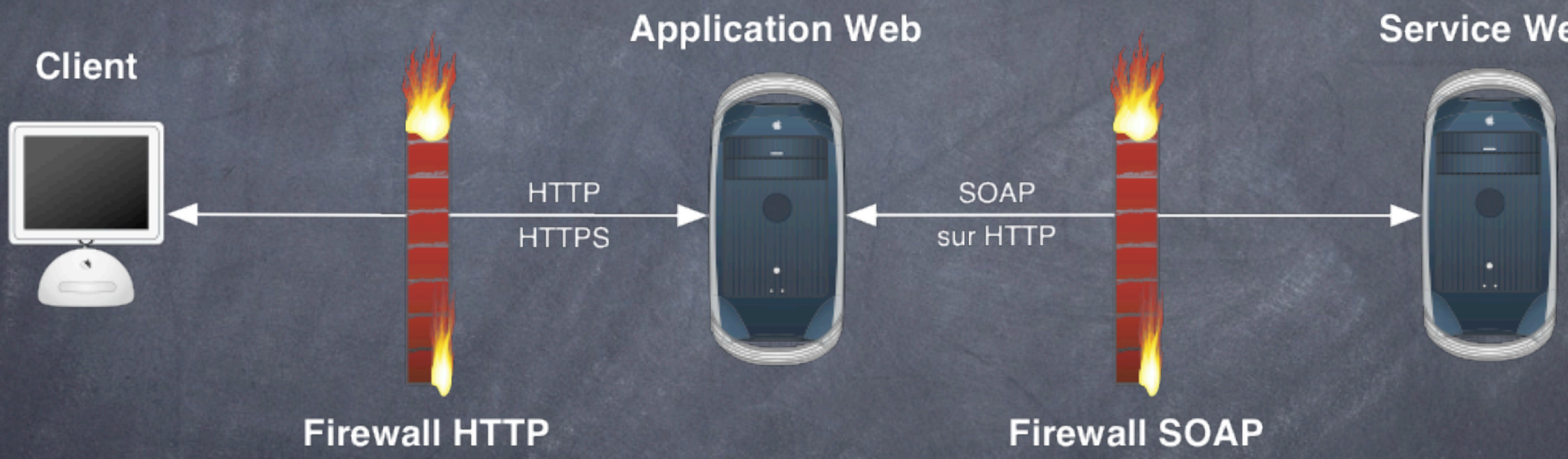
Sécurité des applications Web

Développement d'un firewall applicatif HTTP

Par Sylvain Tissot

Sécurité des applications Web

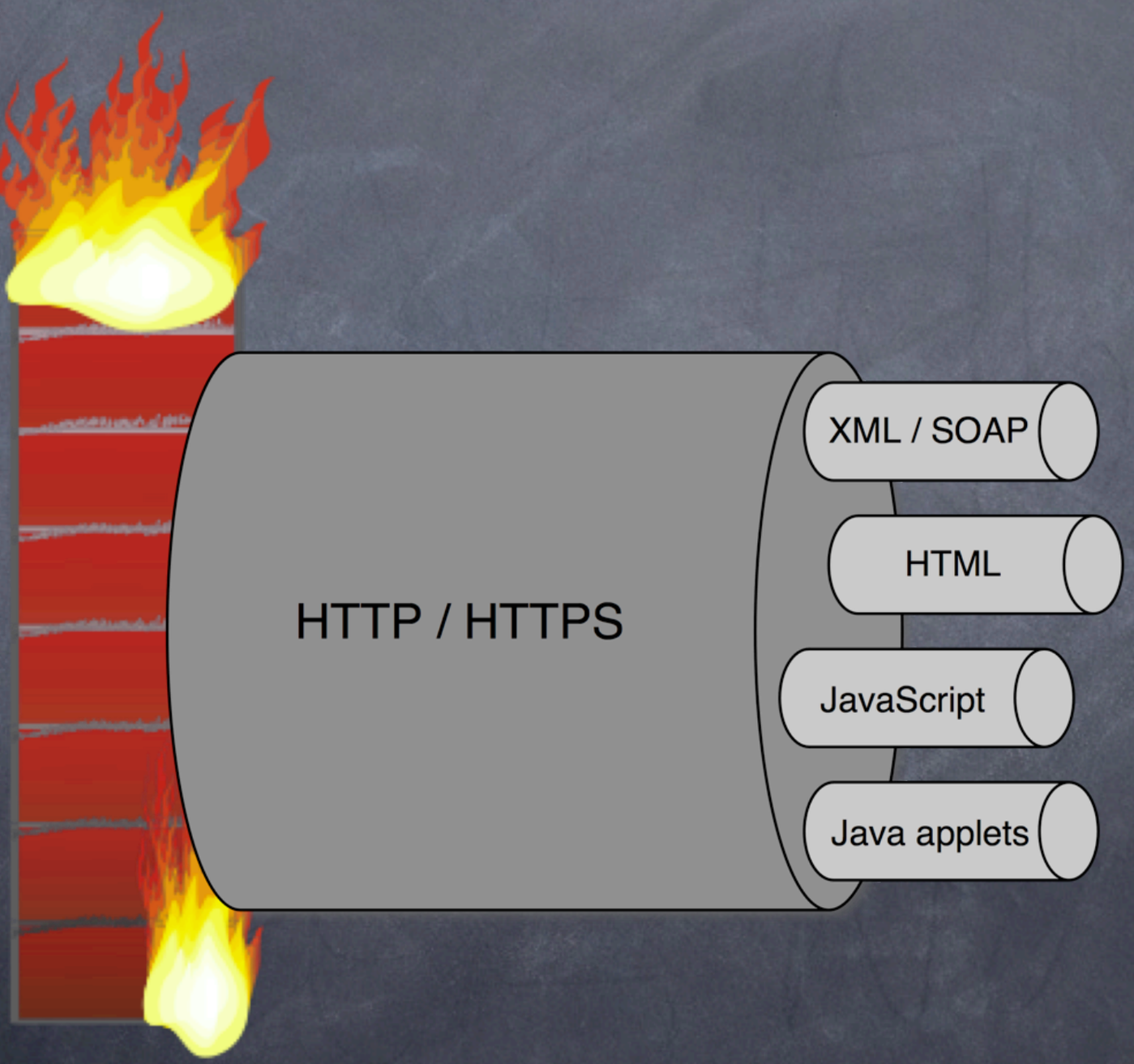


Plan de la présentation

- Introduction
  - Vulnérabilités
  - Aperçu du protocole HTTP
- Étude de Sanctum AppShield
- Développement de ProxyFilter
  - Caractéristiques
  - Syntaxe de configuration
  - Améliorations futures
- Conclusion
- Démo
- Questions

Introduction

Firewall IP classique



3 niveaux de sécurité

Application Web

Serveur Web

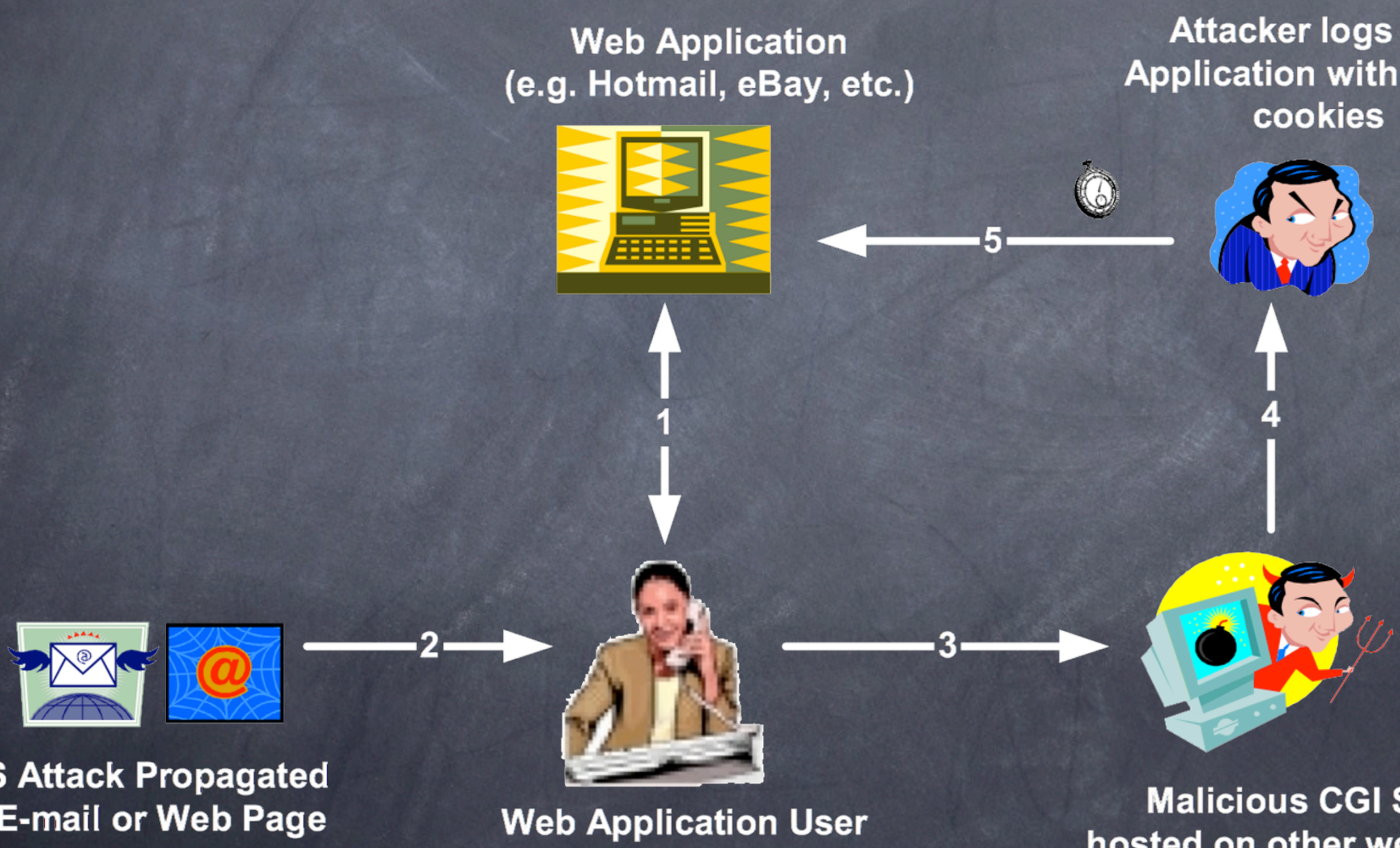
Système d'exploitation

Vulnérabilités

Cross Site Scripting (XSS)

- Attaque visant le client d'une application Web mais utilisant une vulnérabilité de celle-ci
- Absence de validation des données reçues du client avant de les retourner dans une page
- Injection de code mobile dans l'URL d'un lien
- Permet au hacker de voler la clé de session (cookie) de sa victime

Cross Site Scripting (XSS)



SQL Injection

- Utilisation des entrées de l'utilisateur pour composer une requête SQL sans les valider
- Possible de détourner une requête SELECT pour qu'elle retourne d'autres éléments que prévus
- Possible de détourner une requête INSERT ou UPDATE pour modifier des données arbitrairement dans la base

SQL Injection

```
<form action=”login.php”>
      <input type=”text” name=”username”>
      <input type=”text” name=”password”>
</form>
```

```
<?php
mysql_connect();
$query=”select * from user where password='$password' \
        and username='$username'”;
$result = mysql_query($query);
if  (mysql_num_rows($result) > 1) {return true;}
return false;
?>
```

SQL Injection

User : toto' or '1'=1

Password : anything

```
select * from user where
 password='anything' and
username='toto' or '1'='1';
```

Hidden field tampering

```
<form action="shopping_cart.cgi">
    <input type="hidden" name="amount" value="50$">
    <input type="submit" value="Commander">
</form>
```

Informations sensibles pouvant être manipulées par le client

Cookie poisoning

```
HTTP/1.1 200 OK
Date: Mon, 15 Dec 2003 23:48:34 GMT
Server: Apache/1.3.27 (Darwin) PHP/4.3.2 mod_perl/1.26
Cache-Control: max-age=60
Expires: Mon, 15 Dec 2003 23:49:34 GMT
Last-Modified: Fri, 04 May 2001 00:00:38 GMT
ETag: "26206-5b0-3af1f126"
Accept-Ranges: bytes
Set-Cookie: userid=toto; path=/
Content-Length: 1456
Content-Type: text/html
```

Cookie trivial à deviner / manipuler

Protocole HTTP

Protocole HTTP

- Essentiellement utilisé pour le Web depuis 1994
- Repose sur TCP/IP
- Protocole de type requête / réponse
- N'offre pas de notion de session
- Simple, facile à implémenter
- Versions 1.0 (RFC 1945) et 1.1 (RFC 2616)

Uniform Ressource Locator (URL)

http://example.com/path/to/file?param=value#top

Uniform Ressource Locator (URL)

http://example.com/path/to/file?param=value#top

scheme

Uniform Ressource Locator (URL)

http://example.com/path/to/file?param=value#top

nom du serveur

Uniform Ressource Locator (URL)

http://example.com/path/to/file?param=value#top

chemin d'accès (path)

Uniform Ressource Locator (URL)

http://example.com/path/to/file?param=value#top

paramètres GET

Uniform Ressource Locator (URL)

http://example.com/path/to/file?param=value#top

fragment

Requête HTTP

```
GET /index.html HTTP/1.1 ¬
Host: www.example.com ¬
Accept: */* ¬
Accept-Language: fr ¬
User-Agent: Mozilla/5.0 ¬
Connection: Keep-Alive ¬
¬
```

Requête HTTP

```
GET /index.html HTTP/1.1 ¬
Host: www.example.com ¬
Accept: */* ¬
Accept-Language: fr ¬
User-Agent: Mozilla/5.0 ¬
Connection: Keep-Alive ¬
¬
```

Ligne de requête

Requête HTTP

```
GET /index.html HTTP/1.1 ¬
Host: www.example.com ¬
Accept: */* ¬
Accept-Language: fr ¬
User-Agent: Mozilla/5.0 ¬
Connection: Keep-Alive ¬
¬
```

méthode HTTP

Requête HTTP

```
GET /index.html HTTP/1.1 ¬
Host: www.example.com ¬
Accept: */* ¬
Accept-Language: fr ¬
User-Agent: Mozilla/5.0 ¬
Connection: Keep-Alive ¬
¬
¬
```

URI du document

Requête HTTP

```
GET /index.html HTTP/1.1 ¬
Host: www.example.com ¬
Accept: */* ¬
Accept-Language: fr ¬
User-Agent: Mozilla/5.0 ¬
Connection: Keep-Alive ¬
¬
```

Version du protocole

Requête HTTP

```
GET /index.html HTTP/1.1 ¬
Host: www.example.com ¬
Accept: */* ¬
Accept-Language: fr ¬
User-Agent: Mozilla/5.0 ¬
Connection: Keep-Alive ¬
¬
¬
```

Lignes d'entêtes

Requête HTTP

```
GET /index.html HTTP/1.1 ¬
Host: www.example.com ¬
Accept: */* ¬
Accept-Language: fr ¬
User-Agent: Mozilla/5.0 ¬
Connection: Keep-Alive ¬
¬
```

Nom de l'hôte virtuel

Requête HTTP

```
GET /index.html HTTP/1.1 ¬
Host: www.example.com ¬
Accept: */* ¬
Accept-Language: fr ¬
User-Agent: Mozilla/5.0 ¬
Connection: Keep-Alive ¬
¬
¬
```

Types MIME supportés par le client

Requête HTTP

```
GET /index.html HTTP/1.1 ¬
Host: www.example.com ¬
Accept: */* ¬
Accept-Language: fr ¬
User-Agent: Mozilla/5.0 ¬
Connection: Keep-Alive ¬
¬
```

Langues préférées du client

Requête HTTP

```
GET /index.html HTTP/1.1 ¬
Host: www.example.com ¬
Accept: */* ¬
Accept-Language: fr ¬
User-Agent: Mozilla/5.0 ¬
Connection: Keep-Alive ¬
¬
```

Logiciel employé par le client

Requête HTTP

```
GET /index.html HTTP/1.1 ¬
Host: www.example.com ¬
Accept: */* ¬
Accept-Language: fr ¬
User-Agent: Mozilla/5.0 ¬
Connection: Keep-Alive ¬
¬
```

État de la connexion TCP après la requête

Requête HTTP

```
GET /index.html HTTP/1.1 ¬
Host: www.example.com ¬
Accept: */* ¬
Accept-Language: fr ¬
User-Agent: Mozilla/5.0 ¬
Connection: Keep-Alive ¬
¬
```

Fin de requête (2 * CR)

Requête HTTP

```
POST /login.cgi HTTP/1.1 ¬
Host: www.example.com ¬
Accept: */* ¬
Accept-Language: fr ¬
User-Agent: Mozilla/5.0 ¬
Connection: Keep-Alive ¬
Content-Length: 38 ¬
Content-Type: application/x-www-form-urlencoded ¬
¬
user=alice&password=bob&submit=Valider
```

Requête HTTP

```
POST /login.cgi HTTP/1.1 ¬
Host: www.example.com ¬
Accept: */* ¬
Accept-Language: fr ¬
User-Agent: Mozilla/5.0 ¬
Connection: Keep-Alive ¬
Content-Length: 38 ¬
Content-Type: application/x-www-form-urlencoded ¬
¬
user=alice&password=bob&submit=Valider
```

Méthode POST

Requête HTTP

```
POST /login.cgi HTTP/1.1 ¬
Host: www.example.com ¬
Accept: */* ¬
Accept-Language: fr ¬
User-Agent: Mozilla/5.0 ¬
Connection: Keep-Alive ¬
Content-Length: 38 ¬
Content-Type: application/x-www-form-urlencoded ¬
¬
user=alice&password=bob&submit=Valider
```

Longueur de l'entité

Requête HTTP

```
POST /login.cgi HTTP/1.1 ¬
Host: www.example.com ¬
Accept: */* ¬
Accept-Language: fr ¬
User-Agent: Mozilla/5.0 ¬
Connection: Keep-Alive ¬
Content-Length: 38 ¬
Content-Type: application/x-www-form-urlencoded ¬
¬
user=alice&password=bob&submit=Valider
```

Encodage de l'entité

Requête HTTP

```
POST /login.cgi HTTP/1.1 ¬
Host: www.example.com ¬
Accept: */* ¬
Accept-Language: fr ¬
User-Agent: Mozilla/5.0 ¬
Connection: Keep-Alive ¬
Content-Length: 38 ¬
Content-Type: application/x-www-form-urlencoded ¬
¬
user=alice&password=bob&submit=Valider
```

Entité (contenu POST)

Réponse HTTP

```
HTTP/1.1 200 OK
Date: Mon, 15 Dec 2003 23:48:34 GMT
Server: Apache/1.3.27 (Darwin) PHP/4.3.2 mod_perl/1.26
Cache-Control: max-age=60
Expires: Mon, 15 Dec 2003 23:49:34 GMT
Last-Modified: Fri, 04 May 2001 00:00:38 GMT
ETag: "26206-5b0-3af1f126"
Accept-Ranges: bytes
Content-Length: 1456
Content-Type: text/html

<html>
<head>
  <title>Sample page</title>
</head>
...
```

Réponse HTTP

```
HTTP/1.1 200 OK
Date: Mon, 15 Dec 2003 23:48:34 GMT
Server: Apache/1.3.27 (Darwin) PHP/4.3.2 mod_perl/1.26
Cache-Control: max-age=60
Expires: Mon, 15 Dec 2003 23:49:34 GMT
Last-Modified: Fri, 04 May 2001 00:00:38 GMT
ETag: "26206-5b0-3af1f126"
Accept-Ranges: bytes
Content-Length: 1456
Content-Type: text/html

<html>
<head>
  <title>Sample page</title>
</head>
...
```

Ligne de réponse

Réponse HTTP

```
HTTP/1.1 200 OK
Date: Mon, 15 Dec 2003 23:48:34 GMT
Server: Apache/1.3.27 (Darwin) PHP/4.3.2 mod_perl/1.26
Cache-Control: max-age=60
Expires: Mon, 15 Dec 2003 23:49:34 GMT
Last-Modified: Fri, 04 May 2001 00:00:38 GMT
ETag: "26206-5b0-3af1f126"
Accept-Ranges: bytes
Content-Length: 1456
Content-Type: text/html

<html>
<head>
  <title>Sample page</title>
</head>
...
```

Version du protocole

Réponse HTTP

```
HTTP/1.1 200 OK
Date: Mon, 15 Dec 2003 23:48:34 GMT
Server: Apache/1.3.27 (Darwin) PHP/4.3.2 mod_perl/1.26
Cache-Control: max-age=60
Expires: Mon, 15 Dec 2003 23:49:34 GMT
Last-Modified: Fri, 04 May 2001 00:00:38 GMT
ETag: "26206-5b0-3af1f126"
Accept-Ranges: bytes
Content-Length: 1456
Content-Type: text/html

<html>
<head>
  <title>Sample page</title>
</head>
...
```

Code de statut

Réponse HTTP

```
HTTP/1.1 200 OK
Date: Mon, 15 Dec 2003 23:48:34 GMT
Server: Apache/1.3.27 (Darwin) PHP/4.3.2 mod_perl/1.26
Cache-Control: max-age=60
Expires: Mon, 15 Dec 2003 23:49:34 GMT
Last-Modified: Fri, 04 May 2001 00:00:38 GMT
ETag: "26206-5b0-3af1f126"
Accept-Ranges: bytes
Content-Length: 1456
Content-Type: text/html

<html>
<head>
  <title>Sample page</title>
</head>
...
```

Lignes d'entêtes de la réponse

Réponse HTTP

```
HTTP/1.1 200 OK
Date: Mon, 15 Dec 2003 23:48:34 GMT
Server: Apache/1.3.27 (Darwin) PHP/4.3.2 mod_perl/1.26
Cache-Control: max-age=60
Expires: Mon, 15 Dec 2003 23:49:34 GMT
Last-Modified: Fri, 04 May 2001 00:00:38 GMT
ETag: "26206-5b0-3af1f126"
Accept-Ranges: bytes
Content-Length: 1456
Content-Type: text/html

<html>
<head>
  <title>Sample page</title>
</head>
...
```

Nom du logiciel serveur

Réponse HTTP

```
HTTP/1.1 200 OK
Date: Mon, 15 Dec 2003 23:48:34 GMT
Server: Apache/1.3.27 (Darwin) PHP/4.3.2 mod_perl/1.26
Cache-Control: max-age=60
Expires: Mon, 15 Dec 2003 23:49:34 GMT
Last-Modified: Fri, 04 May 2001 00:00:38 GMT
ETag: "26206-5b0-3af1f126"
Accept-Ranges: bytes
Content-Length: 1456
Content-Type: text/html

<html>
<head>
  <title>Sample page</title>
</head>
...
```

Informations pour la gestion du cache

Réponse HTTP

```
HTTP/1.1 200 OK
Date: Mon, 15 Dec 2003 23:48:34 GMT
Server: Apache/1.3.27 (Darwin) PHP/4.3.2 mod_perl/1.26
Cache-Control: max-age=60
Expires: Mon, 15 Dec 2003 23:49:34 GMT
Last-Modified: Fri, 04 May 2001 00:00:38 GMT
ETag: "26206-5b0-3af1f126"
Accept-Ranges: bytes
Content-Length: 1456
Content-Type: text/html

<html>
<head>
  <title>Sample page</title>
</head>
...
```

Longueur et type du document retourné

Réponse HTTP

```
HTTP/1.1 200 OK
Date: Mon, 15 Dec 2003 23:48:34 GMT
Server: Apache/1.3.27 (Darwin) PHP/4.3.2 mod_perl/1.26
Cache-Control: max-age=60
Expires: Mon, 15 Dec 2003 23:49:34 GMT
Last-Modified: Fri, 04 May 2001 00:00:38 GMT
ETag: "26206-5b0-3af1f126"
Accept-Ranges: bytes
Content-Length: 1456
Content-Type: text/html

<html>
<head>
  <title>Sample page</title>
</head>
...
```

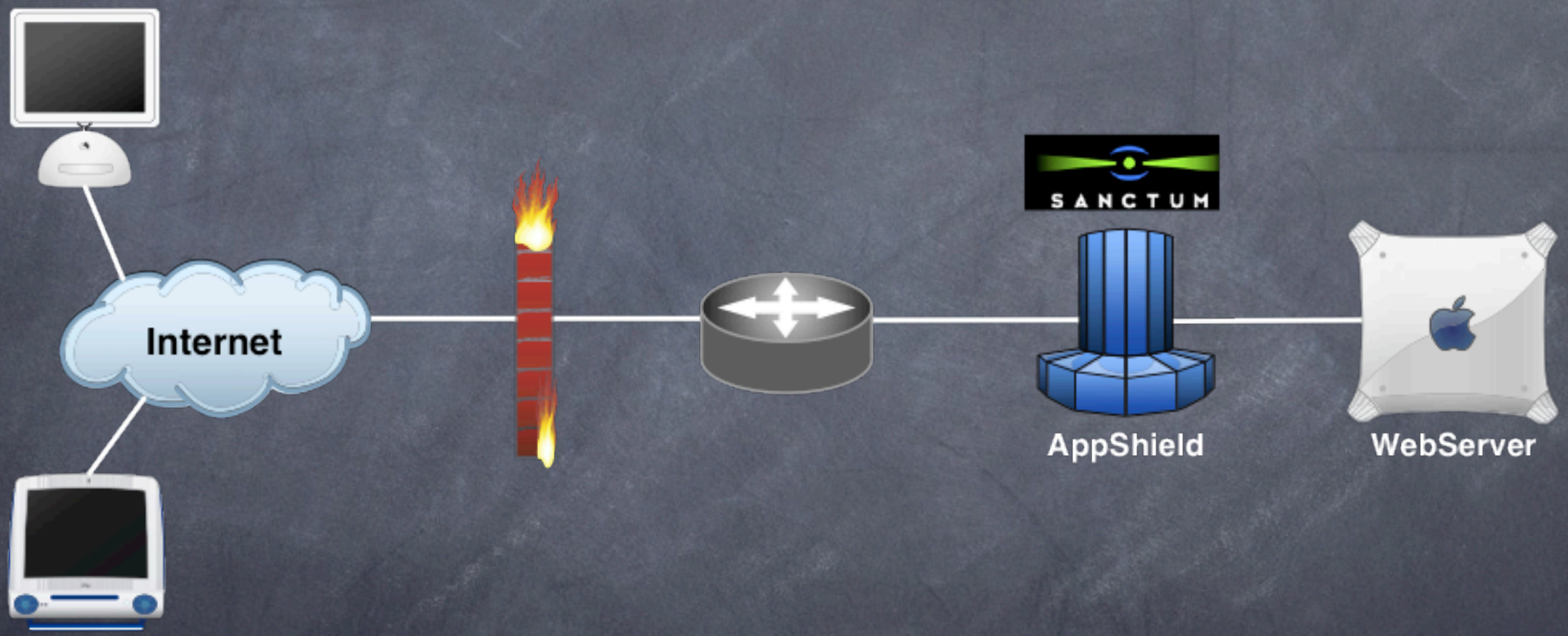
Contenu (entité) de la réponse

Sanctum AppShield

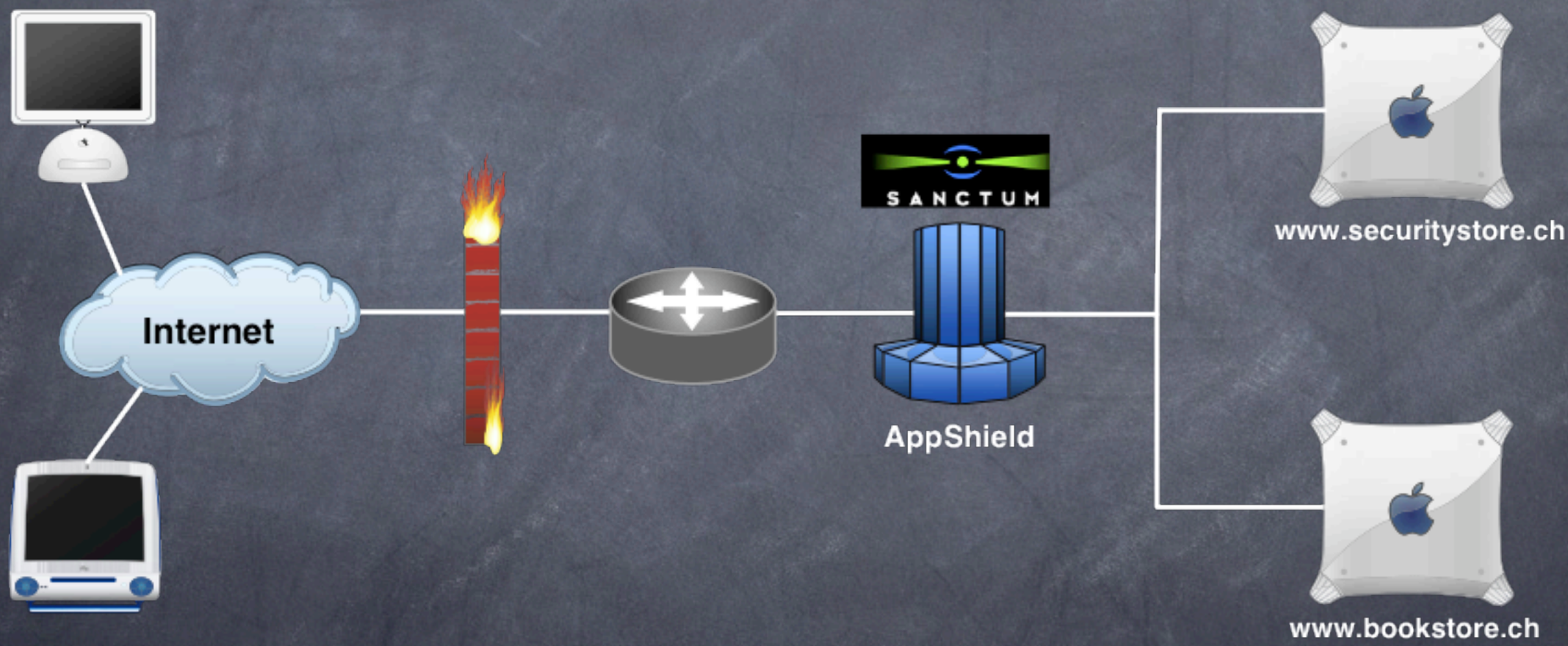
Sanctum AppShield

- Firewall applicatif HTTP commercial
- Depuis 1999 sur le marché, actuellement en version 4.0
- Plateformes Windows NT4, 2000 et Solaris 8
- Prix 15'000$ par serveur
- Firewall de type stateful, stratégie whitelist
- Supporte une variété de topologies

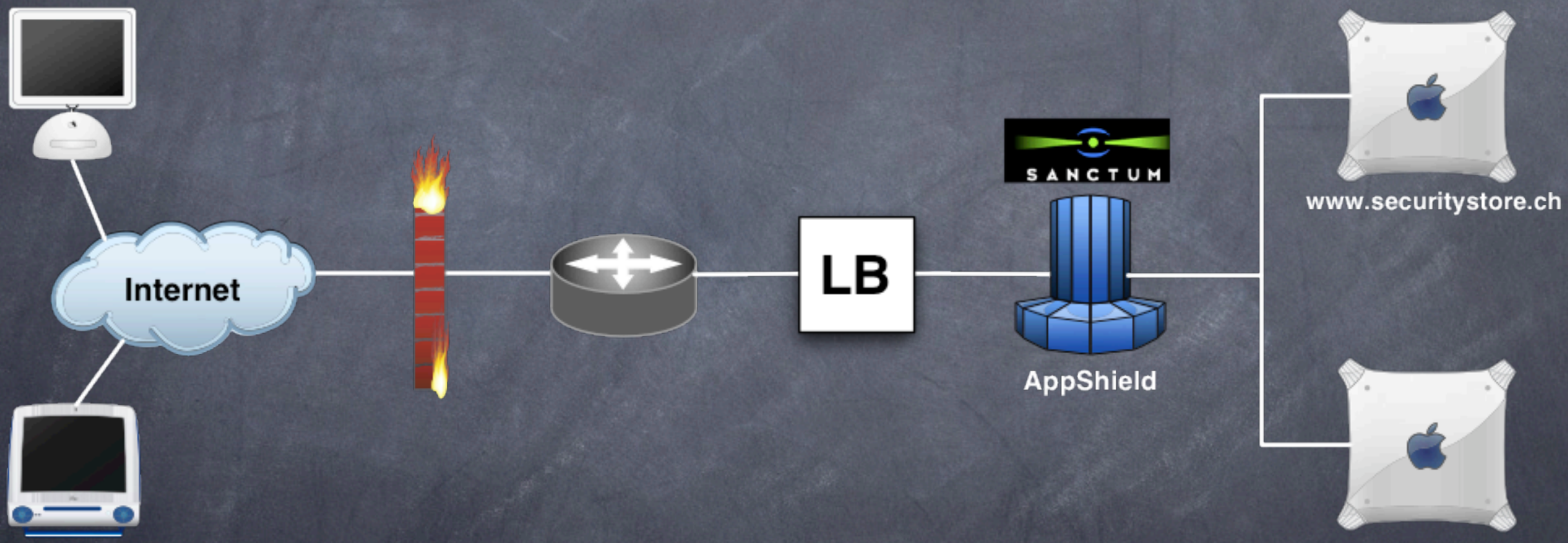
Topologie "one to one"



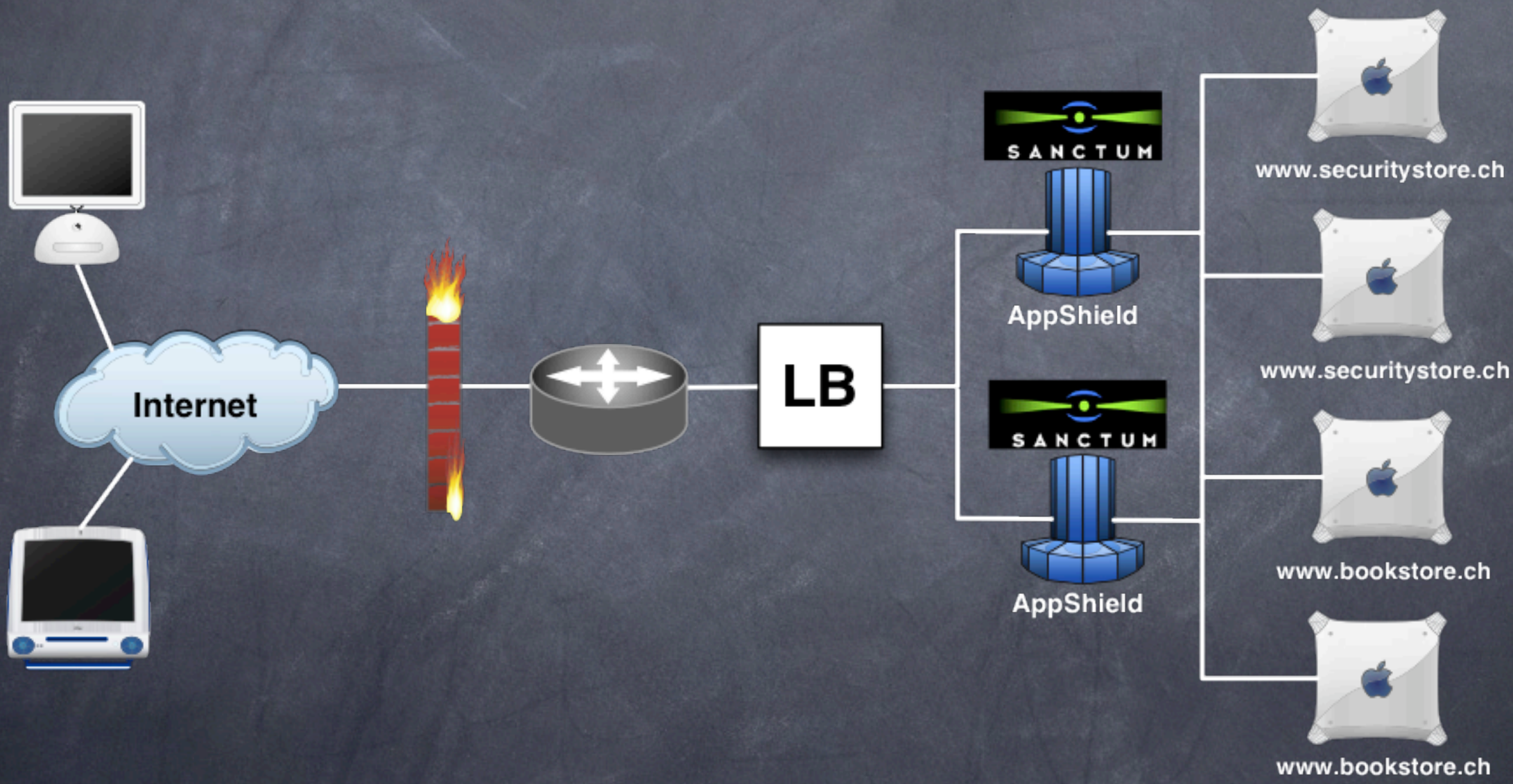
Topologie "serveurs multiples"



Topologie "serveurs miroirs"



Topologie "many to many"



Fonctionnalités

- Haute performance
- Historique détaillé et global
- Support de SSL avant et après le proxy
- Mapping d'URL basé sur le préfixe
- Interropérable (OPSEC, Check Point)
- Pages d'erreur personnalisables

Stateful

- Mémorise tous les liens du document retourné au client et génère dynamiquement des règles positives pour la requête suivante de ce client
- Garde la trace de chaque client au moyen d'un cookie et/ou de l'IP
- On peut forcer le client à visiter les pages dans un certain ordre

Watchdog

- Une tâche dédiée surveille en permanence le bon fonctionnement du noeud AppShield
- En cas de problème, alerte dans l'historique, par e-mail ou pager à l'administrateur
- Lors de détection d'une attaque, possibilité de communiquer via OPSEC avec un firewall Check Point pour bloquer l'IP de l'attaquant

Conclusion

- Haut niveau de sécurité contre la manipulation de paramètres côté client (hidden field, cookie...)
- Configuration parfois complexe de par le mode de fonctionnement Whitelist imposant la définition de règles d'exception
- Mode d'auto-apprentissage et fonction de définition de règles positives à partir d'entrées de l'historique

ProxyFilter

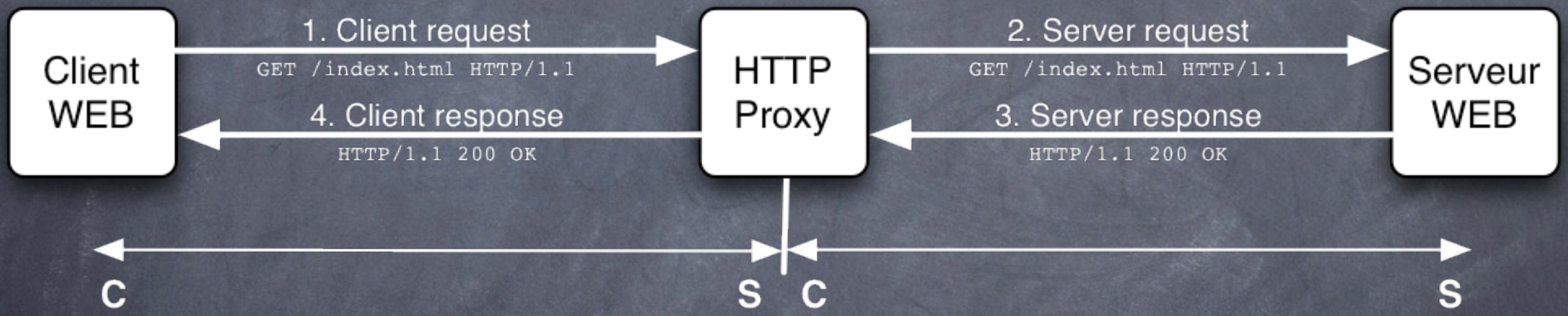
Mandat

- Développer un firewall applicatif HTTP
- Programmation en Perl ou C
- Filtrage des entêtes et du contenu HTTP, en entrée (requête) comme en sortie (réponse)
- Configuration à l'aide de fichiers texte
- Whitelist ou Blacklist
- Compatible mod_proxy et mod_rewrite
- Év. support de SSL, mode auto-apprentissage

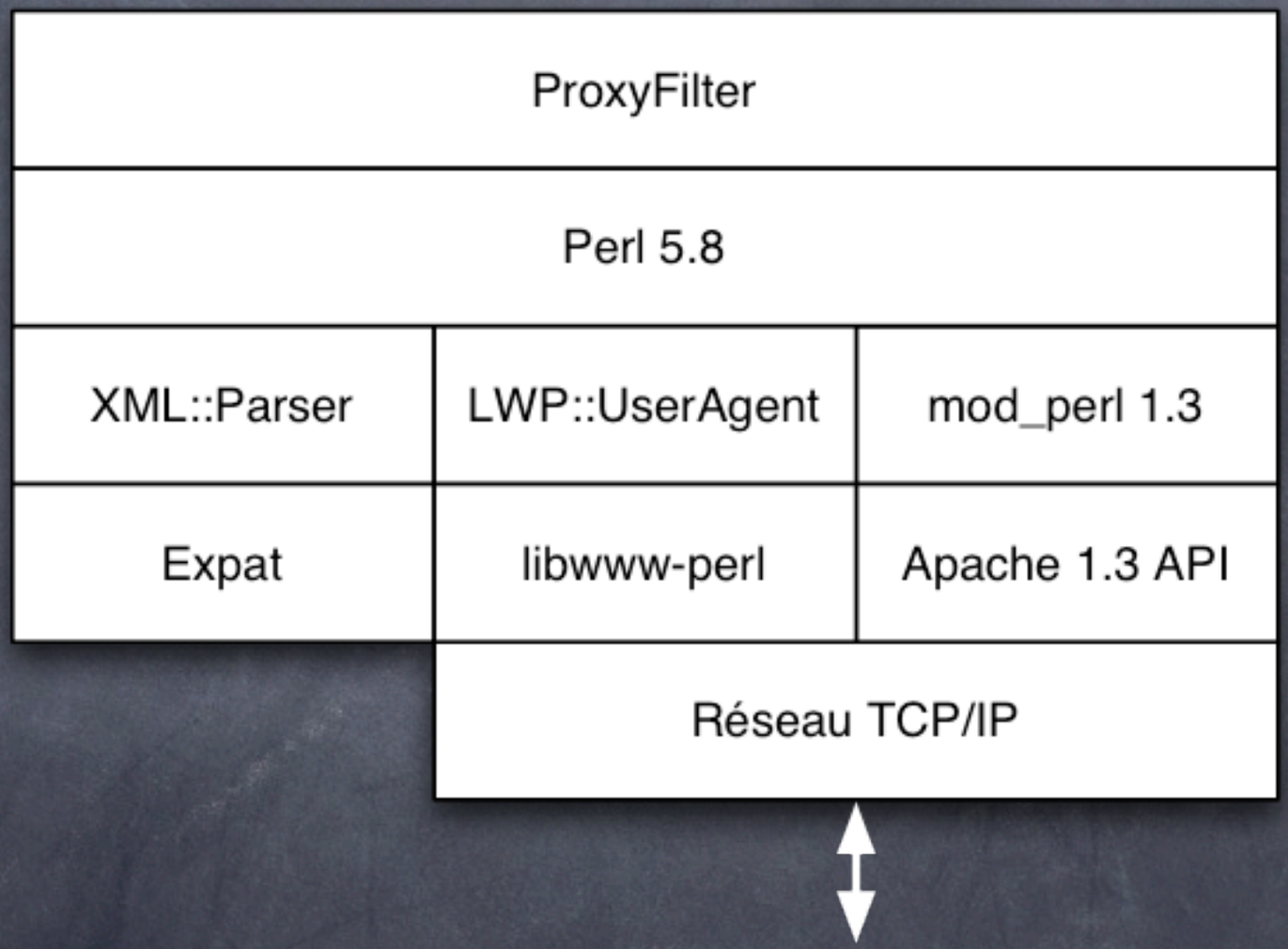
Approche

- Module Apache écrit en Perl
- Fonctionne comme un proxy inverse filtrant
- Fichiers de configuration basés sur XML
- Filtrage Whitelist et/ou Blacklist (mixte)
- Permet de réécrire et contrôler les URLs
- Permet de vérifier et filtrer les entêtes
- Permet de vérifier les paramètres GET et POST
- Log détaillé avec plusieurs niveaux de verbosité
- Supporte SSL en entrée et/ou en sortie

Fonctionnement



Architecture



Perl

- Practical Extraction and Report Language
- Développé par Larry Wall dans les années 1980
- Spécialisé dans le traitement de chaînes de caractères
- Langage de plus haut niveau que le C, plus simple (pas de gestion de la mémoire)
- Moins vulnérable aux dépassements de tampon
- Bien intégré dans Apache grâce à mod_perl

Apache

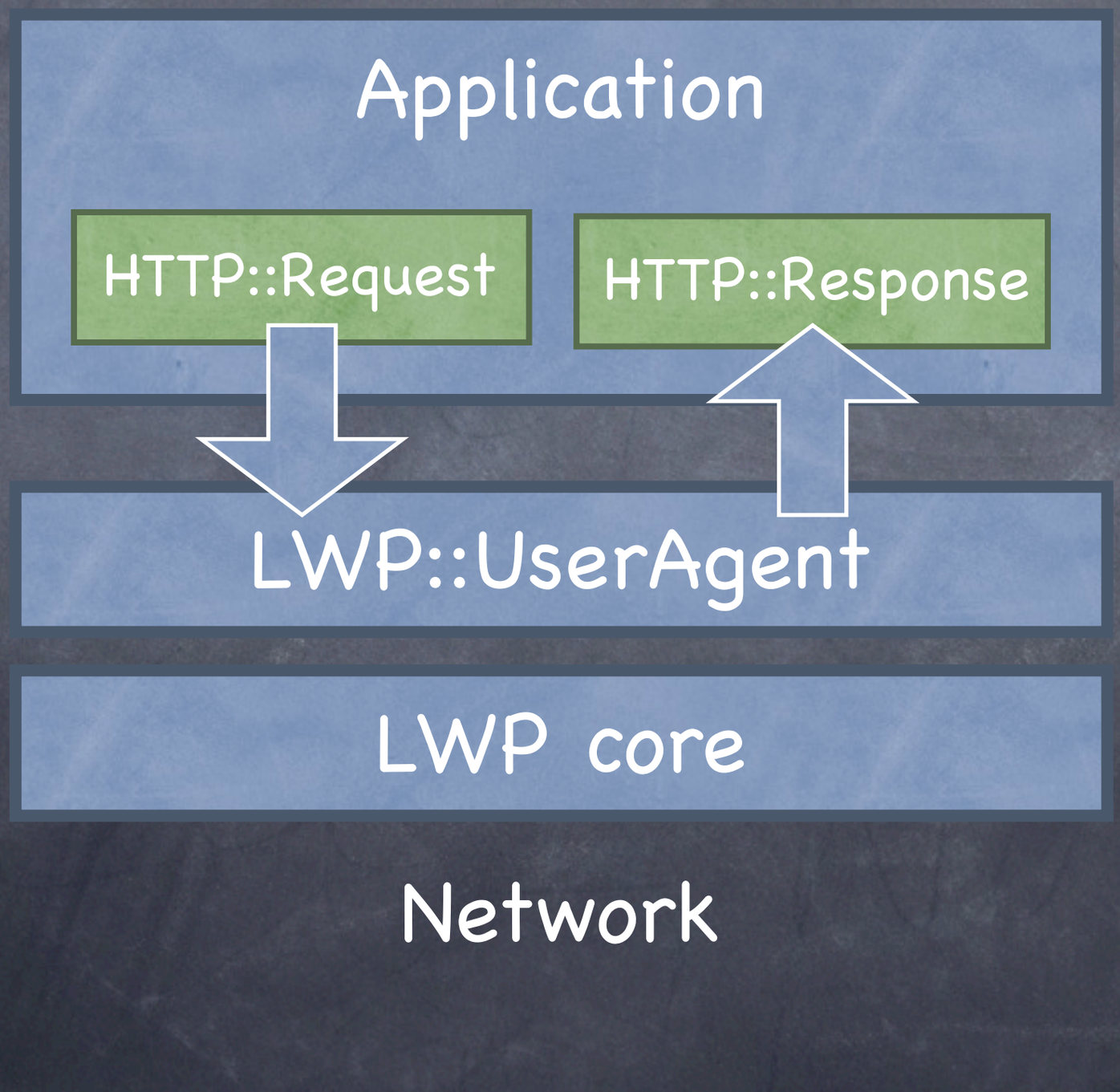
- Projet open source issu des serveurs Web du CERN et NCSA
- Plus de 50% des sites Web dans le monde
- Rapide, stable, multiplateforme, modulaire
- Très bien documenté
- Peu être étendu à l'aide de modules écrits en C ou en Perl, et de dizaines de modules existants (mod_proxy, mod_rewrite, mod_alias, etc...)

libwww-perl

- Collection de modules et paquetages offrant à Perl l'accès au protocole HTTP
- Implémente la fonction de client et proxy HTTP
- Supporte SSL avec des modules additionnels
- Utilisé par ProxyFilter pour émettre une requête interne au serveur cible

ProxyFilter

libwww-perl



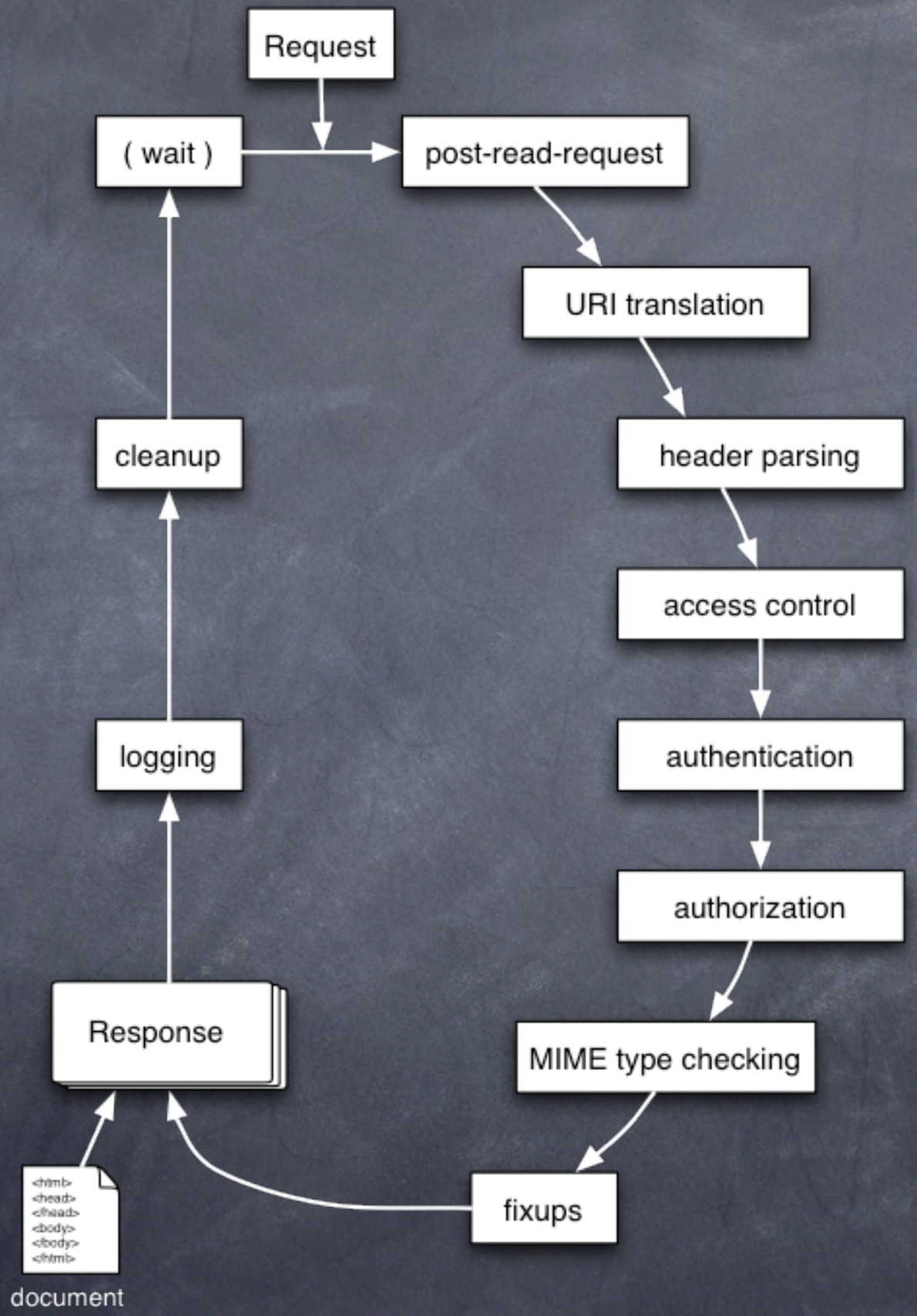
Expat

- Parser XML en mode flux écrit en C
- Très rapide
- S'interface avec Perl au travers de XML::Parser
- Utilisé par ProxyFilter pour lire et valider ses fichiers de configuration

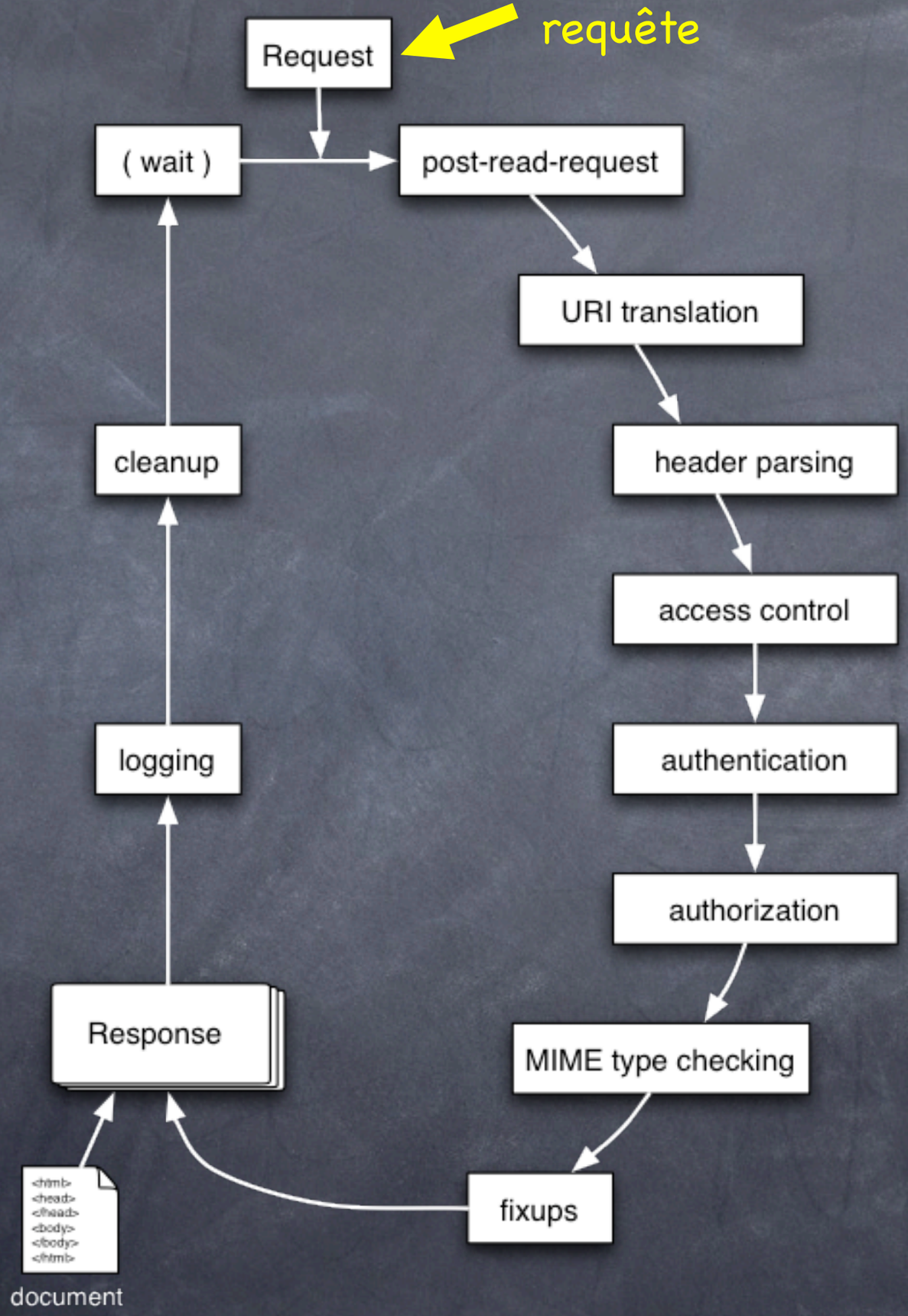
mod_perl

- Environnement d'exécution de programmes Perl dans Apache
- Peut être utilisé pour faire tourner des scripts CGI ou pour développer des modules Apache en Perl
- Lui-même module Apache écrit en C, exporte l'API Apache pour C vers des modules Perl
- Possibilité de précompiler des modules au démarrage du serveur → plus rapide

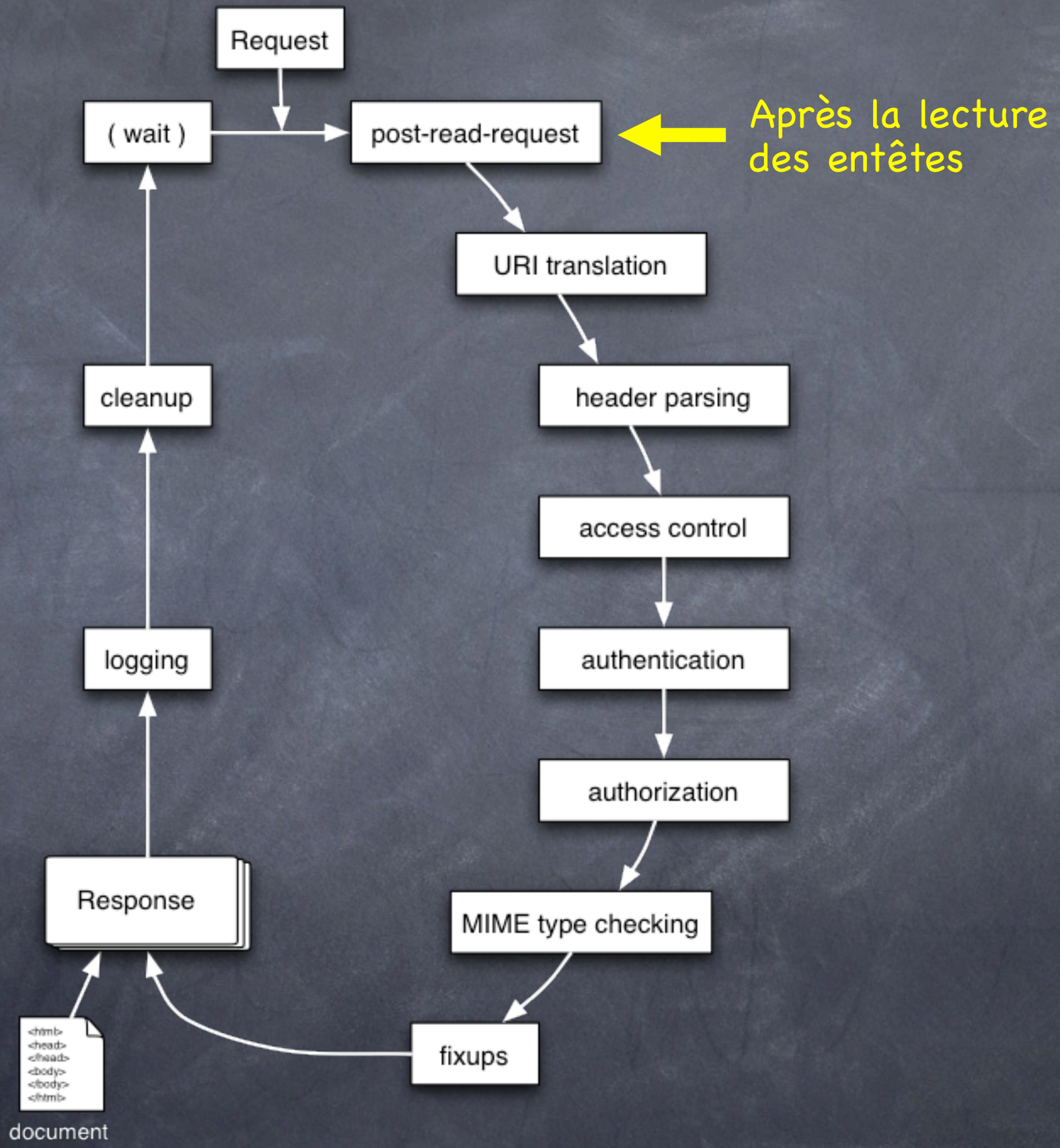
ProxyFilter



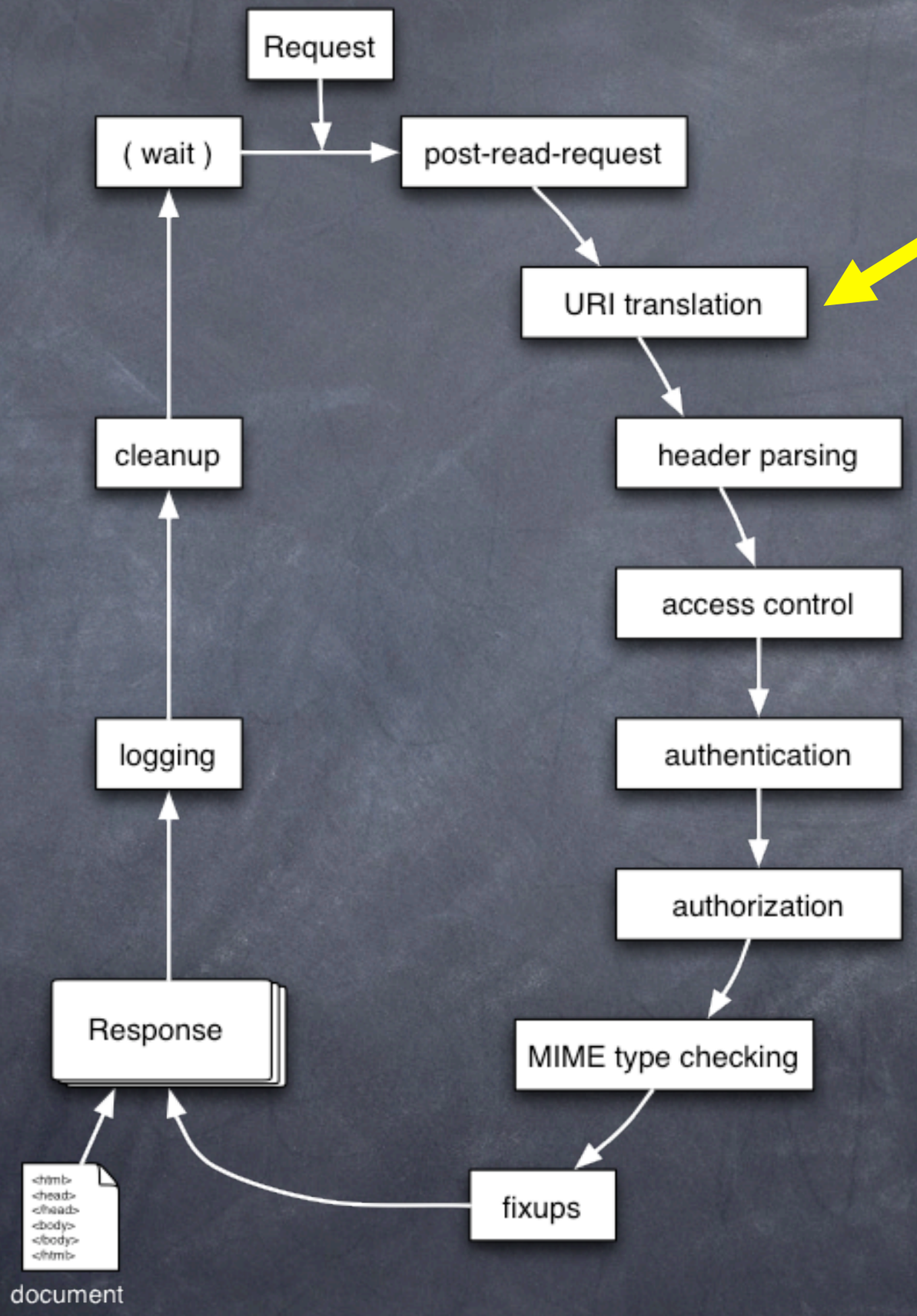
ProxyFilter



ProxyFilter

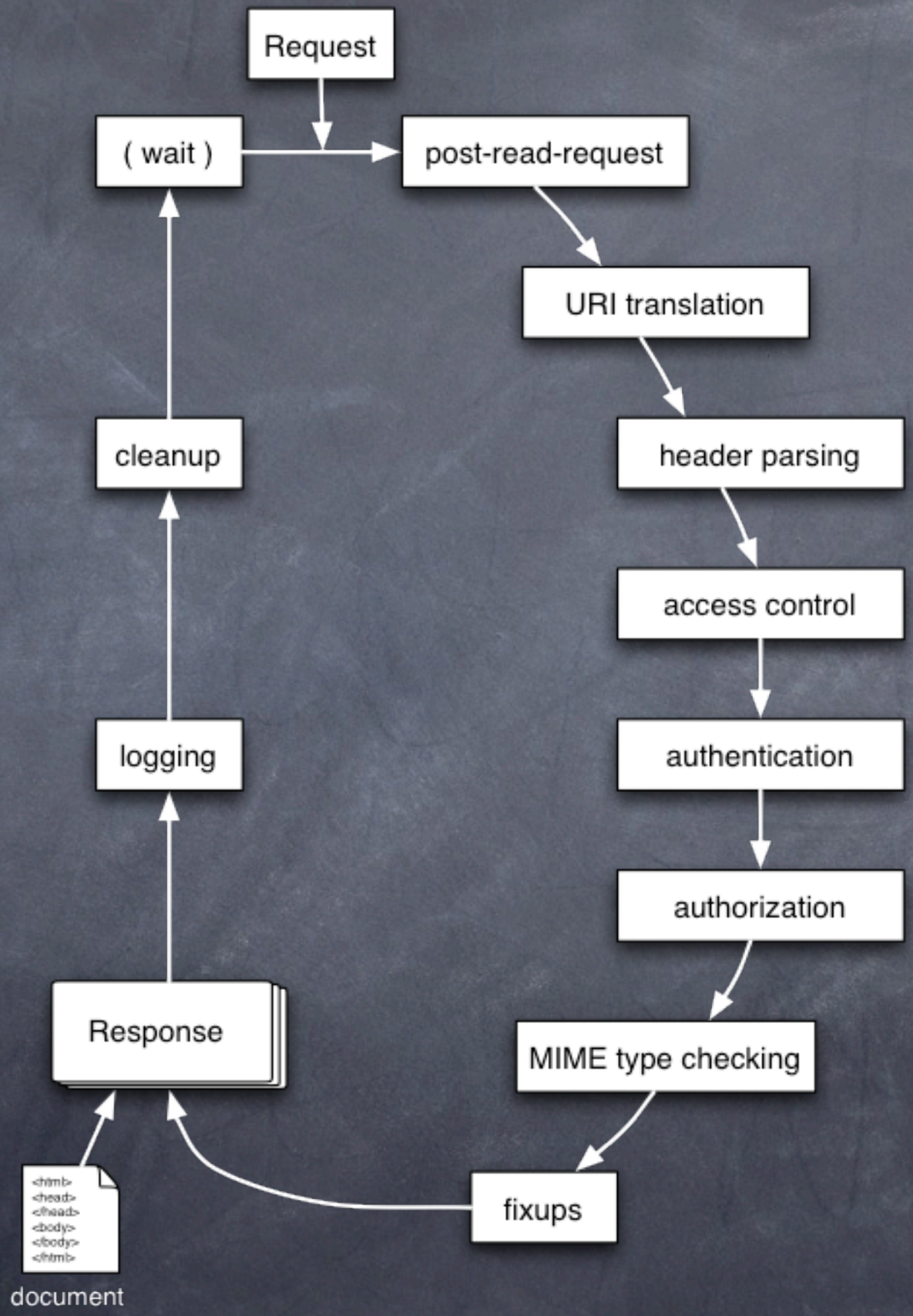


ProxyFilter

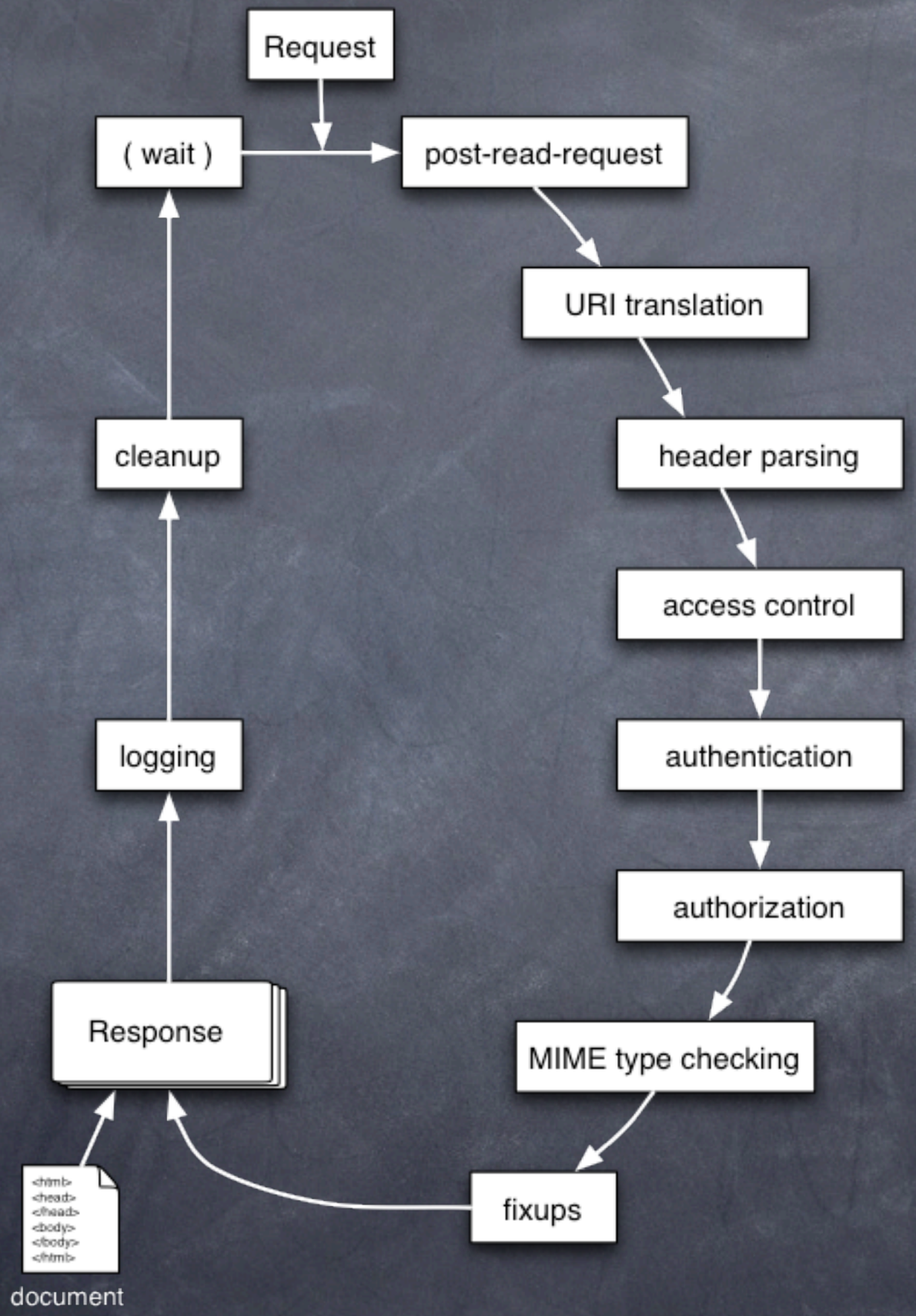


traduction de l'URI en un chemin d'accès local

ProxyFilter

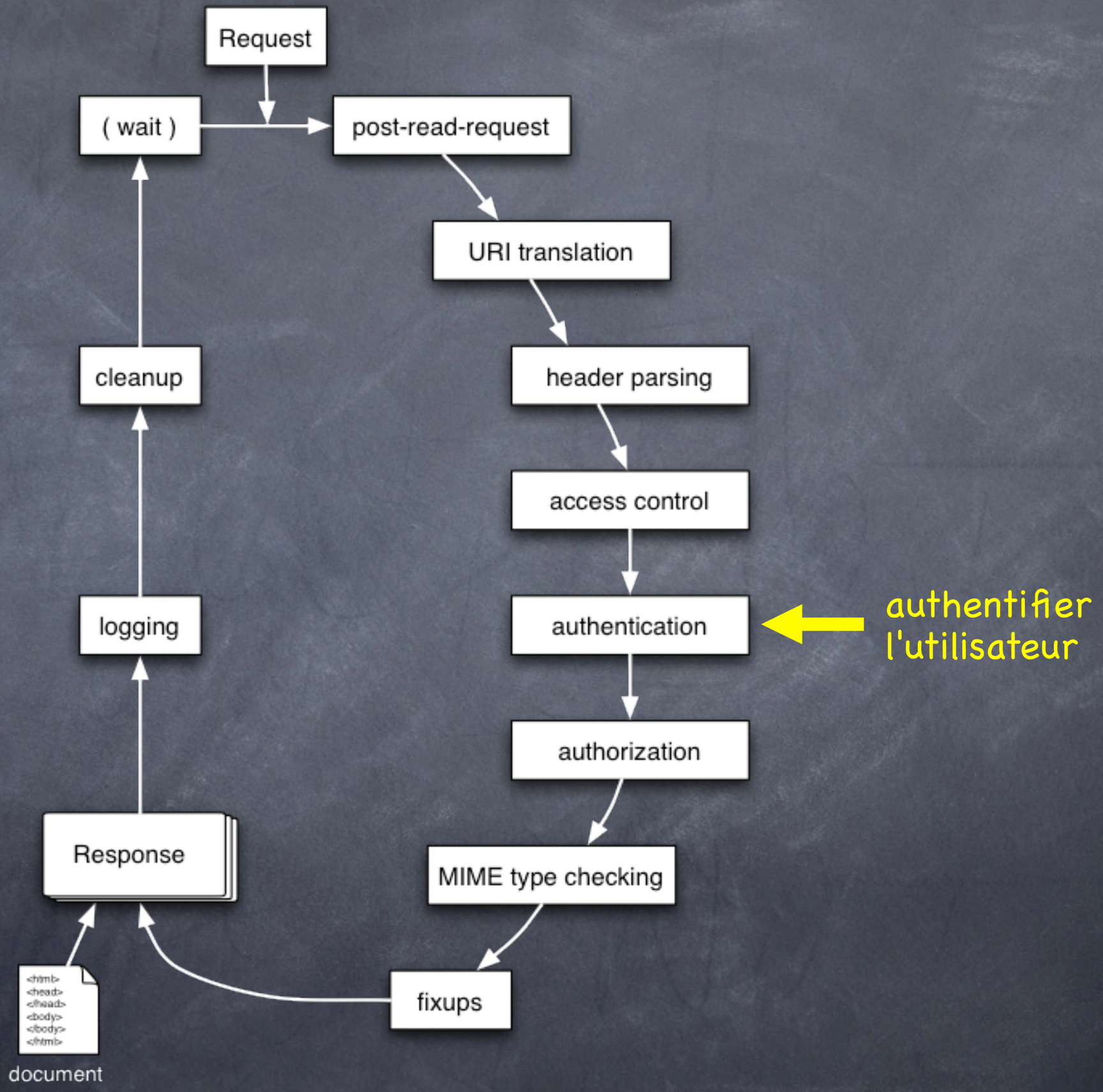


ProxyFilter

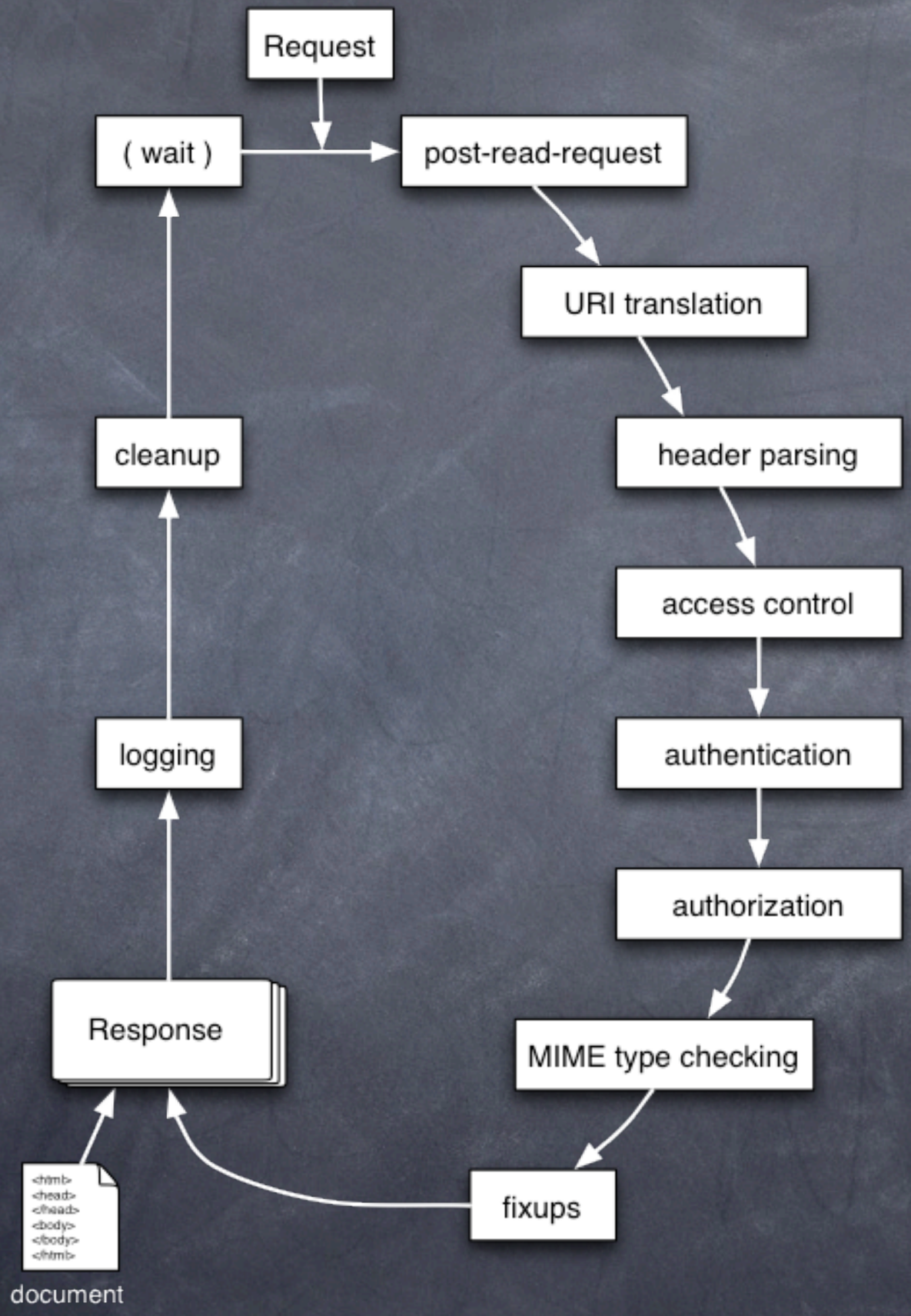


contrôle d'accès
(IP, browser, etc...)

ProxyFilter

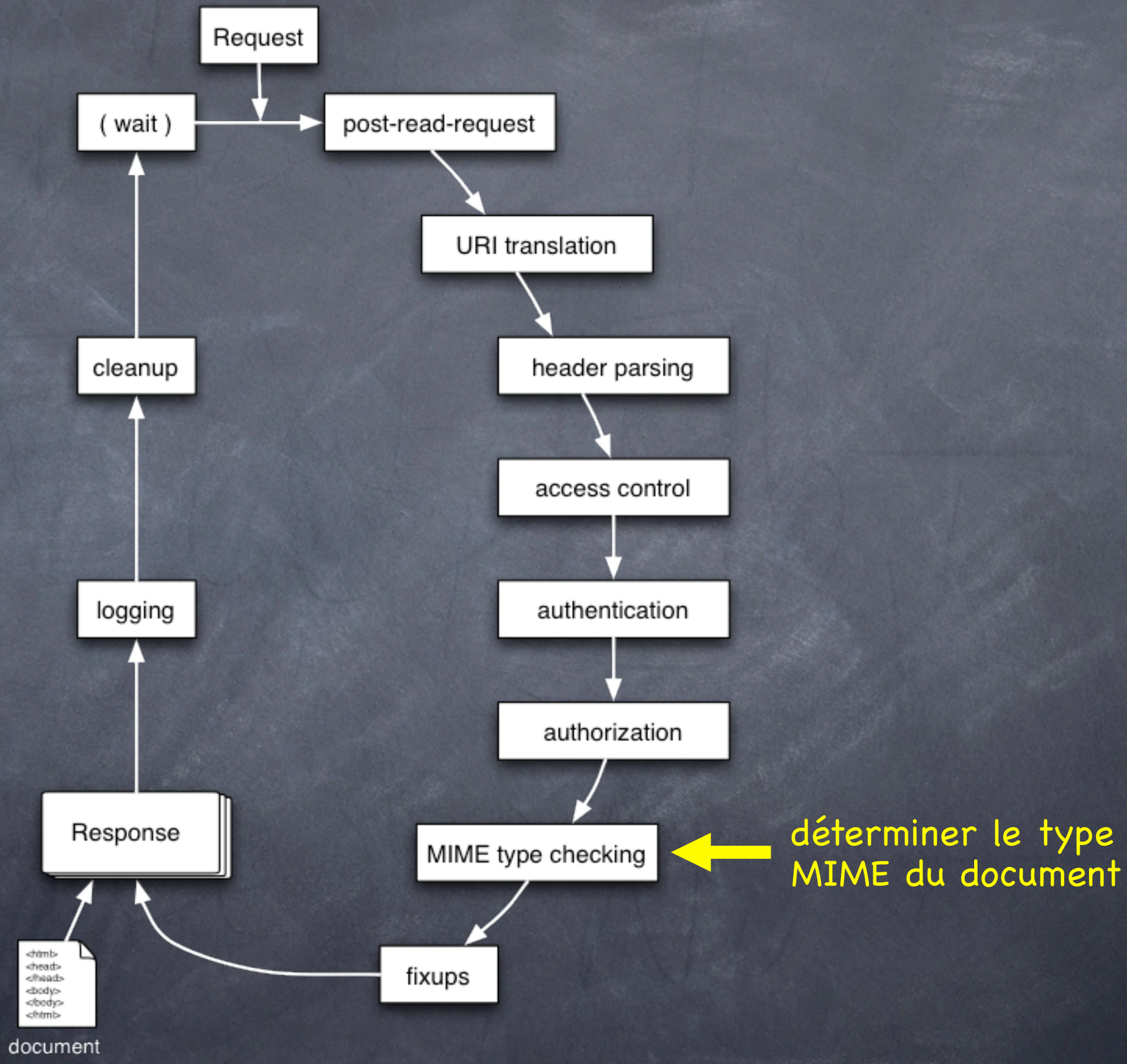


ProxyFilter

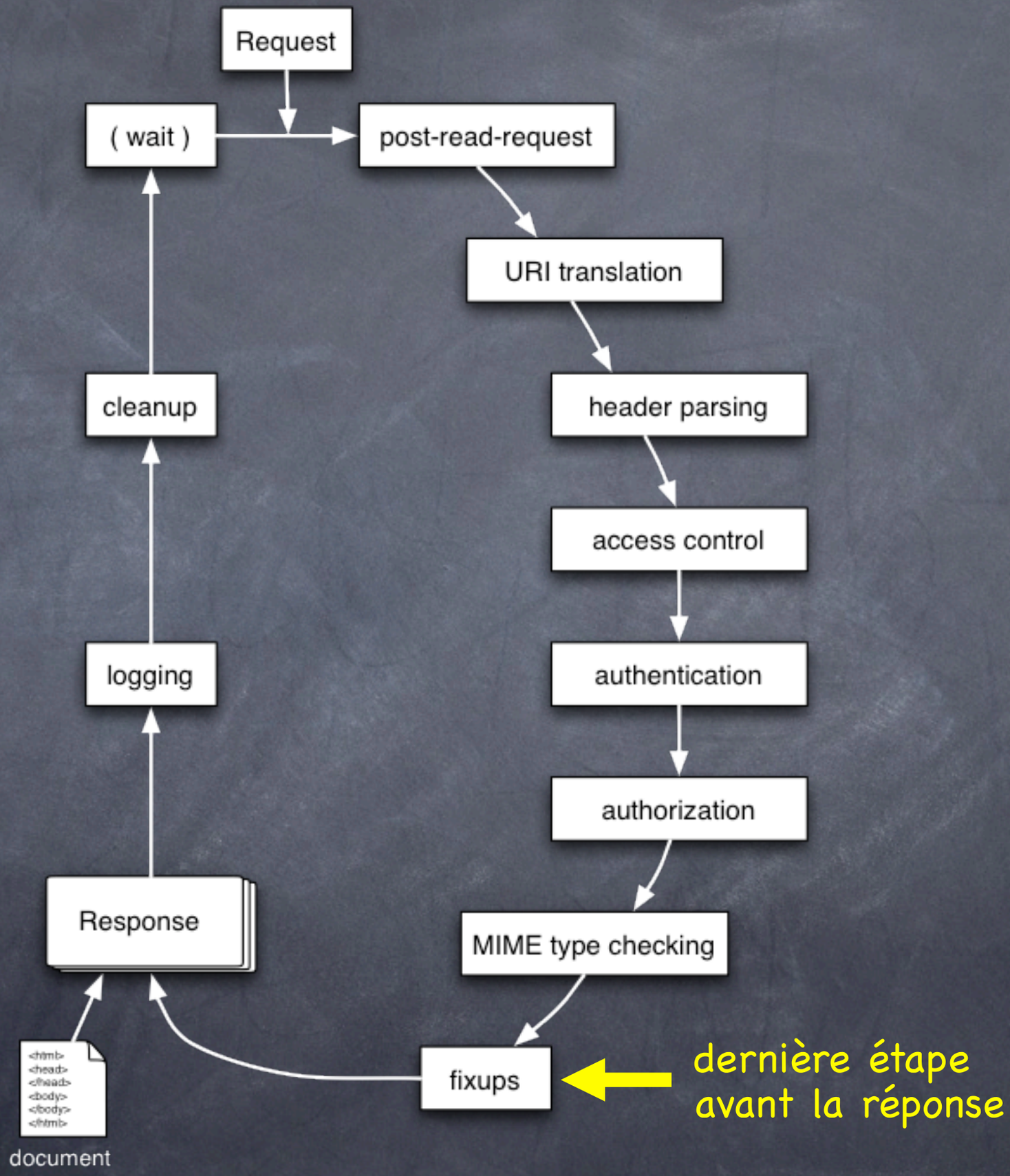


contrôle d'accès en fonction de l'utilisateur

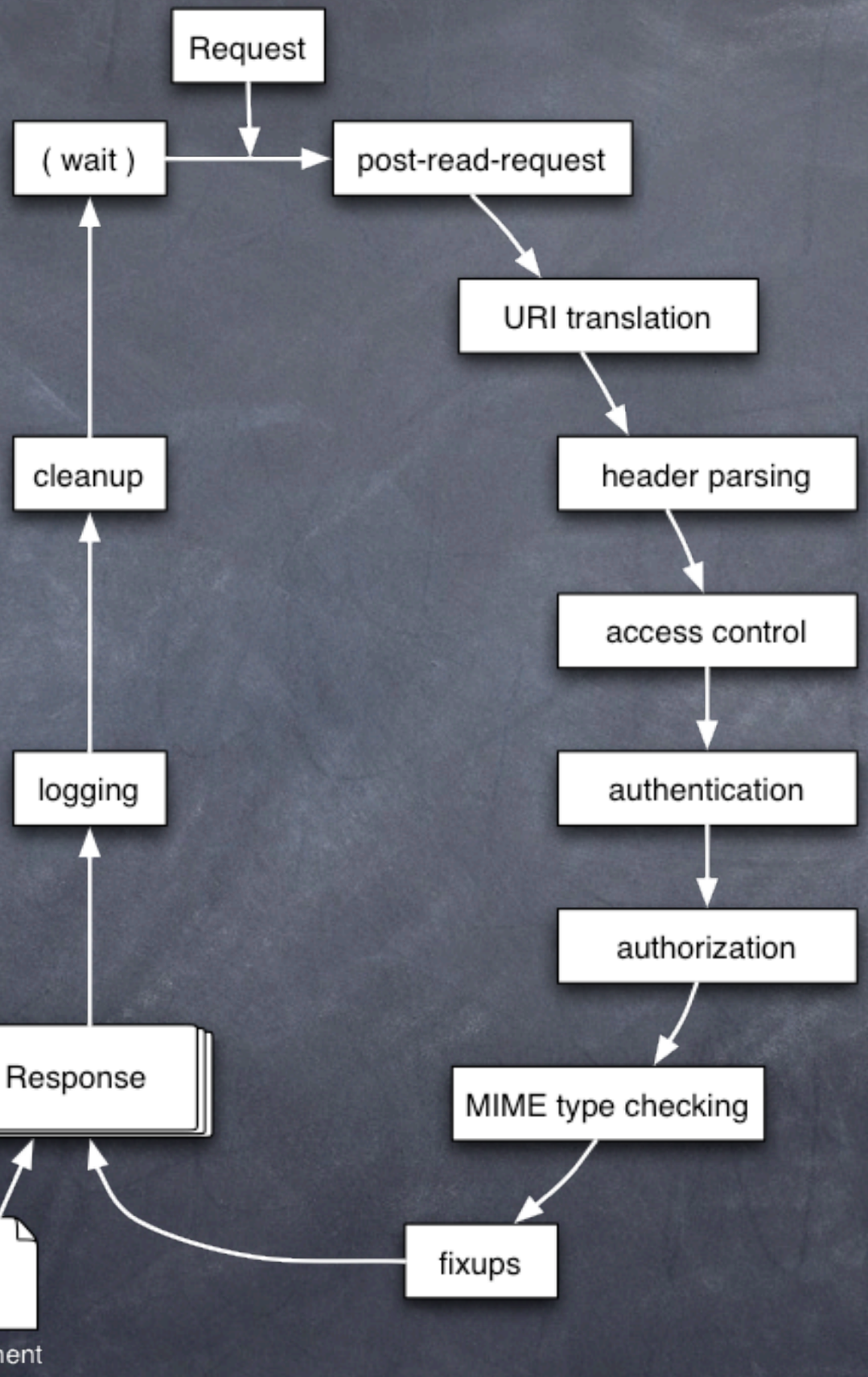
ProxyFilter



ProxyFilter

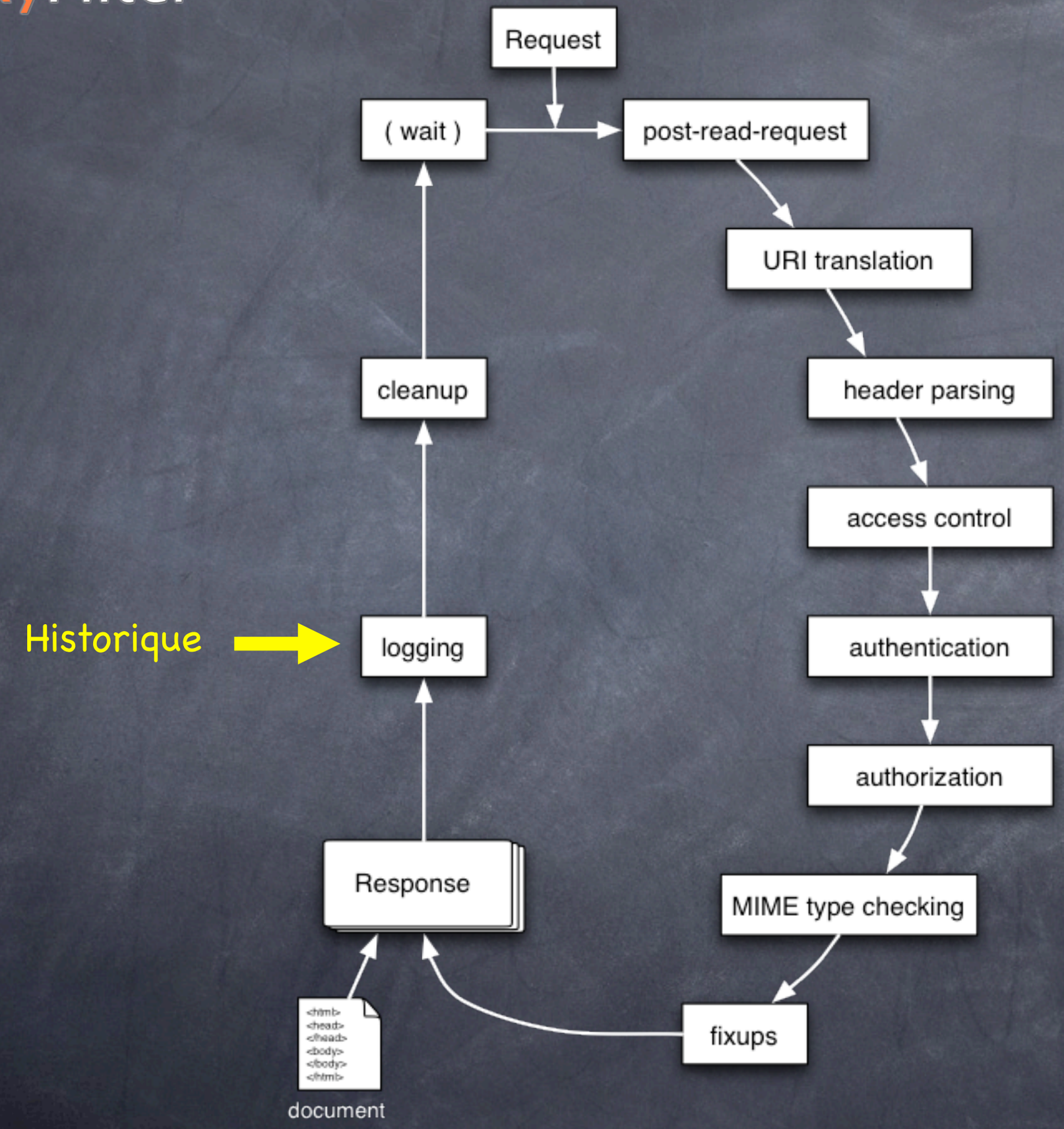


ProxyFilter



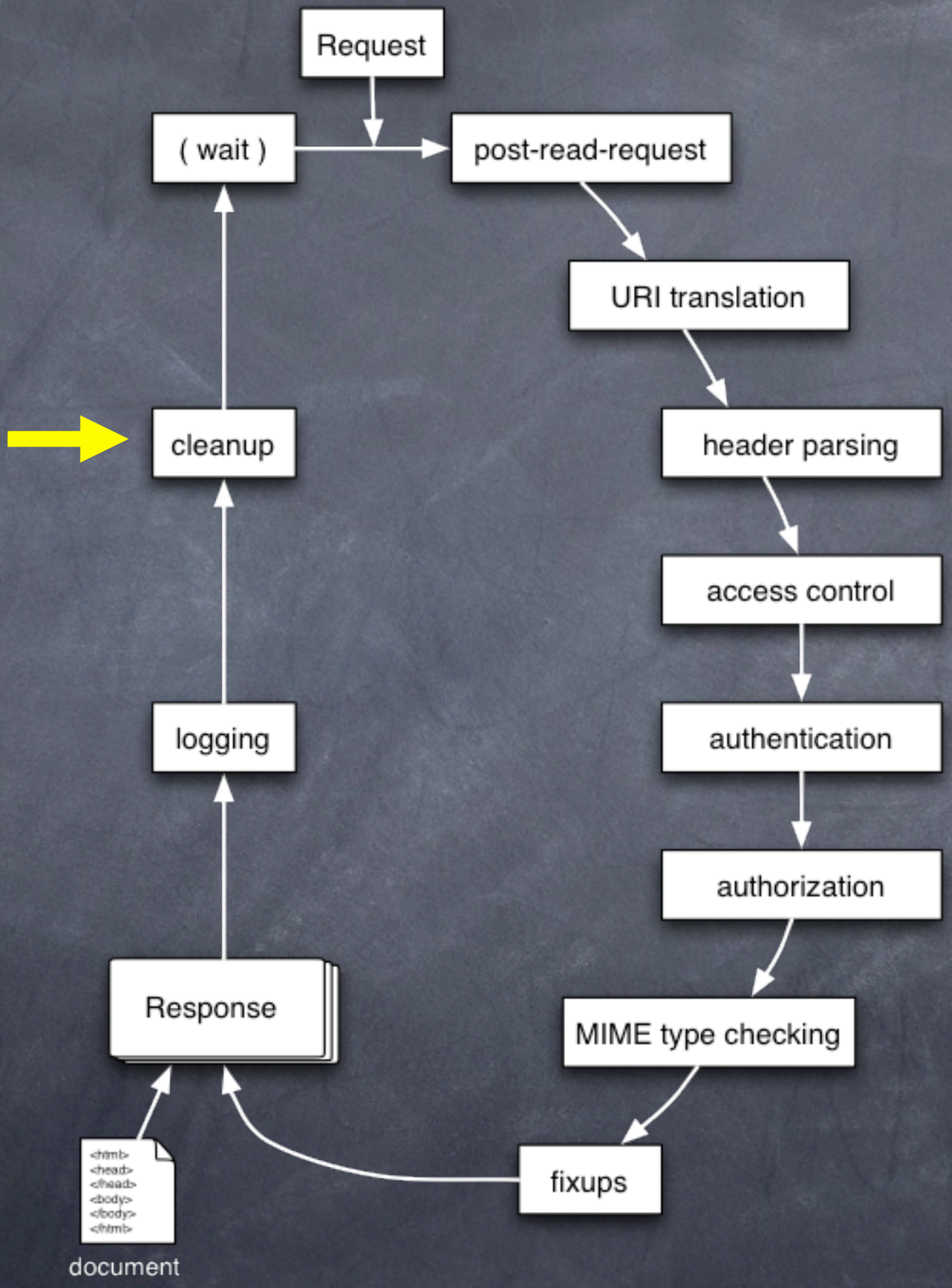
Réponse: on compose le document à retourner (CGI...)

ProxyFilter

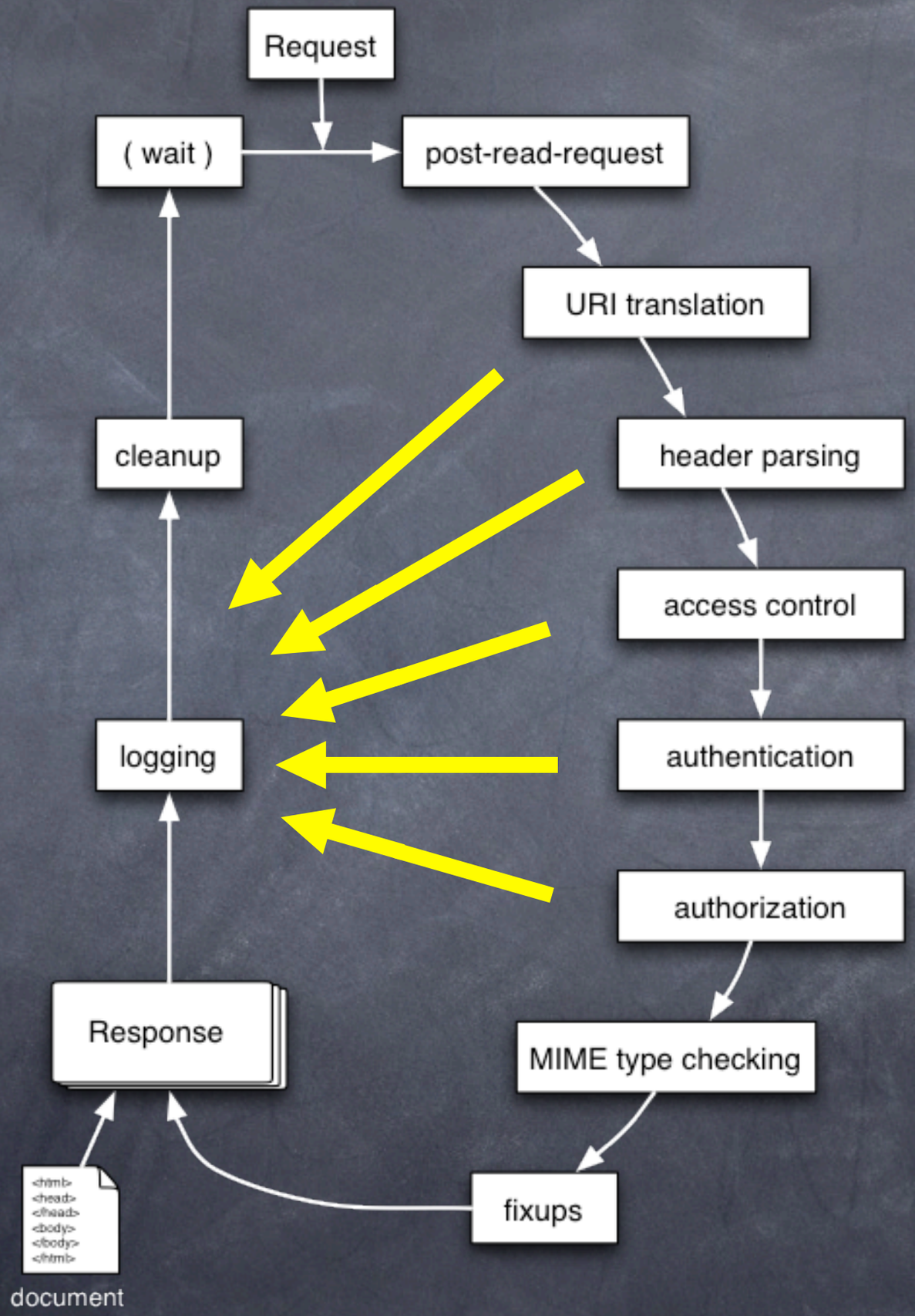


ProxyFilter

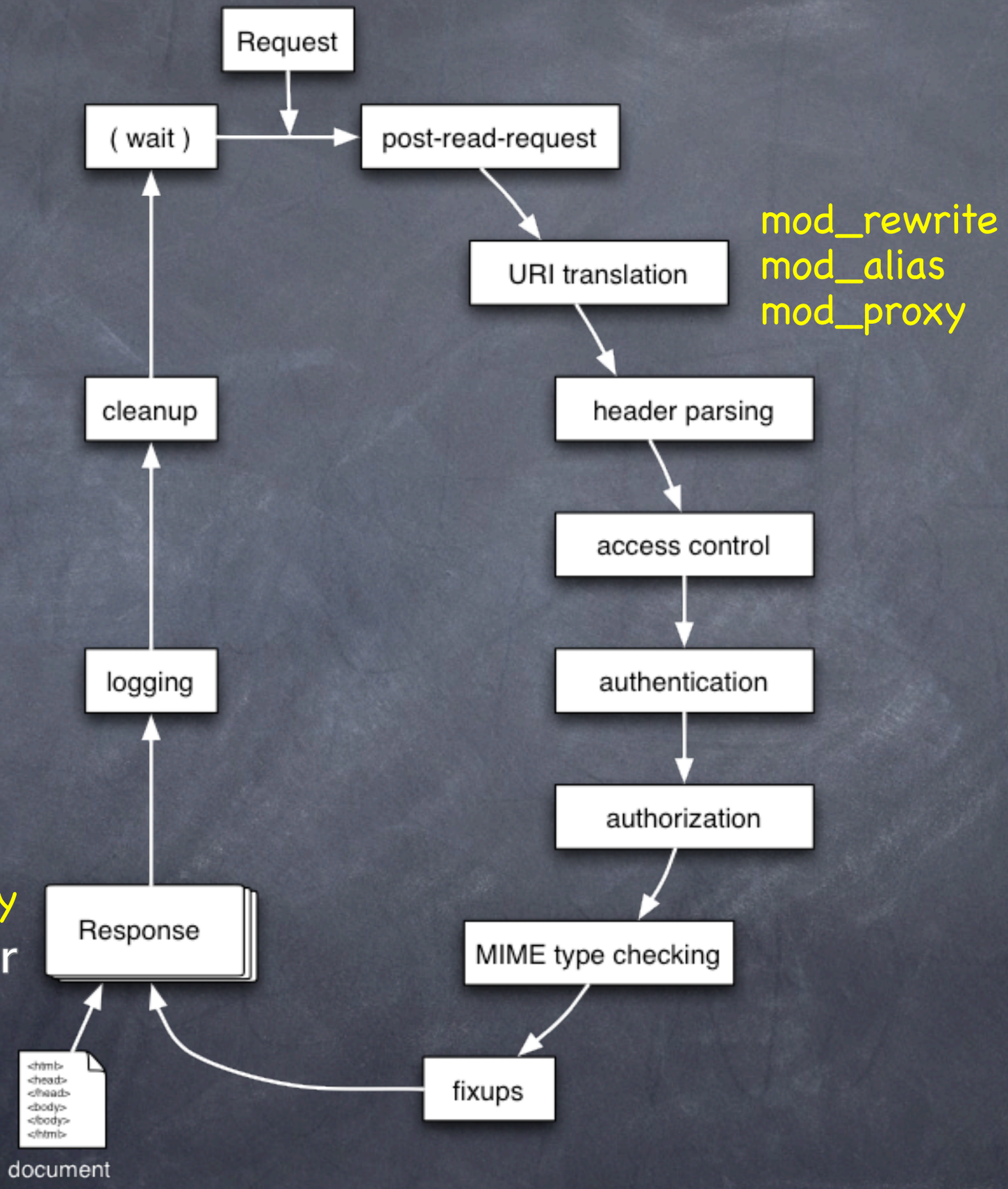
Libération de ressources mémoire



ProxyFilter



ProxyFilter



Fichiers de configuration

- proxyfilter_config.xml
- proxfilter_webapp.xml
- proxyfilter_mappings
- proxyfilter_charsets

proxyfilter_config.xml

- Directives de configuration globales
- Emplacement des autres fichiers de config
- Emplacement et verbosité de l'historique
- Filtrage des entêtes HTTP entrantes/sortantes

proxyfilter_config.xml

```
<config>
   <charsetsfile>/etc/proxyfilter_charsets</charsetsfile>
   <mappingsfile>/etc/proxyfilter_mappings</mappingsfile>
   <webappfile>/etc/proxyfilter_webapp.xml</webappfile>
   <logfile>/var/log/proxyfilter_log</logfile>
   <loglevel>info</loglevel>
   <http_methods>GET,POST,HEAD</http_methods>
   <headers_in default="deny">
      <allow>
      <header name="host" value="~^[-_a-zA-Z0-9\.@]*$" maxlength="50" />
      <header name="cookie" value="~SESSID=%alphanum%" maxlength="150" />
      </allow>
      <filter>
      <header name="pragma" />
      </filter>
   </headers_in>
   <headers_out default="filter">
      <allow>
      <header name="content-type" />
      <header name="content-length" />
      <header name="set-cookie" value="~SESSID=%alphanum%" maxlength="150" />
      </allow>
      <filter>
      <header name="server" />
      </filter>
   </headers_out>
</config>
```

proxyfilter_config.xml

```
<config>
   <charsetsfile>/etc/proxyfilter_charsets</charsetsfile>
   <mappingsfile>/etc/proxyfilter_mappings</mappingsfile>
   <webappfile>/etc/proxyfilter_webapp.xml</webappfile>
   <logfile>/var/log/proxyfilter_log</logfile>
   <loglevel>info</loglevel>
   <http_methods>GET,POST,HEAD</http_methods>
   <headers_in default="deny">
      <allow>
      <header name="host" value="~^[-_a-zA-Z0-9\.@]*$" maxlength="50" />
      <header name="cookie" value="~SESSID=%alphanum%" maxlength="150" />
      </allow>
      <filter>
      <header name="pragma" />
      </filter>
   </headers_in>
   <headers_out default="filter">
      <allow>
      <header name="content-type" />
      <header name="content-length" />
      <header name="set-cookie" value="~SESSID=%alphanum%" maxlength="150" />
      </allow>
      <filter>
      <header name="server" />
      </filter>
   </headers_out>
</config>
```

proxyfilter_config.xml

```
<config>
   <charsetsfile>/etc/proxyfilter_charsets</charsetsfile>
   <mappingsfile>/etc/proxyfilter_mappings</mappingsfile>
   <webappfile>/etc/proxyfilter_webapp.xml</webappfile>
   <logfile>/var/log/proxyfilter_log</logfile>
   <loglevel>info</loglevel>
   <http_methods>GET,POST,HEAD</http_methods>
   <headers_in default="deny">
      <allow>
      <header name="host" value="~^[-_a-zA-Z0-9\.@]*$" maxlength="50" />
      <header name="cookie" value="~SESSID=%alphanum%" maxlength="150" />
      </allow>
      <filter>
      <header name="pragma" />
      </filter>
   </headers_in>
   <headers_out default="filter">
      <allow>
      <header name="content-type" />
      <header name="content-length" />
      <header name="set-cookie" value="~SESSID=%alphanum%" maxlength="150" />
      </allow>
      <filter>
      <header name="server" />
      </filter>
   </headers_out>
</config>
```

proxyfilter_config.xml

```
<config>
   <charsetsfile>/etc/proxyfilter_charsets</charsetsfile>
   <mappingsfile>/etc/proxyfilter_mappings</mappingsfile>
   <webappfile>/etc/proxyfilter_webapp.xml</webappfile>
   <logfile>/var/log/proxyfilter_log</logfile>
   <loglevel>info</loglevel>
   <http_methods>GET,POST,HEAD</http_methods>
   <headers_in default="deny">
      <allow>
      <header name="host" value="~^[-_a-zA-Z0-9\.@]*$" maxlength="50" />
      <header name="cookie" value="~SESSID=%alphanum%" maxlength="150" />
      </allow>
      <filter>
      <header name="pragma" />
      </filter>
   </headers_in>
   <headers_out default="filter">
      <allow>
      <header name="content-type" />
      <header name="content-length" />
      <header name="set-cookie" value="~SESSID=%alphanum%" maxlength="150" />
      </allow>
      <filter>
      <header name="server" />
      </filter>
   </headers_out>
</config>
```

proxyfilter_config.xml

```
<config>
   <charsetsfile>/etc/proxyfilter_charsets</charsetsfile>
   <mappingsfile>/etc/proxyfilter_mappings</mappingsfile>
   <webappfile>/etc/proxyfilter_webapp.xml</webappfile>
   <logfile>/var/log/proxyfilter_log</logfile>
   <loglevel>info</loglevel>
   <http_methods>GET,POST,HEAD</http_methods>
   <headers_in default="deny">
      <allow>
      <header name="host" value="~^[-_a-zA-Z0-9\.@]*$" maxlength="50" />
      <header name="cookie" value="~SESSID=%alphanum%" maxlength="150" />
      </allow>
      <filter>
      <header name="pragma" />
      </filter>
   </headers_in>
   <headers_out default="filter">
      <allow>
      <header name="content-type" />
      <header name="content-length" />
      <header name="set-cookie" value="~SESSID=%alphanum%" maxlength="150" />
      </allow>
      <filter>
      <header name="server" />
      </filter>
   </headers_out>
</config>
```

proxyfilter_config.xml

```
<config>
   <charsetsfile>/etc/proxyfilter_charsets</charsetsfile>
   <mappingsfile>/etc/proxyfilter_mappings</mappingsfile>
   <webappfile>/etc/proxyfilter_webapp.xml</webappfile>
   <logfile>/var/log/proxyfilter_log</logfile>
   <loglevel>info</loglevel>
   <http_methods>GET,POST,HEAD</http_methods>
   <headers_in default="deny">
      <allow>
      <header name="host" value="~^[-_a-zA-Z0-9\.@]*$" maxlength="50" />
      <header name="cookie" value="~SESSID=%alphanum%" maxlength="150" />
      </allow>
      <filter>
      <header name="pragma" />
      </filter>
   </headers_in>
   <headers_out default="filter">
      <allow>
      <header name="content-type" />
      <header name="content-length" />
      <header name="set-cookie" value="~SESSID=%alphanum%" maxlength="150" />
      </allow>
      <filter>
      <header name="server" />
      </filter>
   </headers_out>
</config>
```

proxyfilter_config.xml

```
<config>
   <charsetsfile>/etc/proxyfilter_charsets</charsetsfile>
   <mappingsfile>/etc/proxyfilter_mappings</mappingsfile>
   <webappfile>/etc/proxyfilter_webapp.xml</webappfile>
   <logfile>/var/log/proxyfilter_log</logfile>
   <loglevel>info</loglevel>
   <http_methods>GET,POST,HEAD</http_methods>
   <headers_in default="deny">
      <allow>
      <header name="host" value="~^[-_a-zA-Z0-9\.@]*$" maxlength="50" />
      <header name="cookie" value="~SESSID=%alphanum%" maxlength="150" />
      </allow>
      <filter>
      <header name="pragma" />
      </filter>
   </headers_in>
   <headers_out default="filter">
      <allow>
      <header name="content-type" />
      <header name="content-length" />
      <header name="set-cookie" value="~SESSID=%alphanum%" maxlength="150" />
      </allow>
      <filter>
      <header name="server" />
      </filter>
   </headers_out>
</config>
```

proxyfilter_webapp.xml

- Décrit la structure de l'application Web
- Règles <url>, <file>, <script>, <param>
- Conteneurs <webapp>, <directory>
- Permet d'alterner entre le mode de filtrage Whitelist et Blacklist en fonction du contexte (path) de la requête

proxyfilter_webapp.xml

```
<webapp default="deny" allowindex="true">
  <allow>
    <file name="~%filename%_%digit%\.html?$" inherit="true" maxlength="20" />
    <script name="index.php" inherit="false">
      <param name="id" value="%number%" maxlength="10" optional="true" method="get" />
      <param name="page" value="%alphanum%" maxlength="20" optional="true" method="both" />
    </script>
  </allow>
  <deny>
    <file name="~^\..*" inherit="true" />
    <url value="%xss%" inherit="true" />
  </deny>
  <directory name="images" default="allow" allowindex="false">
    <allow>
      <file name="%filename%.gif" maxlength="50" />
    </allow>
  </directory>
  <directory name="news">
    <allow>
      <file name="~%filename%\.php" inherit="true" />
    </allow>
  </directory>
</webapp>
```

proxyfilter_webapp.xml

```
<webapp default="deny" allowindex="true">
  <allow>
    <file name="~%filename%_%digit%\.html?$" inherit="true" maxlength="20" />
    <script name="index.php" inherit="false">
      <param name="id" value="%number%" maxlength="10" optional="true" method="get" />
      <param name="page" value="%alphanum%" maxlength="20" optional="true" method="both" />
    </script>
  </allow>
  <deny>
    <file name="~^\..*" inherit="true" />
    <url value="%xss%" inherit="true" />
  </deny>
  <directory name="images" default="allow" allowindex="false">
    <allow>
      <file name="%filename%.gif" maxlength="50" />
    </allow>
  </directory>
  <directory name="news">
    <allow>
      <file name="~%filename%\.php" inherit="true" />
    </allow>
  </directory>
</webapp>
```

proxyfilter_webapp.xml

```
<webapp default="deny" allowindex="true">
   <allow>
      <file name="~%filename%_%digit%\.html?$" inherit="true" maxlength="20" />
      <script name="index.php" inherit="false">
         <param name="id" value="%number%" maxlength="10" optional="true" method="get" />
         <param name="page" value="%alphanum%" maxlength="20" optional="true" method="both" />
      </script>
   </allow>
   <deny>
      <file name="~^\..*" inherit="true" />
      <url value="%xss%" inherit="true" />
   </deny>
   <directory name="images" default="allow" allowindex="false">
      <allow>
         <file name="%filename%.gif" maxlength="50" />
      </allow>
   </directory>
   <directory name="news">
      <allow>
         <file name="~%filename%\.php" inherit="true" />
      </allow>
   </directory>
</webapp>
```

proxyfilter_webapp.xml

```
<webapp default="deny" allowindex="true">
  <allow>
    <file name="~%filename%_%digit%\.html?$" inherit="true" maxlength="20" />
    <script name="index.php" inherit="false">
      <param name="id" value="%number%" maxlength="10" optional="true" method="get" />
      <param name="page" value="%alphanum%" maxlength="20" optional="true" method="both" />
    </script>
  </allow>
  <deny>
    <file name="~^\..*" inherit="true" />
    <url value="%xss%" inherit="true" />
  </deny>
  <directory name="images" default="allow" allowindex="false">
    <allow>
      <file name="%filename%.gif" maxlength="50" />
    </allow>
  </directory>
  <directory name="news">
    <allow>
      <file name="~%filename%\.php" inherit="true" />
    </allow>
  </directory>
</webapp>
```

proxyfilter_webapp.xml

```
<webapp default="deny" allowindex="true">
  <allow>
    <file name="~%filename%_%digit%\.html?$" inherit="true" maxlength="20" />
    <script name="index.php" inherit="false">
      <param name="id" value="%number%" maxlength="10" optional="true" method="get" />
      <param name="page" value="%alphanum%" maxlength="20" optional="true" method="both" />
    </script>
  </allow>
  <deny>
    <file name="~^\..*" inherit="true" />
    <url value="%xss%" inherit="true" />
  </deny>
  <directory name="images" default="allow" allowindex="false">
    <allow>
      <file name="%filename%.gif" maxlength="50" />
    </allow>
  </directory>
  <directory name="news">
    <allow>
      <file name="~%filename%\.php" inherit="true" />
    </allow>
  </directory>
</webapp>
```

proxyfilter_webapp.xml

```
<webapp default="deny" allowindex="true">
  <allow>
    <file name="~%filename%_%digit%\.html?$" inherit="true" maxlength="20" />
    <script name="index.php" inherit="false">
      <param name="id" value="%number%" maxlength="10" optional="true" method="get" />
      <param name="page" value="%alphanum%" maxlength="20" optional="true" method="both" />
    </script>
  </allow>
  <deny>
    <file name="~^\..*" inherit="true" />
    <url value="%xss%" inherit="true" />
  </deny>
  <directory name="images" default="allow" allowindex="false">
    <allow>
      <file name="%filename%.gif" maxlength="50" />
    </allow>
  </directory>
  <directory name="news">
    <allow>
      <file name="~%filename%\.php" inherit="true" />
    </allow>
  </directory>
</webapp>
```

proxyfilter_mappings

- Définit les règles de réécriture de l'URL
- 3 types de règles
  - forward : URL mapping à préfixe de la requête
  - reverse : URL mapping à préfixe lors d'une redirection
  - exact : URL mapping exact de la requête

proxyfilter_mappings

/	http://www.example.com/	forward
http://www.example.com/	/	reverse
/stats	/http://stats.example.com/index.cgi	exact
/webmail/	http://webmail.example.com/	forward
http://webmail.example.com/	/webmail/	reverse

proxyfilter_mappings

/	**http://www.example.com/**	**forward**
http://www.example.com/	/	reverse
/stats	/http://stats.example.com/index.cgi	exact
/webmail/	**http://webmail.example.com/**	**forward**
http://webmail.example.com/	/webmail/	reverse

proxyfilter_mappings

/	http://www.example.com/	forward
http://www.example.com/	**/**	**reverse**
/stats	/http://stats.example.com/index.cgi	exact
/webmail/	http://webmail.example.com/	forward
http://webmail.example.com/	**/webmail/**	**reverse**

proxyfilter_mappings

/	http://www.example.com/	forward
http://www.example.com/	/	reverse
/stats	**/http://stats.example.com/index.cgi**	**exact**
/webmail/	http://webmail.example.com/	forward
http://webmail.example.com/	/webmail/	reverse

proxyfilter_charsets

- Permettent de mémoriser et nommer des jeux de caractères souvent utilisés
- Utilisent la syntaxe des regexp Perl
- Peuvent être appelés dans une règle par %charset_name%

proxyfilter_charsets

```
filename    [a-zA-Z0-9][-_.a-zA-Z0-9]{0,255}
textarea    [-a-zA-Z0-9éàê',:;!?_()\[\]\s]*
xss         <script>.*</script>
digit       [0-9]
number      [0-9]{1,20}
alphanum    [a-zA-Z0-9]
url         (http|ftp)://([a-zA-Z][-a-zA-Z0-9]{0,50}\.)?
             [a-zA-Z][-a-zA-Z0-9]{0,50}\.{a-zA-Z}{2,4}
            (/[a-zA-Z0-9])/?(?[-_=a-zA-Z0-9]+)?
```

Exemple :

```
<file name="~^%alphanum%{1,20}_%digit%{3}$" />
```

≡

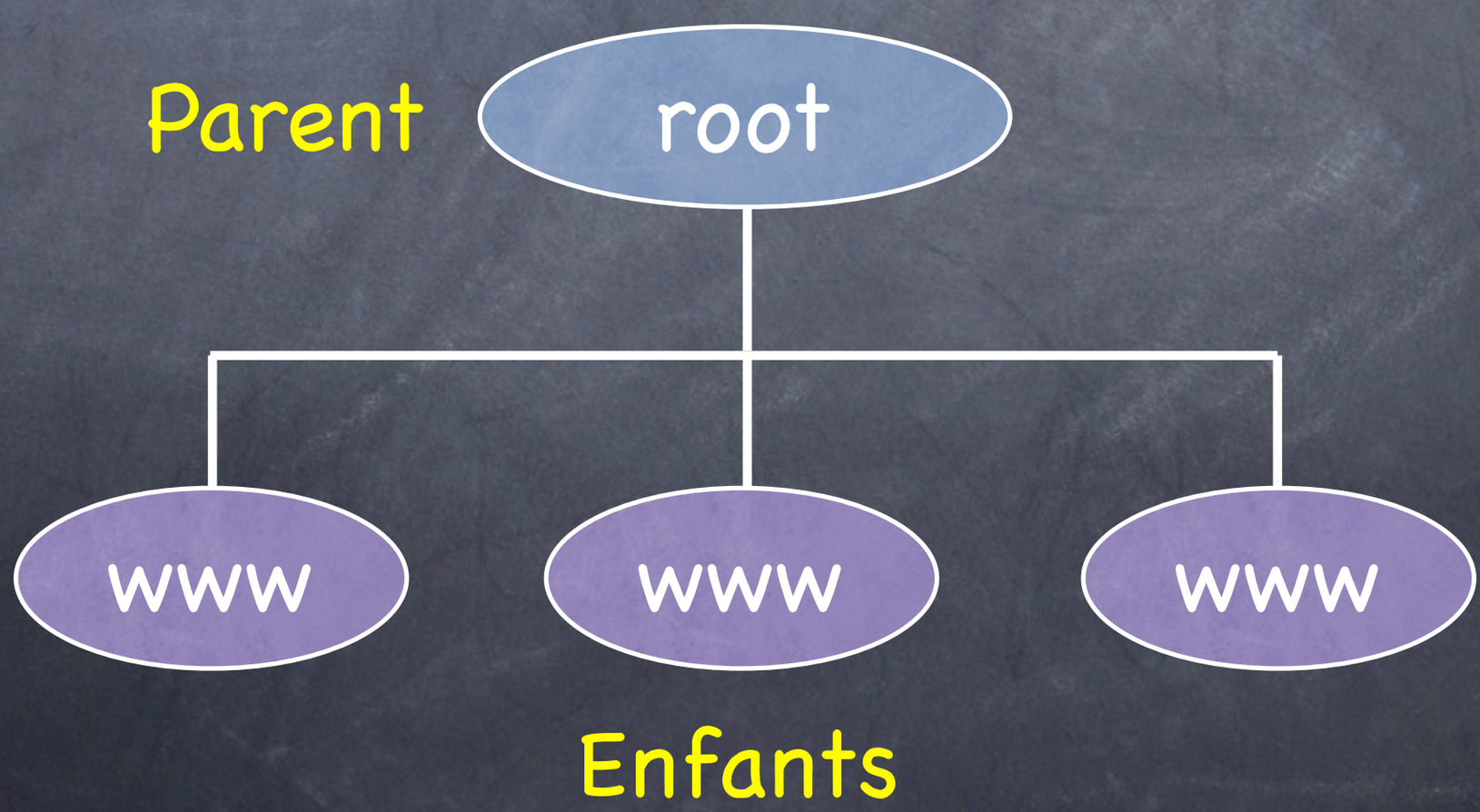
```
<file name="~^[a-zA-Z0-9]{1,20}_[0-9]{3}$" />
```

Types de règles

- règles <file>
- règles <script> et <param>
- règles <url>
- règles <header>

Limitations

- Sous Unix, chaque processus enfant de Apache dispose de son propre espace mémoire, il n'est donc pas aisé de partager des variables globales

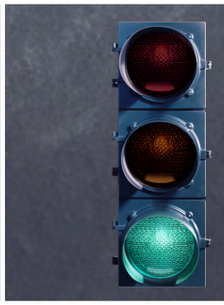


Limitations

- Le contenu de la réponse n'est pas filtré
- Cela demanderait de faire appel à un module spécialisé dans le parsing du HTML, retournant le document sous forme d'un arbre
- Le coût en performances serait pénalisant pour un intérêt limité

ProxyFilter

Limitations

	XSS	BOF	Cookie poisoning	Hidden Field	DOS	SQL
SANCTUM						
ProxyFilter						

Améliorations futures

- Optimisation des performances
- Mode d'auto-apprentissage
- Filtrer le contenu de la réponse
- Interface graphique de configuration
- Mémoriser les états/sessions (stateful)
- Mise à jour dynamique de la blacklist
- Simplification de l'installation

Conclusion

- Augmenter le niveau de sécurité nécessite un firewall stateful
- Il est plus important de filtrer la requête que la réponse du point de vue sécurité
- Ni whitelist ni blacklist ne sont la panacée
- Travail enrichissant qui m'a permis d'approfondir HTTP, Apache et Perl

DEMO