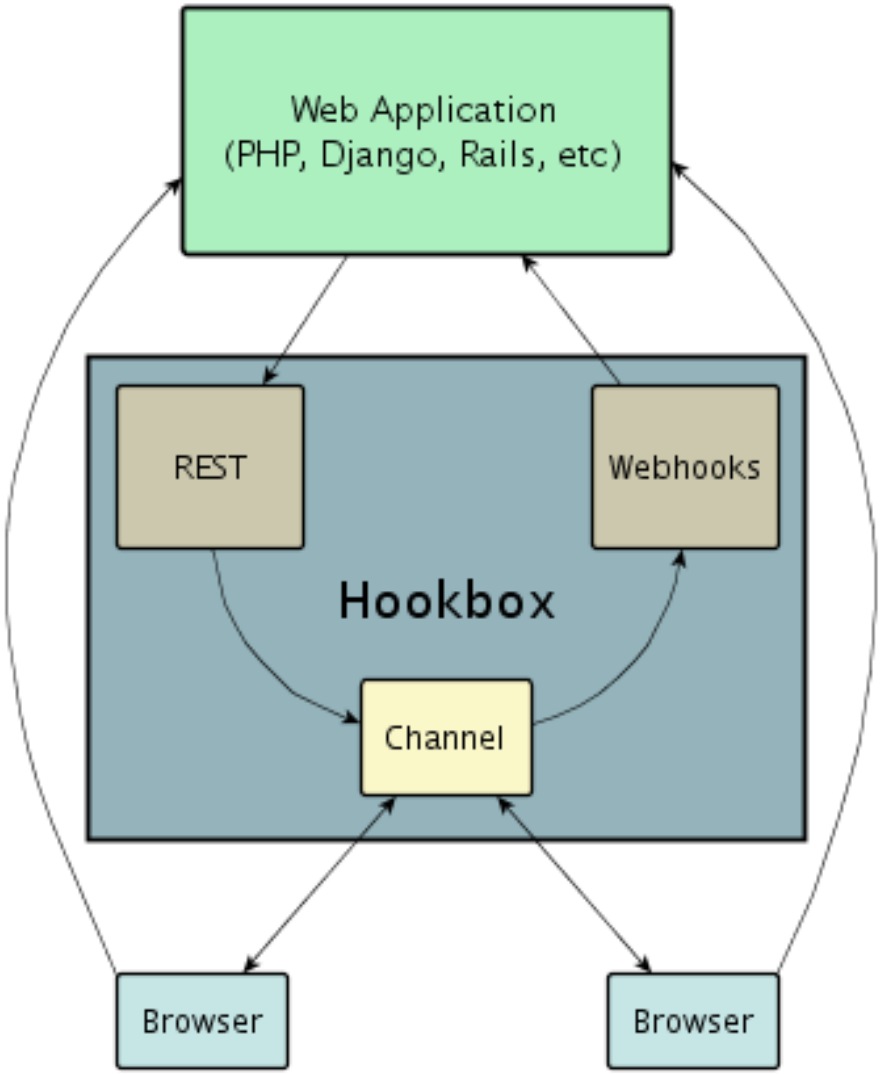
Protocoles

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

Orbited

- « Un socket TCP dans le navigateur »
- Comet (long-polling, streaming), Server–Side Events *et* WebSockets
- supporte directement IRC, XMPP et STOMP
- écrit en Python, licence BSD

Hookbox



- Serveur COMET pour framework web existantes
- Canaux nommés (pas Bayeux)
- écrit en Python, license BSD

Source :
`http://hookbox.org/docs`

Serveurs

Nom	*Protocole*	*Technique*	*écrit en*
Jetty	Bayeux	LP WS	Java
Meteor	[propre]	LP STR SSE	Perl
Ajax Push Engine	[propre]	LP STR SSE WS	C
Faye	Bayeux	LP WS	JavaScript, Ruby
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

Conclusion

- « Une galaxie en plein mouvement »
- Il faut chercher selon la situation :
  - inboard / outboard
  - canaux nommés, sockets ou au plus simple ?
  - interoperabilité souhaitée

Liens

Article Wikipedia . . . . . http://en.wikipedia.org/wiki/Comet_(programming)
AJAX Design Patterns : « 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`

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

- Firefox + Firebug
- Chromium + Web Developer Tools