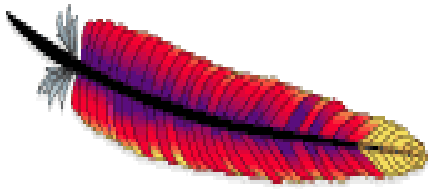
HELHa

Haute École
Louvain en Hainaut

Laboratoire de sécurité internet

Serveur Web Apache



Jean-Louis Gouwy

Plan

HELHa
Haute École
Louvain en Hainaut

- Introduction (Historique / Caractéristiques)
- Le modèle client-serveur (Les échanges)
- Fonctionnalités
- Architecture (Vue d'ensemble / Noyau / Modules / Filtres)
- Installation (Par les sources / Par les binaires / Architecture sous Fedora 26)
- Configuration (Structure de l'httpd.conf / Structure de l'httpd.conf sous Fedora 26 / Le contexte des directives)
- Environnement principal (ServerName / ServerRoot / DocumentRoot / ServerAdmin / ServerTokens/ Listen / ErrorDocument)
- Exercice 1
- Contrôler Apache (L'arbre des processus / Création des instances de httpd Les directives: MinSpareServers - MaxSpareServers – StartServers – MaxRequestWorkers – ServerLimit - User - Group)
- Les sites perso
- Les redirections simples
- Les index de répertoires
- Exercice 2

Plan

HELHa
Haute École
Louvain en Hainaut

- Protection des sites web (Objectifs / Contrôle sur l'origine des clients / Le conteneur <Directory> / Contrôle par authentification / Autres types d'authentification)
- Exercice 3
- Hébergement virtuel (Introduction / Principe / Par Ip / Par nom)
- Exercice 4
- Les fichiers journaux (Le journal des erreurs / Le journal des accès / Les hôtes virtuels)
- Exercice 5
- Les certificats (Objectifs / SSL / Structure générale / Fonctionnement Question de confiance / Signature / Implémentation/ La redirection SSL)
- Exercice 6
- Références

Introduction

HELHa
Haute École
Louvain en Hainaut

- **HISTORIQUE**
  - 1965: Naissance en Suisse de l'idée originale de l'hypertexte (réseau d'un ensemble de documents informatiques liés entre eux) puis de l'httpd (http daemon).
  - 1992: 26 serveurs Web.
  - 1995: Démarrage du projet Apache (version 1.0).
  - 1996: Apache devient le serveur Web le plus répandu.
  - 1999: Constitution de l'ASF (Apache Software Foundation).

http://www.apache.org (ASF)
http://httpd.apache.org (serveur Apache)

Le serveur Apache tient son nom d'une des plus fières tribus indiennes dont la vigueur et la faculté d'adaptation n'était plus à prouver.

Introduction

HELHa
Haute École
Louvain en Hainaut

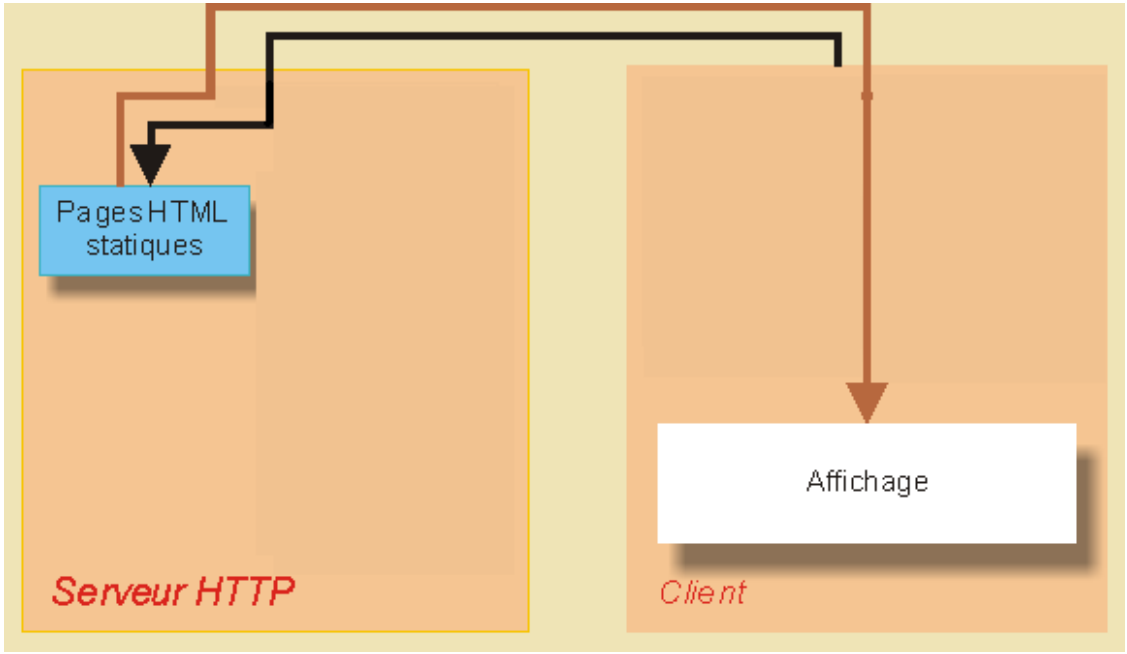
- **CARACTERISTIQUES**

  - Open Source et gratuité.
  - Multi-plateformes (Unix- Unix Like - Linux - Windows – MacOs).
  - C'est le serveur web 'opensource' le plus répandu.
    Source: *http://www.netcraft.com*
  - Modulable, fiable, performant, sécurisé et extensible.
  - Supporte les protocoles: HTTP / HTTPS (le plus souvent), POP3, FTP ...

Le modèle client-serveur

HELHa
Haute École
Louvain en Hainaut

- **LES ECHANGES: Demande d'une page html 'statique'**

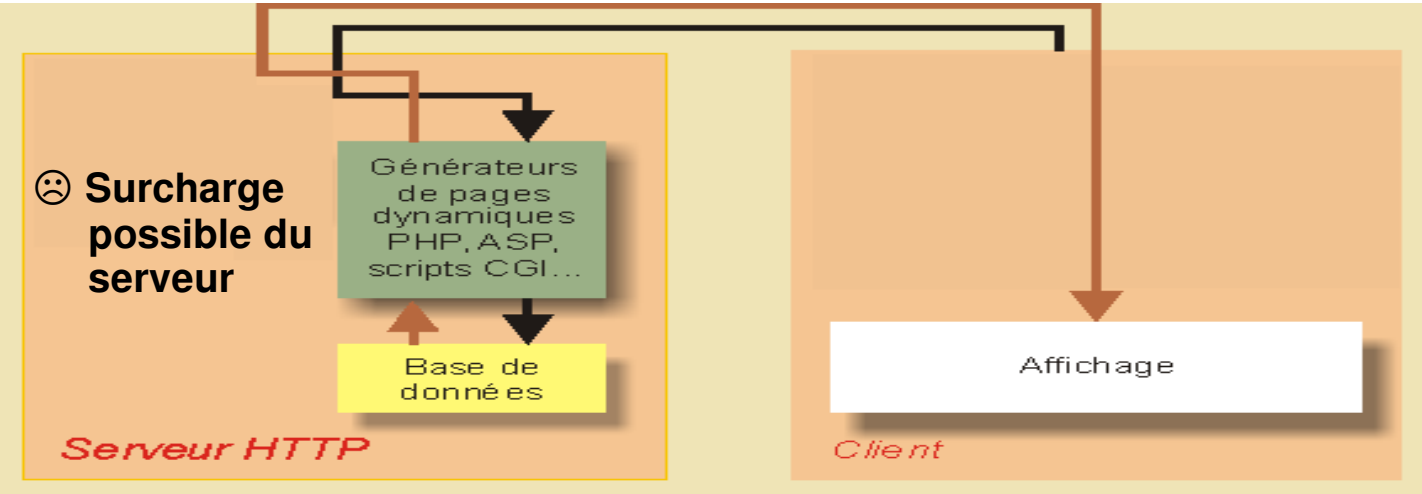


Ici, tout le contenu est défini dans la page HTML demandée.
Le serveur l'envoie tel quel au client qui n'a plus qu'à l'interpréter et l'afficher.

Le modèle client-serveur

HELHa
Haute École
Louvain en Hainaut

- **LES ECHANGES: Demande d'une page html 'dynamique' (server side)**



- Les scripts CGI construisent totalement un flux HTML au moment de leur appel. On utilise de plus en plus souvent le C (compilé) ou le PERL (interprété).

ou

- PHP (libre) ou ASP (Microsoft) qui sont des langages spécialisés et puissants.

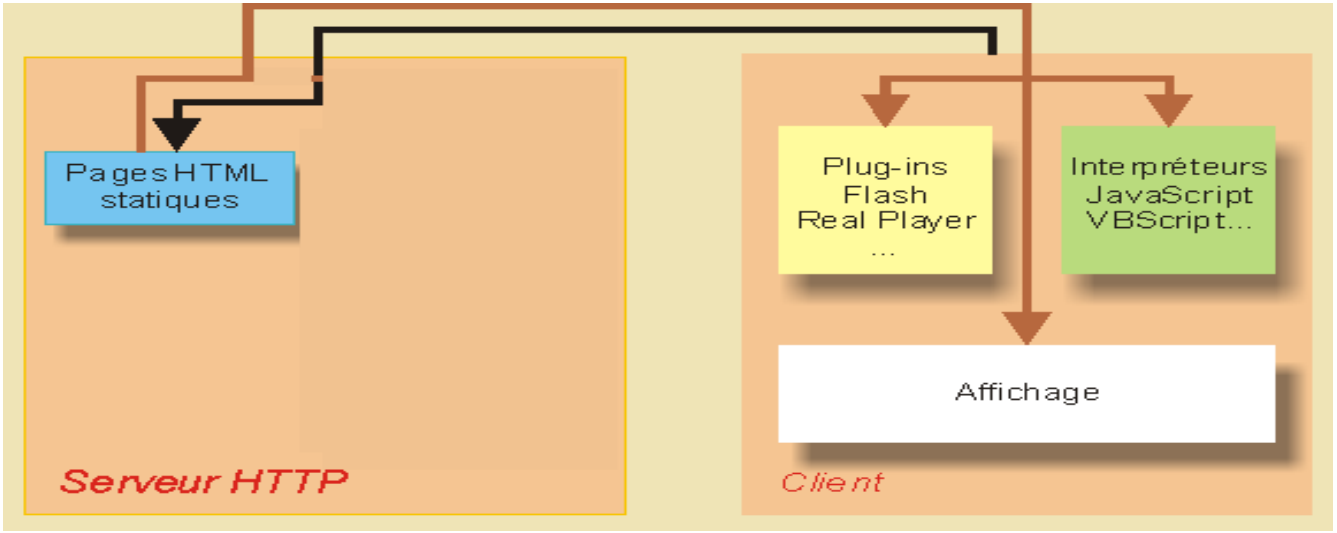
Exemples:

- Demande d'un résultat calculé par le serveur et dont les données sont fournies par le client.
- Demande de données qui se trouvent dans une base de données hébergée côté serveur.

Le modèle client-serveur

HELHa
Haute École
Louvain en Hainaut

- **LES ECHANGES: Demande d'une page html 'dynamique' (client side)**



☺ Pas de surcharge sur le serveur.
Pages animées.

☹ Incompatibilité des navigateurs.
Failles de sécurité possibles côté client.

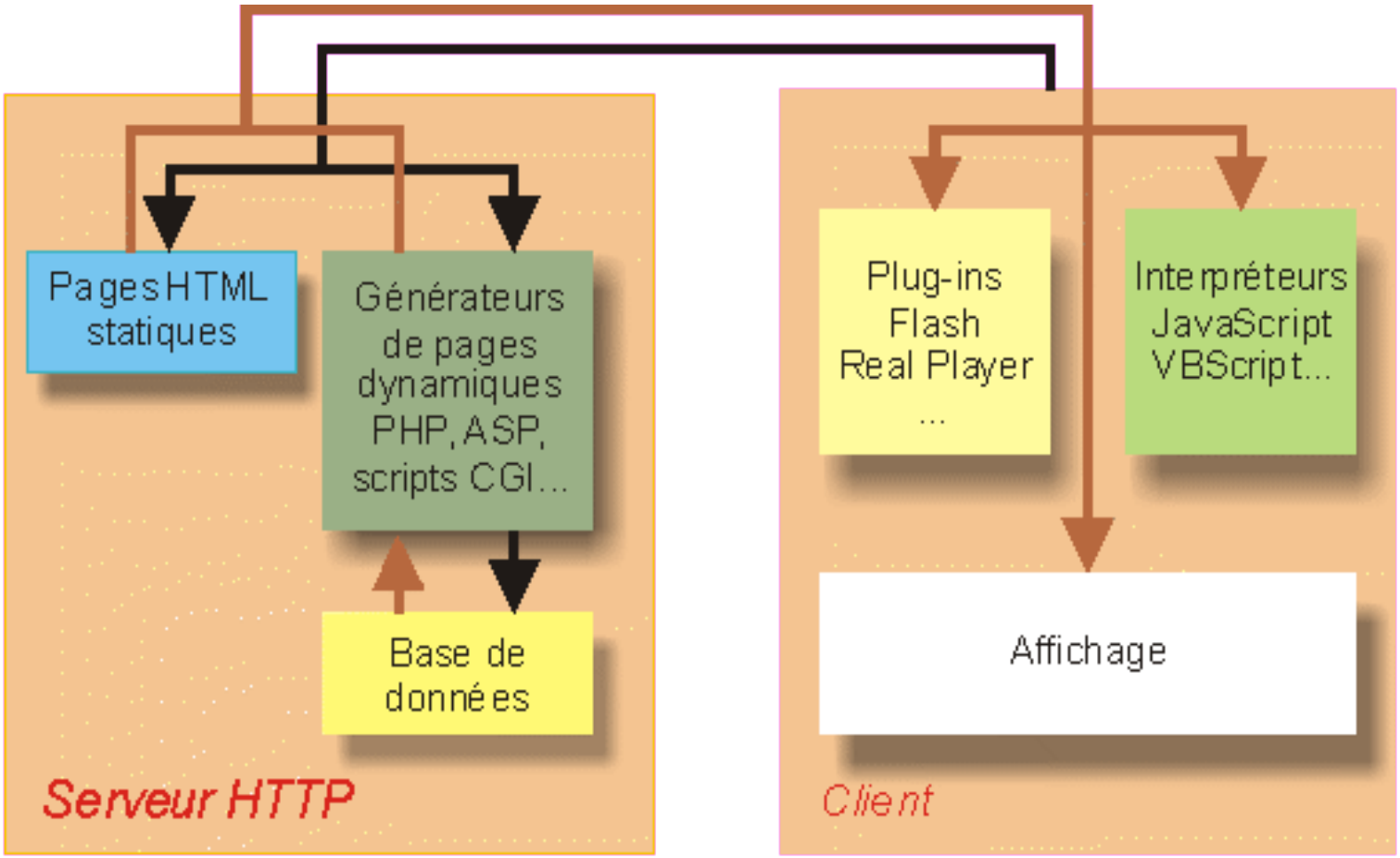
Exemples:

- Contrôler la validité de données avant de les envoyer au serveur.
- Effectuer un traitement local pour afficher un résultat.
- Animations quelconques.

Le modèle client-serveur

HELHa
Haute École
Louvain en Hainaut

- **LES ECHANGES: Modèle complet**



Fonctionnalités

HELHa
Haute École
Louvain en Hainaut

- **Conformité aux standards**:
  Entièrement conforme aux standards HTTP/1.1 – RFC 2616).

- **Scalabilité**:
  Permet d'héberger un grand nombre de sites web sur une même machine sans voir les performances diminuer de manière cruciale.

- **Objets dynamiques partagés (shared objects)**:
  Modules pouvant être compilés séparément du noyau et activés au démarrage d'Apache.

- **Personnalisation**:
  Programmation (en C ou en Perl) de modules personnels via l'API d'Apache.

- **Programmation**:
  Permet la programmation côté serveur (PHP, Perl, servlets Java, Java Server Page, Active Server Pages, CGI, FastCGI, Server-Side Includes)

Fonctionnalités

HELHa
Haute École
Louvain en Hainaut

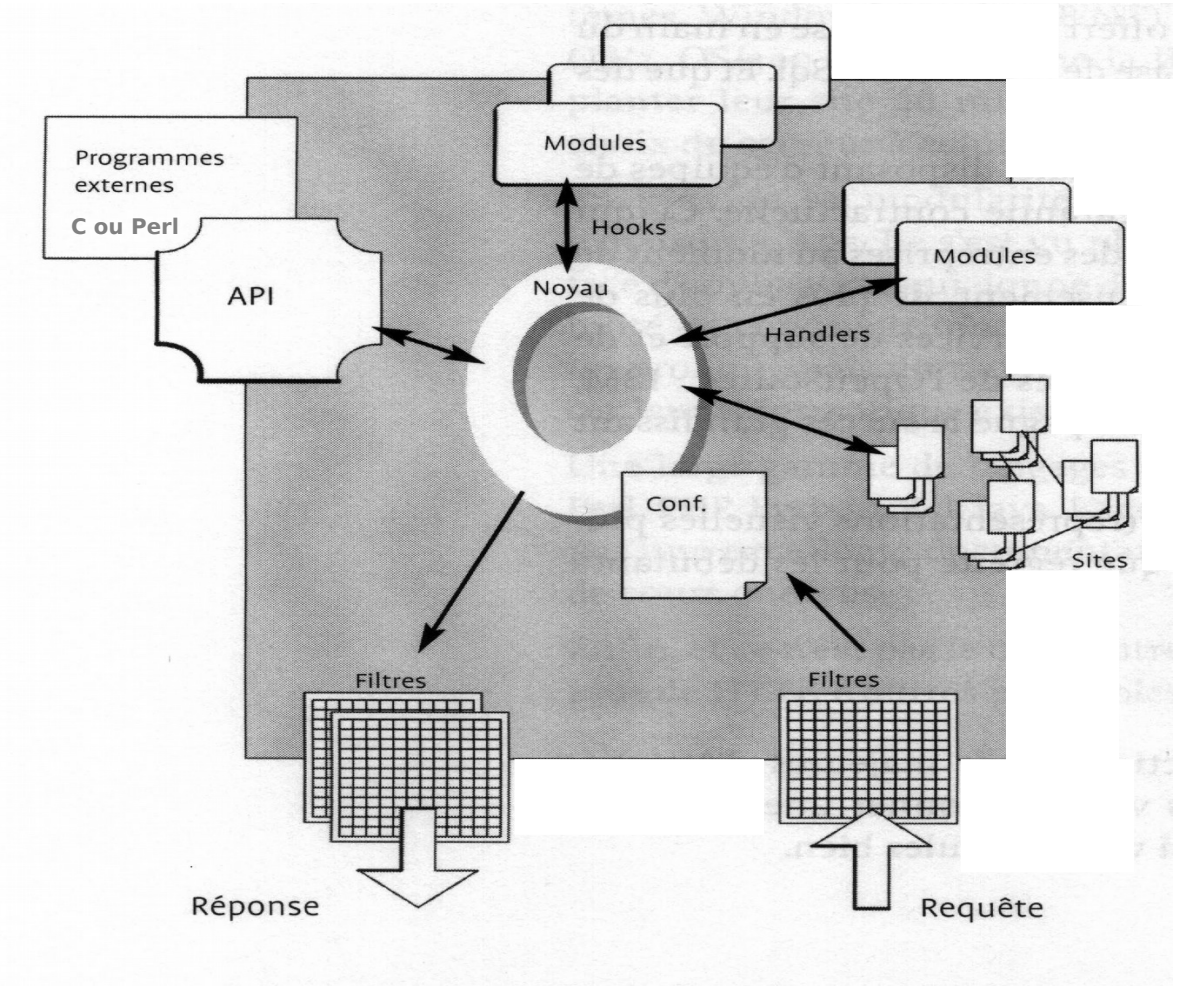
- **Serveur mandataire**:
  Apache peut être configuré en proxy serveur à l'aide de son module `mod_proxy`.

- **Sécurité**:
  Par différentes méthodes d'authentification de l'utilisateur (fichier plats ou bases de données dbm-mysql-...)

  Par l'autorisation du chiffrement sur Internet via l'échange de certificats numériques (SSL)

Architecture

- **VUE D'ENSEMBLE**



Architecture

HELHa
Haute École
Louvain en Hainaut

- **Le noyau**

- Contient toutes les fonctionnalités de base du serveur (ex. directive Listen).

- Il distribue le travail aux modules ou aux programmes externes.

- Les documents sont extraits du ou des sites dont il a la charge.

- Il reçoit et comprend une requête.

- Il prépare et envoie une réponse.

- Il connaît ses ressources disponibles (modules disponibles, liste et localisation des sites à gérer ...) grâce à son fichier de configuration (httpd.conf).

- Pour chaque module qu'il va solliciter, il dispose d'un jeu de directives.

Architecture

HELHa
Haute École
Louvain en Hainaut

- **Les modules**

- Ils étendent les fonctionnalités du noyau.

- Ils peuvent être :

  . Standards: Maintenus par l'ASF (ex. `mod_auth_basic`)
  Font partie de la distribution d'Apache
  (`http://httpd.apache.org/docs/2.4/fr/mod`)

  . Tiers: Maintenus par des tiers (ex. `mod_auth_oracle`)
  Disponibles sur la toile.

- Ils peuvent être installés:

  . Lors de la compilation (ou de l'installation) de la distribution d'Apache (modules standards)

  . Lors d'une compilation séparée (modules tiers) via l'utilitaire *apxs*
  `# path_to/apxs -cia module.c` *(-cia : compile, installe et active)*

Architecture

HELHa
Haute École
Louvain en Hainaut

- **Les modules (suite)**

- Le choix des modules à utiliser se fait dans le fichier de configuration principal. Deux directives sont disponibles pour gérer les modules :

  `LoadModule` : qui permet de charger un module au démarrage

  `<IfModule>` : qui permet d'activer certaines parties du fichier de configuration si un module a été chargé.

- `httpd -l` Pour voir la liste des modules compilés dans le cœur d'Apache. Ce n'est pas la liste des modules chargés dynamiquement via la directive LoadModule.

```
Compiled in modules:
  core.c
  http_core.c
  mod_so.c
```

Doivent toujours être présents.

Permet le bon fonctionnement de la directive LoadModule (chargement de modules de type Shared Object)

`httpd -L | grep -i userdir` Pour connaître le module correspondant à chaque directive (ici la directive 'UserDir').

Architecture

HELHa
Haute École
Louvain en Hainaut

- **Les filtres**

  Les filtres sont des modules standards ou tiers que vous choisissez d'incorporer ou non à Apache.

  . soit à l'entrée, sur le chemin des demandes entrantes
  . soit en sortie, sur le chemin des réponses sortantes

  Par exemple un module de compression des données qui:

  . en entrée décompresse des requêtes compressées par un browser
  . en sortie compresse l'envoi de tous les documents du site

Installation

HELHa
Haute École
Louvain en Hainaut

- **Par les sources**

  - Hors cadre du cours.

  - Plus d'info: `http://httpd.apache.org/docs/2.4/fr/install.html`

Installation

HELHa
Haute École
Louvain en Hainaut

- **Par les binaires**

  - Via la commande d'installation automatique de la distribution.

  - Ce sera la retenue pour la suite du cours malgré que cette installation est moins souple que par compilation des sources (ciblage des modules plus difficile)...
    ... par contre, elle offre un outil de désinstallation.

CentOS 6 : `yum install httpd –y` (Apache version 2.2)

CentOS 7 : `yum install httpd –y` (Apache version 2.4.6)

Fedora 26 Server avec Apache embarqué.
(Apache version 2.4.25) :
Nous retiendrons cette distribution car :
- la version d'Apache offerte est de la dernière branche 2.x
- de plus, le protocole HTTP 2 est correctement supportée à partir d'Apache 2.4.23.

Installation

HELHa
Haute École
Louvain en Hainaut

- **Architecture sous Fedora 26**

/etc/httpd/ → dossier contenant l'ensemble des fichiers de configuration.

- **conf/httpd.conf** → fichier principal de configuration.
- **conf.d** → dossier contenant les fichiers secondaires de configuration.
- **conf.modules.d** → dossier contenant les fichiers de lancement des modules.

/var/www/ → dossier contenant les données du site par défaut.

- **cgi-bin**: dossier (vide) contenant les scripts.
- **html**: dossier (vide) contenant les pages du site par défaut.

/var/log/httpd/ → dossier contenant les journaux

- **access_log** → journal des accès aux pages traitées par le serveur.
- **error_log** → journal des erreurs.

Installation

HELHa
Haute École
Louvain en Hainaut

- **Architecture sous Fedora 26**

/user/share/

- error → dossier contenant les pages affichées en cas d'erreur
- icons → dossier contenant quelques icônes
- manual → dossier contenant la documentation

/usr/lib64/httpd/modules → dossier contenant le binaire des modules (.so)

/usr/sbin/httpd → le daemon Apache

Remarques:

```
# systemctl start httpd.service
```
→ lancement d'Apache (port d'écoute par défaut: 80)

```
# systemctl enable httpd.service
```
→ sera lancé au démarrage du système

Configuration

- **Structure de l'httpd.conf**

Section 1

Environnement global

Les directives contrôlent le comportement global des processus Apache (ex. Nb maximum de clients qu'Apache pourra servir en même temps, le port d'écoute ...)

Section 2

Configuration du site "principal" ou "par défaut"

Les directives fournissent les paramètres du site par défaut, celui qui répond aux requêtes qui ne sont pas gérées par un hôte virtuel. Ces directives servent également à fournir les valeurs par défaut pour tous les hôtes virtuels.

Section 3

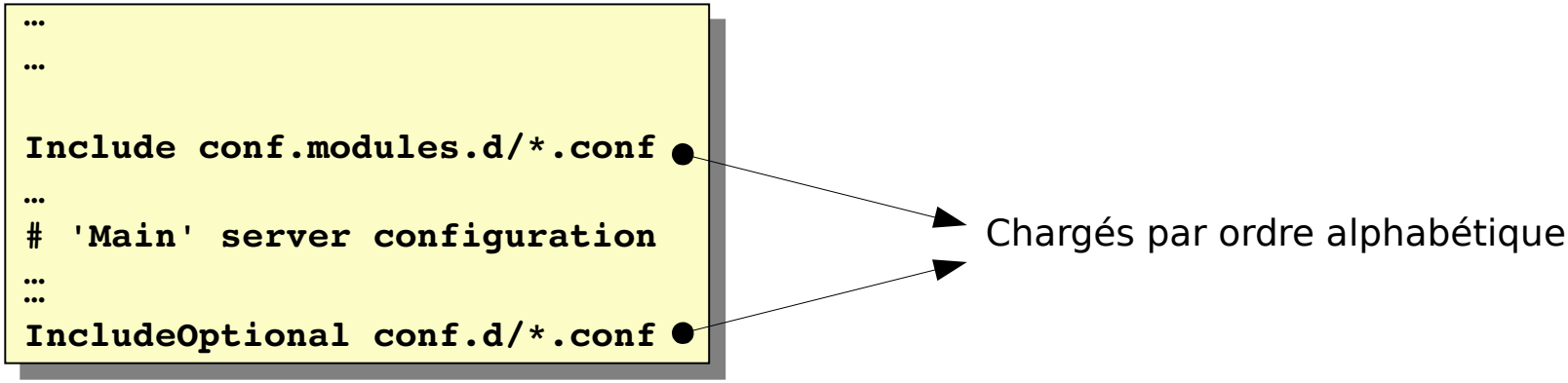
Configuration des "hôtes virtuels"

Directives qui sont propres à hôte virtuel. Le processus central leur attribue les requêtes envoyées à des adresses ou à des noms de sites particuliers.

Configuration

HELHa
Haute École
Louvain en Hainaut

- **Structure de l'httpd.conf sous Fedora 26**



Il est préférable de ne pas le modifier.

On préfèrera la création d'un fichier spécifique à nos besoins dans le dossier `/etc/httpd/conf.d`.

Cela permet de ne pas bloquer les mises à jour éventuelles du fichier principal et simplifiera grandement les migrations (il suffit de récupérer notre fichier de configuration).

Configuration

HELHa
Haute École
Louvain en Hainaut

- **Le contexte des directives**

L'aide officielle d'Apache classe les directives dans 4 catégories

http://httpd.apache.org/docs/2.4/fr/mod/quickreference.html

s	server config
v	virtual host
d	directory
h	.htaccess

Configuration

HELHa
Haute École
Louvain en Hainaut

- **Le contexte des directives (suite)**

  - Contexte général du serveur (server config - s)

    Agit sur tout le serveur.

    ex. StartServers (uniquement dans ce contexte)
    ServerName (dans ce contexte mais peut être redéfinie pour chaque hôte virtuel)

  - Contexte hôte virtuel (virtualhost – v)

    De nombreuses directives d'hôte virtuel (définies dans le conteneur `<VirtualHost>`) surchargent celles qui sont générales au serveur.

    ex. ServerName
    DocumentRoot

Configuration

HELHa
Haute École
Louvain en Hainaut

- **Le contexte des directives (suite)**

  - Contexte conteneur (directory – d)

    Inclut les directives qui ne peuvent s'appliquer que dans un des 3 conteneurs (`<Directory>`, `<Files>` et `<Location>`) et dont la portée est limitée à ce conteneur.

    ex. Require

  - Contexte .htaccess (.htaccess – h)

    Sont traitées comme celles d'un conteneur `<Directory>` de l'httpd.conf. La principale différence est que les directives d'un fichier `htaccess` peuvent être désactivées au moyen de la directive `AllowOverrides` dans `l'httpd.conf`.

L'environnement principal

HELHa
Haute École
Louvain en Hainaut

- **La directive ServerName**

  Apache doit toujours pouvoir déterminer le nom d'hôte de la machine sur laquelle il tourne car il l'utilise pour créer des URL d'autoréférence.

  Exemple: `ServerName www.mysite.be`

- **La directive ServerRoot**

  Répertoire dans lequel les fichiers de configuration, les logs et les modules sont gardés. Ce nom de dossier servira à préfixer tout chemin relatif rencontré dans l'httpd.conf.

  Exemple: `ServerRoot  "/etc/httpd"`

L'environnement principal

HELHa
Haute École
Louvain en Hainaut

- **La directive DocumentRoot**

  Dossier dans lequel les documents du site par défaut sont déposés.
  C'est donc ce dossier qui contient les fichiers qu'Apache fournit lorsqu'il reçoit des requêtes avec l'URL /.

  Exemple: `DocumentRoot "/var/www/html"`

  ☞ Le changement de nom du `DocumentRoot` doit aussi être effectué dans Son conteneur `<Directory ...>`, qui regroupe toutes les directives s'appliquant à `DocumentRoot` et ses sous-répertoires.

- **La directive ServerAdmin**

  Adresse mail du webmaster qui pourrait s'afficher sur certains documents construits par le serveur et renvoyés au client en cas d'incidents.
  Valable si la directive **ServerSignature** est à l'état on permettant alors l'ajout de cette information en bas de page.

  Exemple: `ServerAdmin webmaster@mysite.be`

L'environnement principal

HELHa
Haute École
Louvain en Hainaut

- **La directive ServerTokens**

  Permet de contrôler le contenu de l'en-tête `Server` inclus dans la réponse envoyée au client.

  Exemples:

  `ServerTokens Prod[uctOnly]`
  ➔ Le serveur renvoie (par ex.): `Server: Apache`

  `ServerTokens Major`
  ➔ Le serveur renvoie (par ex.): `Server: Apache/2`

  `ServerTokens Minor`
  ➔ Le serveur renvoie (par ex.): `Server: Apache/2.0`

  `ServerTokens Min[imal]`
  ➔ Le serveur renvoie (par ex.): `Server: Apache/2.0.41`

  `ServerTokens OS`
  ➔ Le serveur renvoie (par ex.): `Server: Apache/2.0.41 (Unix)`

  `ServerTokens Full` (valeur par défaut)
  ➔ Le serveur renvoie (par ex.): `Server: Apache/2.0.41 (Unix) PHP/4.2.2`

L'environnement principal

HELHa
Haute École
Louvain en Hainaut

- **La directive Listen**

  Pour définir les adresses IP et les numéros de ports sur lesquels Apache attend et reçoit les connexions des clients.

  Exemples:

  `Listen 80` ➔ Apache écoute sur toutes les interfaces sur le port 80.

  `Listen 10.0.0.7:80` ➔ Apache écoute sur le port 80 sur l'interface d'IP 10.0.0.7.

  ```
  Listen 80
  Listen 8080
  ```
  ➔ Apache écoute sur le port 80 et 8080 sur toutes les interfaces.

  ```
  Listen 192.168.1.1:80
  Listen 216.180.25.168:443
  ```
  ➔ Apache répond aux requêtes http sur l'interface interne et aux requêtes https (connexions SSL) sur l'interface publique.

L'environnement principal

HELHa
Haute École
Louvain en Hainaut

- **La directive ErrorDocument**

  Pour remplacer les pages d'erreur standards envoyées au client en cas de problème.

  Exemples:

  ```
  ErrorDocument 403 "Vous n'êtes pas autorisé à lire cette page !
  ```
  Ici on affiche simplement un texte adapté à l'erreur.

  ```
  ErrorDocument 401 /missing.html
  ```
  Ici on affiche une page html sensée se trouver à la racine du site web.

  ```
  ErrorDocument 500 http://www.bidon.com/erreur.html
  ```
  Ici on affiche une page html extérieure au site.

  ☞ `ErrorDocument 401` nécessite toujours une URL interne.

Exercice 1

HELHa
Haute École
Louvain en Hainaut

- **Un serveur simple**

a) Clonez le .vdi de la machine Fedora contenant une version supérieure à 2.4.23 du serveur Apache.

b) Créez une MV 'apache' liée au disque cloné.

c) Préfixez tous les fichiers d'extension `.conf` du dossier `/etc/httpd/conf.d` par '`01-`' et créez les fichier `01-main.conf` et `00-server.conf`

d) Configurez le serveur pour:
. qu'il écoute sur le port 80 sur toutes les interfaces
. qu'il présente une page d'accueil `index.html` lors d'une requête vers l'URL de ce site (inventez son contenu)
. qu'il affiche une page html personnalisée en cas d'erreur 404

e) Vérifiez la configuration du serveur.

Exercice 1

HELHa
Haute École
Louvain en Hainaut

- **Un serveur simple (suite)**

f) Testez votre serveur pour vérifier si la page d'accueil est bien offerte:
. à l'aide de l'utilitaire telnet
. à l'aide d'un navigateur quelconque

g) Reconfigurez Apache pour qu'il écoute cette fois sur le port 8080:

h) Testez votre serveur:
. à l'aide de l'utilitaire telnet
. à l'aide d'un navigateur quelconque

i) Compilez, installez et testez un module tiers (`mod_pony`)

Contrôler Apache

HELHa
Haute École
Louvain en Hainaut

- **L'arbre des processus**

Apache lance plusieurs daemons en parallèle; ceux-ci se trouvent en permanence à l'écoute du réseau afin de pouvoir répondre rapidement à un grand nombre de requêtes simultanées.

```
# service httpd restart
# ps -ef
                    …
root    2116 1      S      0:00 /usr/sbin/httpd
apache  2119 2116   S      0:00 /usr/sbin/httpd
apache  2120 2116   S      0:00 /usr/sbin/httpd
apache  2121 2116   S      0:00 /usr/sbin/httpd
apache  2122 2116   S      0:00 /usr/sbin/httpd
apache  2123 2116   S      0:00 /usr/sbin/httpd
apache  2124 2116   S      0:00 /usr/sbin/httpd
apache  2125 2116   S      0:00 /usr/sbin/httpd
apache  2126 2116   S      0:00 /usr/sbin/httpd
```

Processus principal.
C'est lui qui reçoit les requêtes et les distribue à ses fils.

Maximum 256 instances (fils).

Contrôler Apache

HELHa
Haute École
Louvain en Hainaut

- **Création des instances de httpd**

Plus d'info.: http://httpd.apache.org/docs/2.4/fr/misc/perf-tuning.html

Les instances sont créées en fonction du type de module Multi-Processus (MPM) chargé :

- **Le MPM prefork:** chaque processus enfant possède un seul thread et chaque processus gère une seule connexion à la fois.

☺ Aussi rapide qu'en 'worker'.
Stable et universel (utilisable avec les modules tiers qui ne supportent pas le threading et compatible avec des logiciels ou OS anciens).

☹ Plus gourmand en mémoire qu'en 'worker'.

- Le MPM worker: chaque processus enfant possède plusieurs threads et chaque thread gère une seule connexion à la fois.

☺ Moins gourmand en mémoire qu'en 'prefork'.

☹ Pas universel

- **le MPM event** utilise les threads, mais il a été conçu pour traiter davantage de requêtes simultanément.

☺ Moins gourmand en mémoire qu'en 'prefork' et plus rapide qu'en 'worker'

☹ Pas universel

Contrôler Apache

HELHa
Haute École
Louvain en Hainaut

- **Les directives MinSpareServers / MaxSpareServers / StartServers / MaxRequestWorkers / ServerLimit**

MinSpareServers: Nombre minimal d'instances de serveurs.

MaxSpareServers: Nombre maximum d'instances de serveurs.

StartServers: Nombre de serveurs supplémentaires créés au démarrage d'Apache.

MaxRequestWorkers / ServerLimit:

Limite le nombre de processus qui peuvent tourner simultanément (chaque connexion cliente en utilise un).

Exemple: Voir exercice 2

Contrôler Apache

HELHa
Haute École
Louvain en Hainaut

- **Les directives User et Group**

  Utilisateur et groupe Linux sous lequel s'exécuteront les processus fils d'Apache chargés de répondre aux requêtes des clients.
  Le processus maître doit être lancé sous le compte `root` pour pouvoir changer le user et le group de ses processus fils.

  Exemples:

```
User apache
Group apache
```

Les sites perso

HELHa
Haute École
Louvain en Hainaut

- Il est possible de permettre aux utilisateurs disposant d'un compte sur le serveur de posséder leur propre site.

- Pratique très courante parmi les FAI qui proposent l'hébergement de pages web de leurs clients.

- Cette fonctionnalité est fournie par le module standard `mod_userdir`.

```
LoadModule userdir_module ...
```

- Elle est activée par la directive `Userdir`.

```
<IfModule mod_userdir.c>
        #UserDir disabled
        UserDir public_html
</IfModule>
```

Si le module 'userdir' est chargé, alors un utilisateur du système pourra y héberger son site Web dont la racine se trouvera dans le dossier 'public_html' de sa home directory.

- Le site sera accessible vial'URL: `http://ip_serveur/~login_utilisateur`
  Exemple: `http://www.mysite.be/~jean`

Les sites perso

HELHa
Haute École
Louvain en Hainaut

- Autres formes de la directive `UserDir` :

`UserDir disabled <user1 user2 ...>`

Désactive la gestion des sites perso pour la liste des utilisateurs indiquée.

`UserDir disabled`

Désactive la gestion de tous les sites perso.
Souvent utilisée avant une directive `UserDir enabled`.

`UserDir enabled <user1 user2 ...>`

Active la gestion des sites perso pour la liste des utilisateurs indiquée.

Exemple:

```
UserDir disabled
UserDir enabled jean louis
```

Les redirections simples

HELHa
Haute École
Louvain en Hainaut

- La directive standard `Alias` permet d'accéder facilement à des documents HTML en dehors de l'arborescence `DocumentRoot`.

- Cette fonctionnalité est fournie par le module standard `mod_alias`.

```
LoadModule alias_module ...
```

- Soit accéder à la page HTML `/usr/share/doc/HTML/index.html` via le site `www.mysite.be` par redirection:

```
Alias /CentOS "/usr/share/doc/HTML/"

<Directory "/usr/share/doc/HTML">
      Require all granted
</ Directory>
```

Car la cible est dans un dossier situé en dehors de l'arborescence du site web
→ permettre explicitement l'accès à ce dossier.

Cette page sera accessible via l'URL:

```
http://www.mysite.be/CentOS
```

Les index de répertoires

HELHa
Haute École
Louvain en Hainaut

- **Recherche d'une page d'accueil**

  Apache est capable de retrouver la page `index.html` sans pour autant que celle-ci soit indiquée dans l'URL.

  Exemple:

  `http://www.mysite.be/cours`

  équivaut à:

  `http://www.mysite.be/cours/index.html`

Les index de répertoires

HELHa
Haute École
Louvain en Hainaut

- **Recherche d'une page d'accueil (suite)**

  - C'est le module standard `mod_dir` qui recherche et fournit une page (`index.html` par défaut) susceptible de se trouver dans le répertoire indiqué dans l'URL.

  - Si la directive `DirectoryIndex` est présente, elle permettra d'ajouter des références à d'autres pages.

```
DirectoryIndex index.html index.htm index.php
```

Les index de répertoires

HELHa
Haute École
Louvain en Hainaut

- **Recherche d'une page d'accueil (suite)**

  - Si aucune page d'accueil n'est trouvée, le module `mod_autoindex` (piloté par sa directive `IndexOptions`) crée un index des fichiers du répertoire concerné.

    Exemples:

    `IndexOptions None` (indexation classique)
    `IndexOptions FancyIndexing` (indexation au look plus agréable)
    `IndexOptions FancyIndexing VersionSort`
    (idem + tri des entrées contenant des numéros de versions)

  - Pour supprimer l'indexation sur un dossier particulier

```
<Directory directory_name>
        Options -Indexes
</Directory>
```

Exercice 2

HELHa
Haute École
Louvain en Hainaut

a) Reprendre l'exercice 1 et reconfigurez le serveur pour qu'il écoute sur le port 80 sur votre interface.

b) Quel est le Module Multi-Processus (MPM) utilisé actuellement ?

c) Combien de processus enfants y a-t-il actuellement ?

d) Configurez-le pour:

- qu'il lance 7 processus enfant en mode '`prefork`' au démarrage et qu'il s'assure qu'il en reste toujours au moins 3 en réserve. En cas de montée en charge, 20 processus enfants maximum seront créés pour satisfaire les requêtes.
  Lorsque le serveur aura satisfait toutes les requêtes de cette montée, il se stabilisera avec 10 processus enfants.

  Testez tout cela, à l'aide d'outils adéquats.

Exercice 2 (Suite)

HELHa
Haute École
Louvain en Hainaut

- que la requête `http://www.mysite.be/pamauthor` présente la page `/usr/share/doc/pam/html/sag-author.html`
- qu'il donne la possibilité aux utilisateurs `jean`, `louis` et `bernard` de disposer d'un site perso dont la page d'accueil puisse être `index.html` ou `accueil.html` (inventez leur contenu).
  Le site perso de bernard ne sera pas accessible pour le moment.
- qu'un index des entrées du dossier '`cours`' soit présenté lors de la requête `http://www.mysite.be/cours`

e) Testez vos configurations.

Protection des sites web

HELHa
Haute École
Louvain en Hainaut

- **Objectifs**

Bloquer l'accès à une partie d'un site en limitant sa consultation à quelques utilisateurs privilégiés ou à quelques machines.

- Accéder à une porte d'administration du site (login membres ...)

- Verrouiller une partie du site en construction.

- Permettre des accès seulement à un ou plusieurs réseaux de machines.

- ...

Protection des sites web

HELHa
Haute École
Louvain en Hainaut

- **Contrôle sur l'origine du client**

La directive `Require [not]` est utilisée au cours de la phase d'**autorisation** pour accorder ou refuser l'accès à une ressource à un client.

Exemples

`Require all granted` → l'accès est autorisé sans restriction.
`Require all denied` → l'accès est systématiquement refusé.

☛ Le fournisseur d'autorisation générique `all` est pris en charge par le module `mod_authz_core`

Plus d'info `https://httpd.apache.org/docs/2.4/fr/mod/mod_authz_core.html`

Protection des sites web

HELHa
Haute École
Louvain en Hainaut

- **Contrôle sur l'origine du client (suite)**

Autres exemples

```
Require ip 10/255.0.0.0 172.20/16 192.168.2.4
```
→ Accès si l'@Ip du client fait partie d'une des tranches spécifiées.

```
Require not ip 10 172.20/16 192.168.2.4
```
→ Refus si l'@Ip du client fait partie d'une des tranches spécifiées.

```
Require host www.example.org
```
→ Les hôtes dont les noms correspondent ou se terminent par la chaîne spécifiée se verront accorder l'accès.

```
Require not host .net example.edu
```
→ Les hôtes dont les noms correspondent ou se terminent par la chaîne spécifiée se verront refuser l'accès.

☛ Les fournisseurs d'autorisation **`ip`** `et` **`host`** sont pris en charge par le module `mod_authz_host`

Plus d'info `https://httpd.apache.org/docs/2.4/fr/mod/mod_authz_host.html`

Protection des sites web

HELHa
Haute École
Louvain en Hainaut

- **Contrôle sur l'origine du client (suite)**

L'intervant DNS

```
Require host www.example.org
Require not host .net example.edu
```

☛ Le fournisseur d'autorisation **host** va effectuer une double recherche DNS sur l'adresse IP du client :

- une recherche DNS inverse sur l'adresse IP source pour trouver le nom associé
- puis une recherche DNS directe sur le nom renseigné dans la directive pour trouver l'adresse IP originale.

L'accès ne sera accordé que si les recherches DNS inverse et directe sont cohérentes.

Tutoriel sur le contrôle d'accès

```
https://httpd.apache.org/docs/2.4/fr/howto/access.html
```

Protection des sites web

HELHa
Haute École
Louvain en Hainaut

- **Contrôle sur l'origine du client (suite)**

  Les conteneurs d'autorisation: `<RequireAll>`, `<RequireAny>` et `<RequireNone>`

  Elles sont prises en charge par le module `mod_authz_core`.

```
<RequireAll>
    Require …
    Require …
    …
</Require>
```

Si toutes les directives d'autorisation sont vraies alors l'accès est autorisé.

```
<RequireAny>
    Require …
    Require …
    …
</Require>
```

Si au moins une directive d'autorisation est vraie alors l'accès est autorisé.

```
<RequireNone>
    Require …
    Require …
    …
</Require>
```

Si au moins une directive d'autorisation est vraie alors l'accès est refusé.

  ☛ Les directives d'autorisation inversées (not) sont interdites dans les directives `<RequireAny>` et `<RequireNone>`.

Plus d'info

`https://httpd.apache.org/docs/2.4/fr/mod/mod_authz_core.html#requireall`

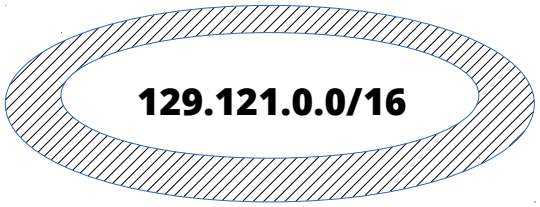
Protection des sites web

HELHa
Haute École
Louvain en Hainaut

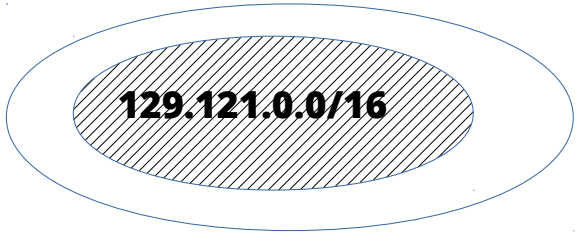
- **Contrôle sur l'origine du client (suite)**

Exemples

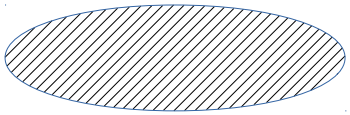
`Require ip 129.121` →



`Require not ip 129.121` →

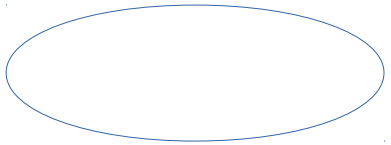


`Require host ex.org` →



← `Require all denied`

`Require all granted` →



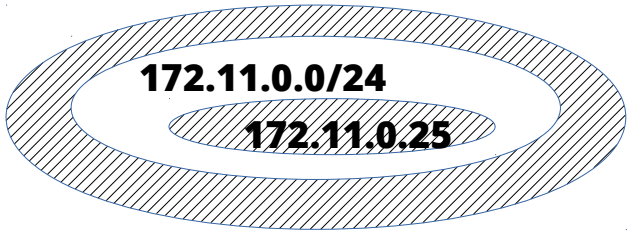
Protection des sites web

HELHa
Haute École
Louvain en Hainaut

- **Contrôle sur l'origine du client (suite)**

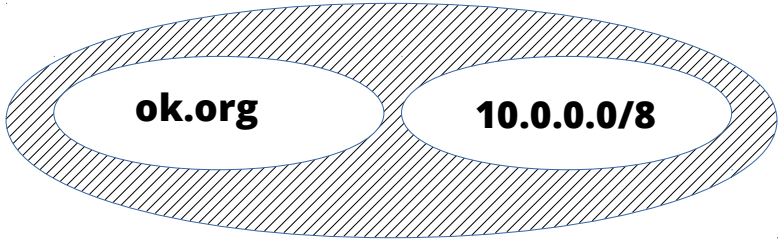
Exemples

```
<RequireAll>
  Require ip 172.11.0.0/24
  Require not ip 172.11.0.25
</Require>
```




```
<RequireAll>
  Require host ok.org
  Require not host foo.ok.org
</Require>
```


```
<RequireAny>
  Require host ok.org
  Require ip 10
</Require>
```



Protection des sites web

HELHa
Haute École
Louvain en Hainaut

- **Le conteneur <Directory>**

  Il contient des directives qui ne s'appliquent qu'à celui-ci.

```
<Directory />
        …
        …
</Directory>

<Directory /var/www/html>
        …
        …
</Directory>

<Directory /var/www/html/private>
        …
        …
</Directory>
```

Héritage

Ajouts ou redéfinitions

Héritage

Ajouts ou redéfinitions

Protection des sites web

HELHa
Haute École
Louvain en Hainaut

- **Contrôle par authentification**

L'authentification HTTP de base:

- Gérée par les modules `mod_auth_basic` (type : Basic ou Digest), `mod_authz_user` (**autorisation**) et `mod_authn_file` (**authentification**).
- Les "*users/passwords*" sont stockés dans un fichier texte.
- L'utilitaire `htpasswd` permet de créer et mettre à jour ce fichier.
- Les mots de passe de ce fichier sont cryptés à l'aide de la fonction `crypt`.
- Cette méthode est la plus simple mais la moins sécurisée car les identifiants sont transmis chiffrés en 'Base64' (très facilement déchiffrables). A n'utiliser que sous https !!!

Exemple:

```
# htpasswd -c /var/www/securite/userfile toto  → création (en dehors du site... sécurité oblige ! )

# cat /var/www/securite/userfile
toto:cnVPtfAz2xw60
# htpasswd -b /var/www/securite/userfile albert secret → ajout avec mdp
```

☞ `-m → pour chiffrer le mdp en MD5` `-p → pas de chiffrement`
`-s → pour chiffrer le mdp en SHA`

Protection des sites web

HELHa
Haute École
Louvain en Hainaut

- **Contrôle par authentification**

  L'authentification HTTP de base: (suite)

  Activation de l'autorisation:

  Exemple

```
<Directory rep_a_proteger>
   AuthName "Domaine de test" → identificateur d'autorisation
   AuthType Basic → type d'authentification (Basic ou Digest)

   AuthUserFile /var/www/securite/userfile → (1)

   Require valid-user → (2)
</Directory>
```

(1) *Fichier contenant la liste des utilisateurs qui ont le droit de s'authentifier* (`mod_authn_file`).
C'est le ***fournisseur d'authentification*** **.**

(2) *Liste des utilisateurs qui peuvent accéder aux ressources après s'être authentifier correctement* (`mod_authz_user`).
Avec `Require valid-user`: *tous les utilisateurs repris dans la liste* (1).

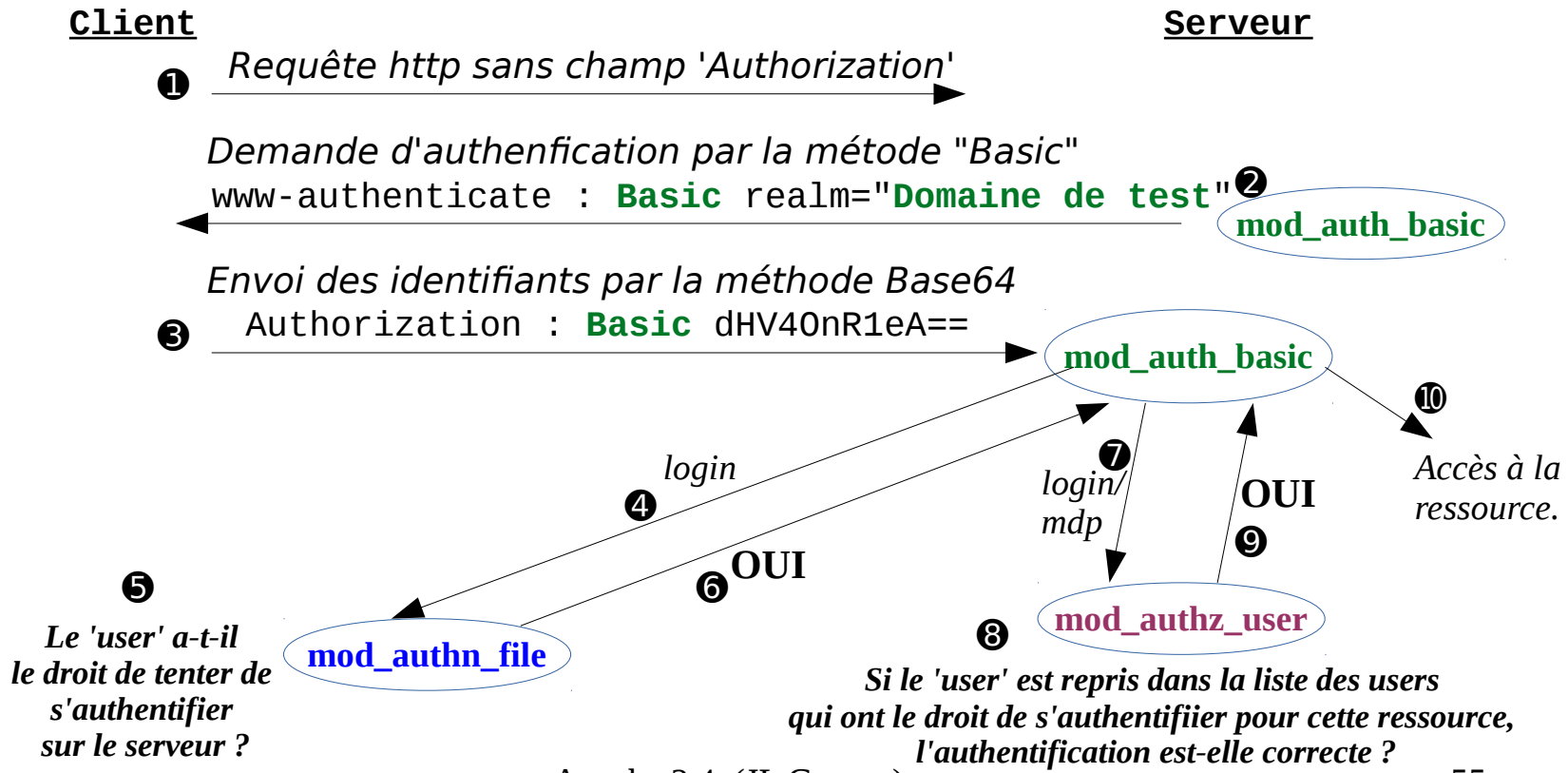
Protection des sites web

HELHa
Haute École
Louvain en Hainaut

- **Contrôle par authentification**

  L'authentification HTTP de base: (suite)

  Synoptique:



Protection des sites web

HELHa
Haute École
Louvain en Hainaut

- **Contrôle par authentification**

  L'authentification HTTP de base: (suite)

```
<Directory rep_a_proteger>
   AuthName "Domaine de test"
   AuthType Basic
   AuthUserFile /var/www/securite/userfile
   Require user toto albert
</Directory>
```

☞ Ici, seuls 'toto' et 'albert' repris dans 'userfile' ont le droit d'accéder aux ressources du répertoire moyennant une authentification valide.

Protection des sites web

HELHa
Haute École
Louvain en Hainaut

- **Contrôle par authentification**

  L'authentification HTTP de base: (suite)

  Soit un fichier texte `/var/www/securite/groupfile` contenant:

```
admins:toto albert
friends:toto linda
associates:bruno
```

  alors:

```
<Directory rep_a_proteger>
   AuthName "Domaine de test"
   AuthType Basic
   AuthUserFile /var/www/securite/userfile
   AuthGroupFile /var/www/securite/groupfile
   Require group admins friends
</Directory>
```

  ☞ Ici, seuls les membres des groupes 'admins' et 'friends' ont le droit d'accéder aux ressources du répertoire moyennant une authentification valide.

  La directive **AuthUserFile** reste obligatoire car elle contient la liste des **users qui peuvent s'authentifier.**

  La directive **Require** détermine **qui a l'autorisation** d'accéder aux ressources **après authentification** réussie).

Protection des sites web

HELHa
Haute École
Louvain en Hainaut

- **Autres types d'authentification**

  - Par digest: le mot de passe ne circule plus en clair mais tous les navigateurs ne gèrent pas cette méthode.

  - Par db: Unix DBM, MYSQL, Microsoft SQL Server, Oracle, Postgres ...

  - Par serveurs d'authentification: Radius, Ldap, Kerberos, Samba, Nis ...

  ☞ Pour une liste à jour :
  `httpd.apache.org/docs/2.4/fr/mod` (mot clé: *auth*)

  ☞ Pour un tuto sur l'authentification et autorisation :
  `https://httpd.apache.org/docs/2.4/fr/howto/auth.html`

Exercice 3

HELHa
Haute École
Louvain en Hainaut

- Configurez Apache pour:

  . rendre l'entrée d'un intranet (`http://www.mysite.be/intranet`) uniquement accessible par les machines de notre réseau local `10.103.0.0/16`.

  . que les utilisateurs 'tux', 'bill', 'louis' et 'jean' puissent s'authentifier lorsqu'ils accèdent au site `http://www.mysite.be/beta`; en sachant que seuls 'tux' et 'bill' auront le droit d'accéder à ses ressources.

- Testez vos configurations.
- Expliquez le principe de l'authentification Basic.
- Expliquez le codage Base64.
- Testez avec tshark que la méthode d'authentification 'Basic' n'est pas sécurisée.
- Testez d'autres outils permettant de (dé)coder un message chiffré par la méthode Base64.

Hébergement virtuel

HELHa
Haute École
Louvain en Hainaut

- **Introduction**
  - Pour gérer plusieurs sites web sur une seule machine serveur.
  - 2 méthodes:

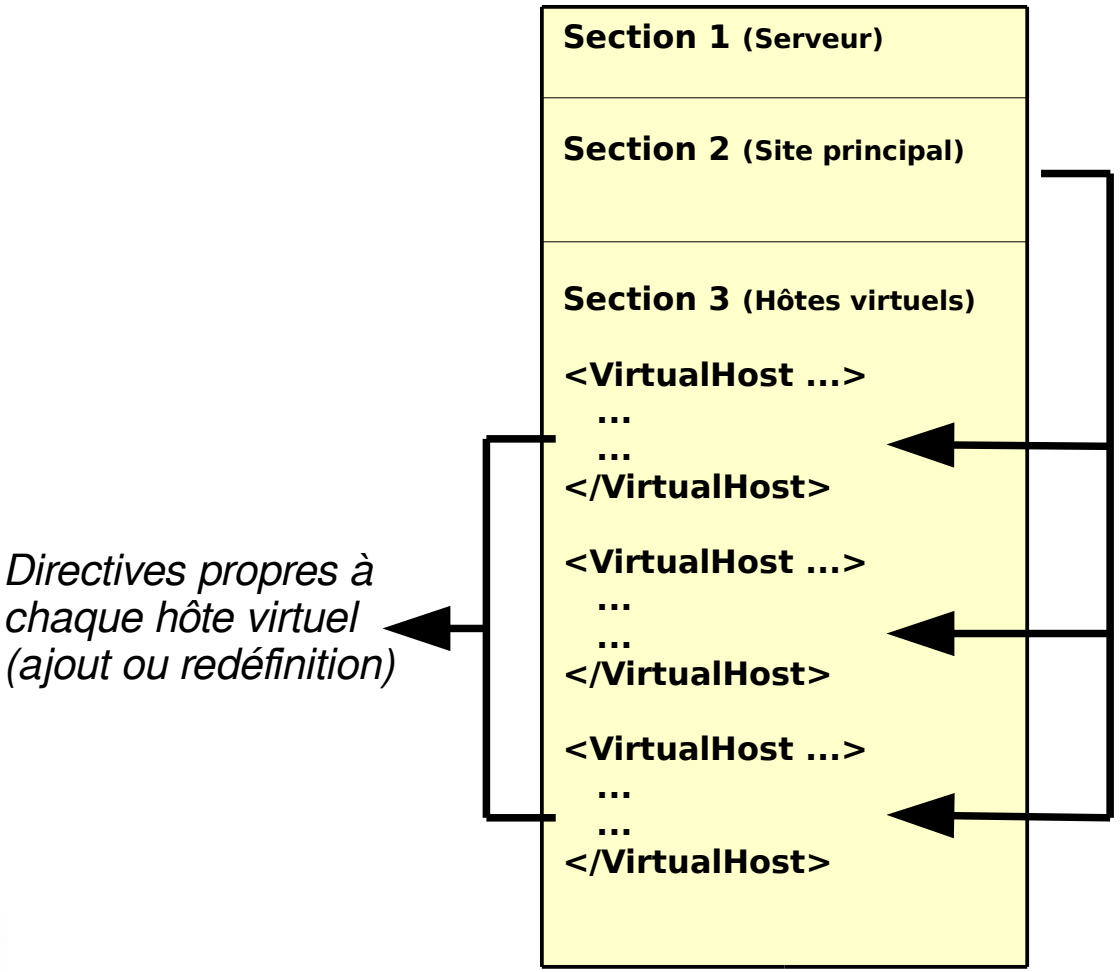
    a) Hébergement virtuel par adresse IP.

    b) Hébergement virtuel par nom.

Hébergement virtuel

HELHa
Haute École
Louvain en Hainaut

- **Principe**



☞ *La directive 'Listen' n'est pas autorisée en VirtualHost.*

Hébergement virtuel

HELHa
Haute École
Louvain en Hainaut

- **Hébergement par IP**
  - Les sites sont identifiés par l'IP sur laquelle arrivent les requêtes.
  - Une IP dédicacée à un seul site
  - Besoin de plusieurs interfaces physiques → nécessité d’alias.
  - De moins en moins utilisé (beaucoup d'Ip monopolisées).
  - Les clients non HTTP1.1 (rare) ont néanmoins besoin de ce type d'hébergement.

Hébergement virtuel

HELHa
Haute École
Louvain en Hainaut

- **Hébergement par IP**

httpd.conf

```
ServerName www.mysite.be
DocumentRoot /var/www/html

<VirtualHost 192.168.1.4>
  ServerName vhost1.mysite.be
  DocumentRoot /var/www/html/vhost1
</VirtualHost>

<VirtualHost 192.168.1.5>
  ServerName vhost2.mysite.be
  DocumentRoot /var/www/html/vhost2
</VirtualHost>
```

CONNECT 192.168.1.1:80
GET / HTTP/1.1

http://192.168.1.1

On reçoit la page d'accueil du site principal.

CONNECT 192.168.1.4:80
GET / HTTP/1.1

http://192.168.1.4

On reçoit la page d'accueil de vhost1.mysite.be

CONNECT 192.168.1.5:80
GET / HTTP/1.1

http://vhost2.mysite.be

Après résolution du nom, on reçoit la page d'accueil de vhost2.mysite.be

Hébergement virtuel

HELHa
Haute École
Louvain en Hainaut

- **Hébergement par IP**

  Remarques:

  - La création d'un alias sur une interface *iface* se fait par la création d'un fichier :

    /etc/sysconfig/network-scripts/ifcfg-iface:0
    ...
    IPADDR=nouvelle_ip
    NETMASK=nouveau_masque
    ONBOOT=yes
    NAME=iface:0
    DEVICE=iface:0

    Numéro d'alias

  - Pour créer d'autres alias sur cette même interface, il suffit de créer d'autres fichiers en incrémentant le numéro d'alias (...**:1**, ...**:2**, etc) et en adaptant les valeurs des variables NAME et DEVICE en conséquence.

  - Si une adresse IP peut être atteinte mais qu'aucun hôte n'est défini sur celle-ci, c'est le site principal qui répondra à la requête.

Hébergement virtuel

HELHa
Haute École
Louvain en Hainaut

- **Hébergement par nom**

  - Une IP dédicacée à plusieurs sites.
  - Les sites sont identifiés par le nom du serveur auquel s'adresse le client.
  - Un nom par hôte virtuel.
  - Exploitable à partir de HTTP1.1 qui véhicule le nom de l'hôte via une en-tête `Host` dans sa requête. Il permet ainsi d'identifier sa cible.
  - Si le serveur utilise un hébergement par IP , l'en-tête `Host` est ignorée.
  - Très utilisé chez les hébergeurs.

Hébergement virtuel

HELHa
Haute École
Louvain en Hainaut

- **Hébergement par nom**

httpd.conf

```
ServerName www.mysite.be
DocumentRoot /var/www/html

<VirtualHost 192.168.1.1>
  ServerName www.mysite.be
  DocumentRoot /var/www/html
</VirtualHost>

<VirtualHost 192.168.1.1>
  ServerName vhost1.mysite.be
  DocumentRoot /var/www/html/vhost1
</VirtualHost>

<VirtualHost 192.168.1.1>
  ServerName vhost2.mysite.be
  DocumentRoot /var/www/html/vhost2
</VirtualHost>

<VirtualHost 192.168.1.2>
  ServerName vhost3.mysite.be
  DocumentRoot /var/www/html/vhost3
</VirtualHost>

<VirtualHost 192.168.1.2>
  ServerName vhost4.mysite.be
  DocumentRoot /var/www/html/vhost4
</VirtualHost>
```

CONNECT 192.168.1.1:80
GET / HTTP/1.1
Host : 192.168.1.1

http://192.168.1.1

On reçoit la page d'accueil du site principal.

CONNECT 192.168.1.1:80
GET / HTTP/1.1
Host : www.mysite.be

http://www.mysite.be

Après résolution du nom, on reçoit la page d'accueil du site principal.

CONNECT 192.168.1.1:80
GET / HTTP/1.1
Host : vhost1.mysite.be

http://vhost1.mysite.be

Après résolution du nom, on reçoit la page d'accueil de vhost1.mysite.be

CONNECT 192.168.1.2:80
GET / HTTP/1.1
Host : vhost3.mysite.be

http://vhost3.mysite.be

Après résolution du nom, on reçoit la page d'accueil de vhost3.mysite.be

Hébergement virtuel

HELHa
Haute École
Louvain en Hainaut

- **Hébergement par nom**

  - A la lecture de `httpd.conf`, Apache crée une table contenant la liste des hôtes virtuels déclarés pour chaque Ip.

  Ainsi, pour l'illustration précédente, la table se présente comme suit :

Hébergement par nom sur l'IP	ServerName		
192.168.1.1	www.mysite.be	vhost1.mysite.be	vhost2.mysite.be
192.168.1.2	vhost3.mysite.be	vhost4.mysite.be	

Hôte virtuel primaire

Lorsque Apache reçoit une requête sur une Ip se trouvant dans la table, il recherche le ServerName qui correspond à l'en-tête Host de la requête …Si aucune correspondance n'est trouvée, c'est l'hôte virtuel primaire qui sera sélectionné.

Ainsi, pour qu'il soit toujours accessible, on indique souvent le site maître en tant qu'hôte virtuel primaire dans l'entrée correspondant à son Ip

Exercice 4

HELHa
Haute École
Louvain en Hainaut

Configurez Apache pour qu'il puisse en même temps gérer de l'hébergement par ip et par nom.

- `192.168.27.10` permettra d'héberger l'unique site `jean.mysite.be`

- `192.168.27.11` permettra d'héberger l'unique site `louis.mysite.be`

- `192.168.27.2` permettra d'héberger les sites `vh2.mysite.be` et `vh3.mysite.be`. Ce dernier ne sera accessible qu'aux users repris dans `/var/www/securite/pwd`.

- `192.168.27.1` permettra d'héberger le site `vh1.mysite.be`

- Le site maître sera toujours accessible par l'Ip `192.168.27.1` ou par `www.mysite.be`. Ce sera aussi le site par défaut si une requête arrive via une interface associée à aucun site.

Remarque: Les fonctionnalités des exercices précédents doivent toujours être opérationnelles.

Relancez Apache et testez sa configuration

Les fichiers journaux

HELHa
Haute École
Louvain en Hainaut

Pour véritablement gérer un serveur web, il est nécessaire de disposer d'un retour d'informations à propos de l'activité et des performances du serveur, ainsi que de tout problème qui pourrait survenir.

- **Le journal des erreurs** (Directives `ErrorLog` & `LogLevel`)

  Directive `ErrorLog` : Le nom et la localisation du journal.

  Exemples

```
ErrorLog logs/errors-logs
        Enregistrement vers un fichier particulier.

ErrorLog "|/usr/local/bin/erreurs_httpd"
        Traitement de l'erreur par un binaire.

ErrorLog syslog:user
        Traitement de l'erreur par le daemon syslogd.
```

Les fichiers journaux

HELHa
Haute École
Louvain en Hainaut

- **Le journal des erreurs**

Directive `LogLevel` : Indique quels sont les messages à écrire dans le fichier journal.

Exemple `LogLevel warn`

Tableau des criticités en ordre croissant

`emerg`	Urgences - le serveur est inutilisable.
`alert`	Des mesures doivent être prises immédiatement.
`crit`	Conditions critiques (accès réseau impossible par ex.).
`error`	Erreurs dans les pages, les scripts.
`warn`	Avertissements (pages mal codées, erreurs non bloquantes dans un script...
`notice`	Evénement important mais normal.
`info`	Informations.
`debug`	Enregistre TOUT ce qui se passe sur le serveur.

☞ *Lorsqu'un niveau particulier est spécifié, les messages de tous les autres niveaux de criticité supérieure seront aussi enregistrés (ex. le niveau* `crit` *enregistre en plus les messages de niveau* `alert` *et* `emerg` *).*

Les fichiers journaux

HELHa
Haute École
Louvain en Hainaut

- **Le journal des erreurs**

Format

[Wed Oct 11 14:32:52 2015] [error] [client 127.0.0.1] client denied by server configuration: /export/home/live/ap/htdocs/test

Date et l'heure du message.

Sévérité de l'erreur rapportée (directive LogLevel)

Adresse IP du client qui a généré l'erreur.

Le message proprement dit, qui indique dans ce cas que le serveur a été configuré pour interdire l'accès au client. Le serveur indique le chemin système du document requis (et non son chemin web).

Les fichiers journaux

HELHa
Haute École
Louvain en Hainaut

- **Le journal des accès**

Directives : `CustomLog, LogFormat & SetEnvIf`
Modules: `mod_log_config & mod_setenvif`

Directive `CustomLog` : La localisation du journal.

Exemple

```
CustomLog logs/access_log common
```

Les logs seront enregistrés selon l'alias 'common' défini dans la directive LogFormat (voir ci-après).

Les fichiers journaux

HELHa
Haute École
Louvain en Hainaut

- **Le journal des accès**

  Directive `LogFormat` : Le formatage des records du journal.

  Exemple

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
```

Enregistrement des entrées de journalisation selon le format "Common Log Format" (CLF). Ce format standard peut être produit par de nombreux serveurs web différents et lu par de nombreux programmes d'analyse de journaux.

127.0.0.1 - frank [10/Oct/2015:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326

(%h)
Adresse IP du client qui a envoyé la requête.

- (%l)
Information non disponible.

(%u)
Identifiant de la personne qui a demandé le document, issu d'une authentification HTTP. (tiret si absent)

(%t)
L'heure à laquelle la requête a été reçue.

(\"%r\")
Requête du client

(%>s)
Code de statut que le serveur retourne au client.

(%b)
Taille de l'objet retourné, en-têtes non compris. (tiret si aucun contenu retourné).

Les fichiers journaux

HELHa
Haute École
Louvain en Hainaut

- **Le journal des accès**

  - Autres types de journaux d'accès possibles :

    Journalisation combinée.
    Journalisation multiple.
    Journalisation conditionnelle.

  - Autres tuning possibles :

    Rotation des journaux via des journaux redirigés.

- **Les hôtes virtuels**

  - Même journal pour tous les logs de tous les hôtes virtuels.
  - Un journal séparé pour chaque hôte virtuel.
  - Un journal unique mais pouvant être parser via un programme tel que `split-logfile`.

Plus d'info : `https://httpd.apache.org/docs/2.4/fr/logs.html`

Exercice 5

HELHa
Haute École
Louvain en Hainaut

Continuez la configuration d'Apache pour qu'il réponde aux exigences suivantes :

	Site maître	Sites virtuels
Journal des erreurs		
Localisation	`logs/error_log`	`log/error_log`
Niveau de criticité	`debug`	`warn`
Journal des accès		
Localisation	`logs/access_log`	`logs/access_log.`***`prefixe`***
Format	`combined`	`common`

Qu'est-ce que le format combiné ? → Voir l'aide sur le site d'Apache

Testez votre nouvelle configuration.

Les certificats

HELHa
Haute École
Louvain en Hainaut

- **Objectifs**

  - Pour pallier aux faiblesses du protocole http dans lequel rien n'est chiffré (y compris les échanges login/password)

  - Pour pallier à la lacune les cryptosystèmes asymétriques où, lors du partage de la clé publique, rien ne garantit que la clé est bien celle de l'utilisateur à qui elle est associée.

  - Ainsi, un certificat pourra prouver qu'une clé publique appartient bien à la personne qui prétend en être le propriétaire c'est-à-dire qui détient la clé privée correspondante.

Règle de **non-répudiation**

Les certificats

HELHa
Haute École
Louvain en Hainaut

- **SSL**

  - Secure Socket Layer

  - TLS : Transport Layer Security (nouveau standard)

  - Ensemble de bibliothèques contenant tous les algorithmes permettant de chiffrer une communication:
    . sans certificat (ex. ssh)
    . avec certificat (ex. https)

  - Exemple:

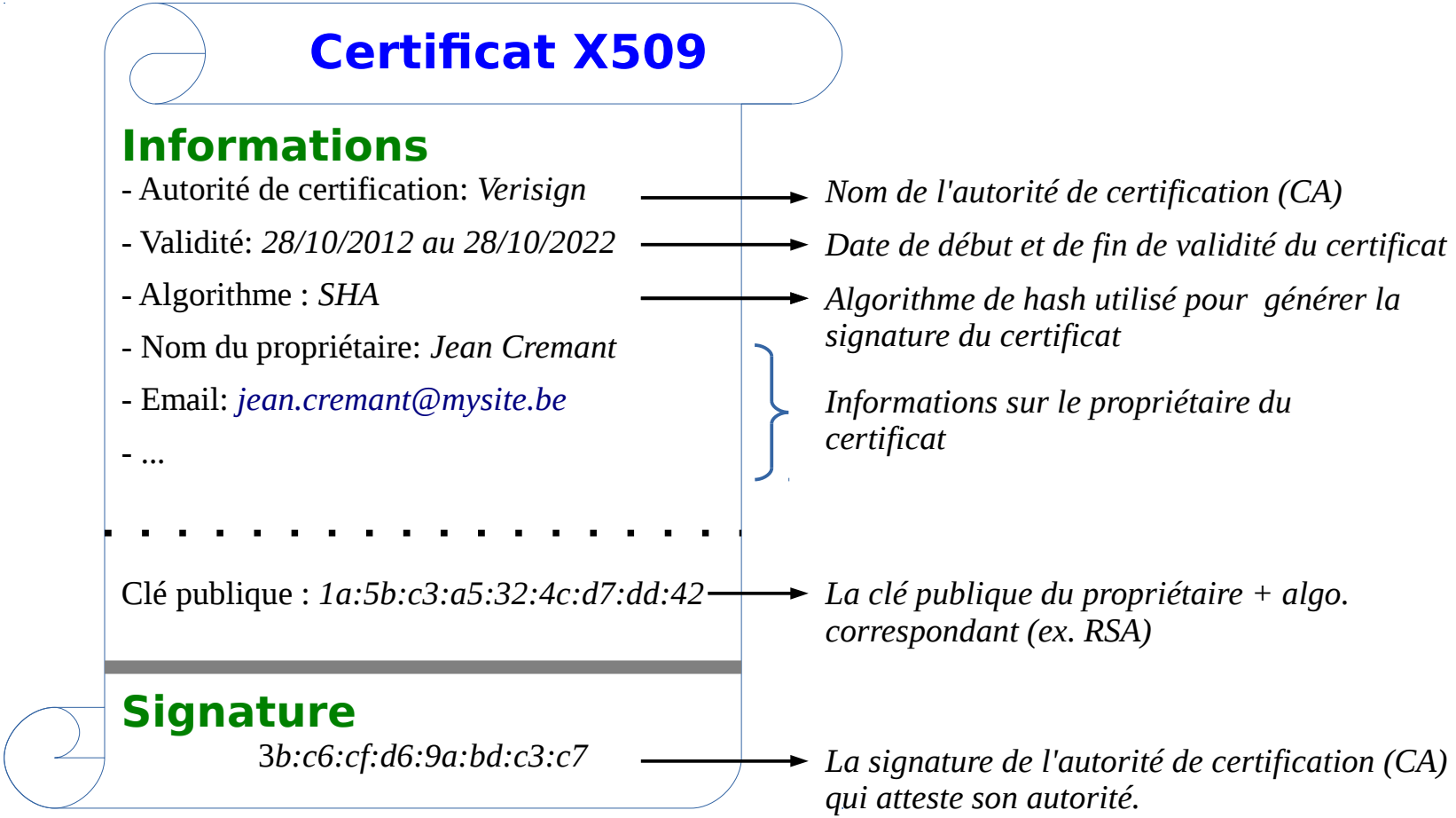
    `https://secure.mysite.be`

    Permet d'ouvrir une connexion vers le `port tcp 443` afin de négocier et d'entamer une communication http sécurisée avec le serveur.

Les certificats

HELHa
Haute École
Louvain en Hainaut

- **Structure générale (rfc5280)**



Les certificats

HELHa
Haute École
Louvain en Hainaut

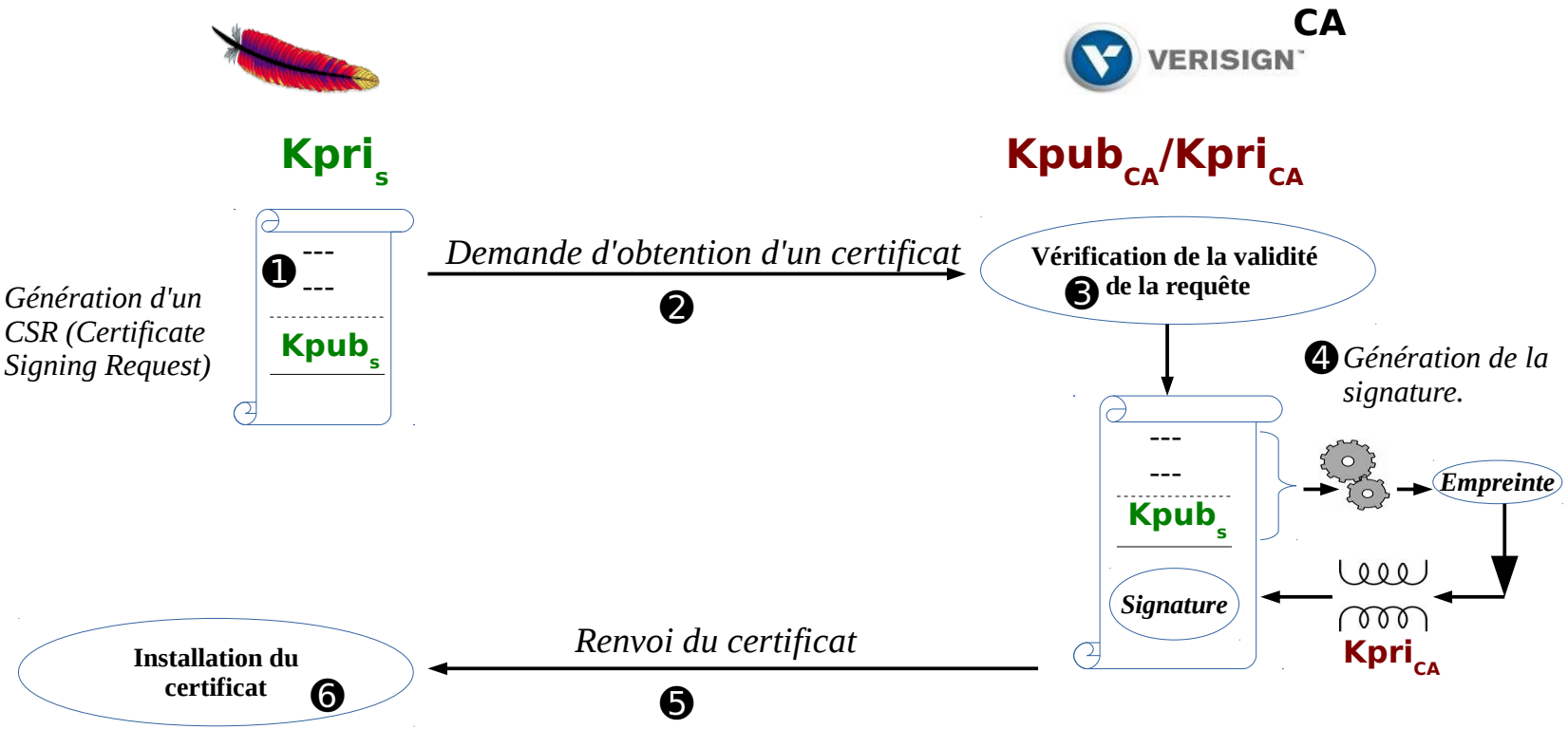
- **Fonctionnement**

  Trois intervenants :

  - Un client Web
  - Un serveur Web
  - Une autorité de certification (CA)

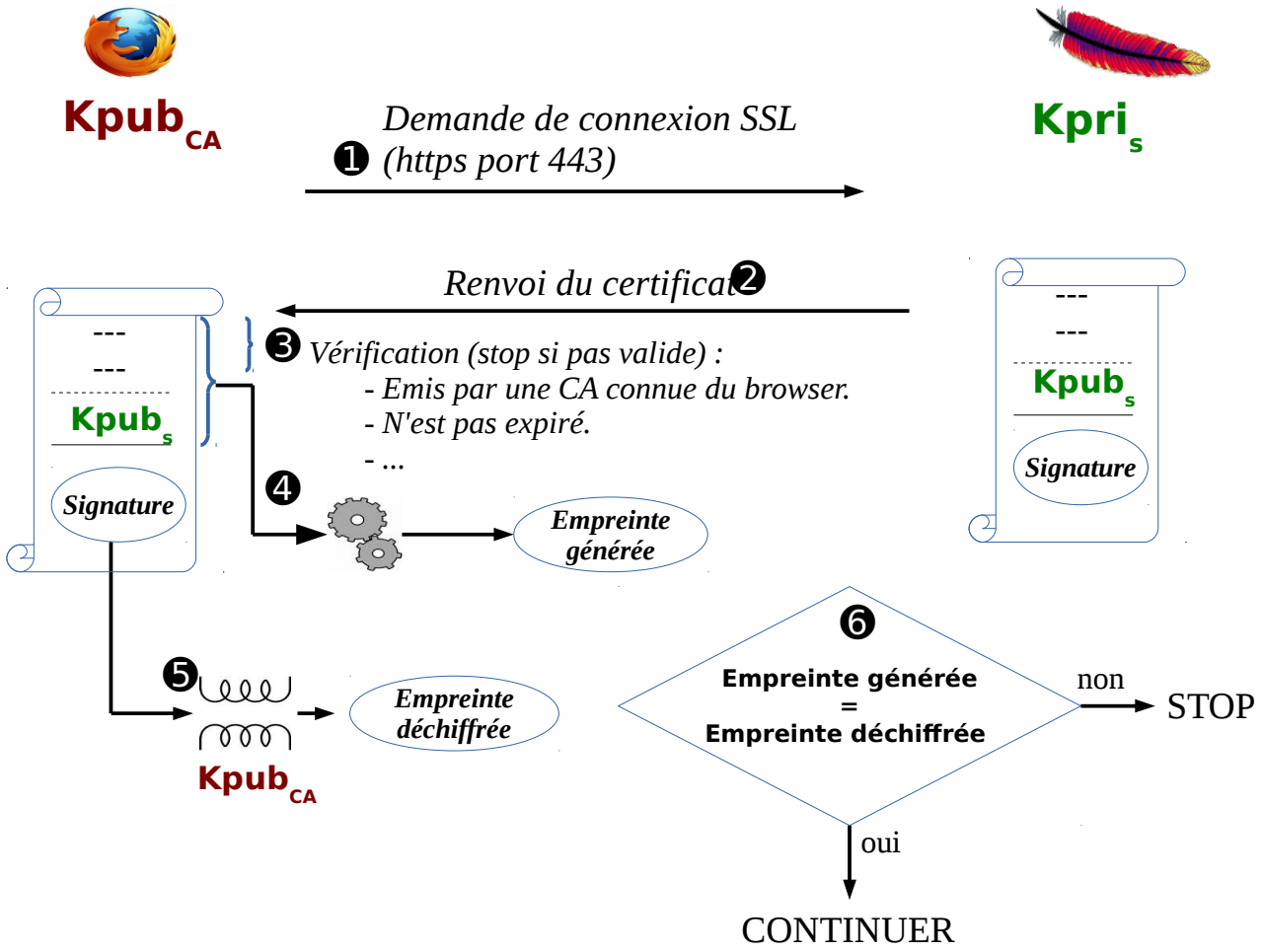
Les certificats

- **Fonctionnement : Obtention d'un certificat**



Les certificats

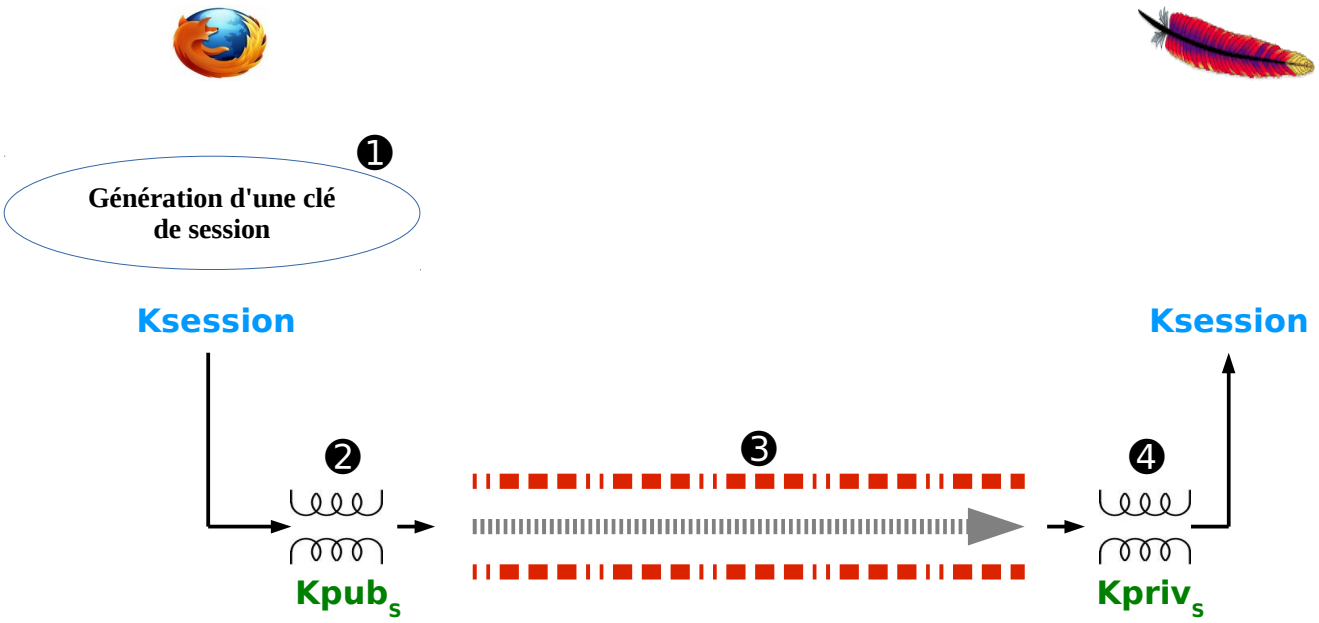
- **Fonctionnement : Connexion SSL**



Les certificats

HELHa
Haute École
Louvain en Hainaut

- **Fonctionnement : Génération d'une clé de chiffrement symétrique**



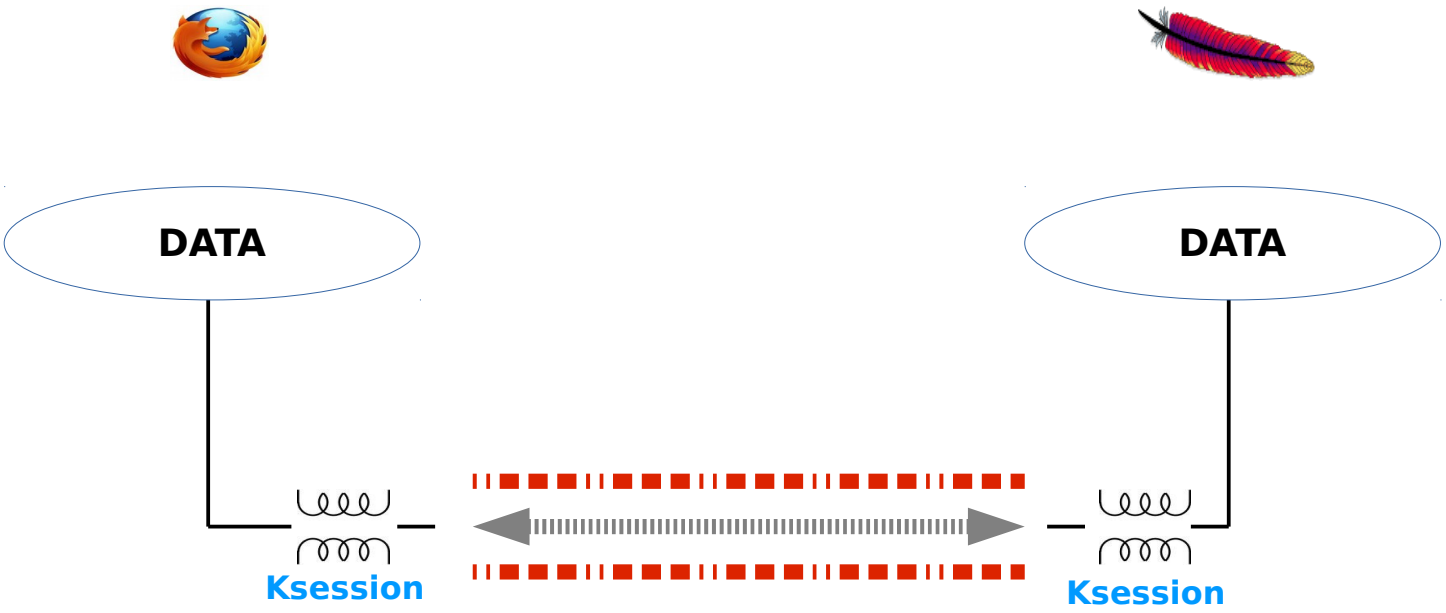
L'échange de la clé de session connaît d'autres variantes en fonction de la version du protocole SSL utilisé (SSL v2, SSL v3) ...

Plus d'info sur un échange SSL: http://httpd.apache.org/docs/trunk/fr/ssl/ssl_intro.html

Les certificats

HELHa
Haute École
Louvain en Hainaut

- **Fonctionnement : Communication SSL**



Les certificats

HELHa
Haute École
Louvain en Hainaut

- **Question de confiance**

  - Lorsqu'un client désire communiquer en SSL avec un serveur, il lui suffit de se procurer le certificat de ce dernier.

  - Ce certificat valide l'identité du serveur.

  - Ici, la règle de non-répudiation repose donc sur la confiance envers l'autorité de certification.

Les certificats

HELHa
Haute École
Louvain en Hainaut

- **Signature**

Les certificats auto-signés:

- Signés par un serveur local.
- Pour garantir la confidentialité des échanges au sein d'un intranet.

Les certificats signés par un CA:

- Pour garantir la confidentialité des échanges au sein de l'internet.
- Le certificateur tiers permet d'assurer à l'utilisateur que le certificat appartient bien à l'organisation à laquelle il est déclaré appartenir.

Les certificats

HELHa
Haute École
Louvain en Hainaut

- **Implémentation**

  Implémentation d'un certificat auto-signé

  Nous travaillerons ici avec un certificat auto- signé pour ne pas devoir l'enregistrer auprès d'une autorité de certification extérieure... (quelques centaines d'euros/an).

  Etape 1: Installer OpenSSL et le module ssl d'Apache

```
# yum install openssl mod_ssl
```

Les certificats

HELHa
Haute École
Louvain en Hainaut

- **Implémentation**

  Implémentation d'un certificat auto-signé (suite)

  Etape 2: Création de la clé privée du serveur

Clé de type rsa

Clé unique pour chaque site web SSL.
Convention : Nom du site + extension .key

2048 bits

```
# cd /tmp
# openssl genrsa -out secure.mysite.be.key 2048
# cat secure.mysite.be.key
…
```

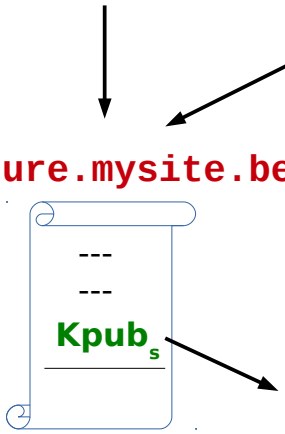
Les certificats

- **Implémentation**

  Implémentation d'un certificat auto-signé (suite)

  Etape 3: Création du CSR (Certificate Signing Request)

Kpri$_s$




```
# cd /tmp

# openssl req -new -key secure.mysite.be.key
                      -out secure.mysite.be.csr

…
Remplir tous les champs d'information
concernant le formulaire.
…

# cat secure.mysite.be.csr
…
```

C'est la clé publique correspondante à la clé privée (Kpri$_s$)

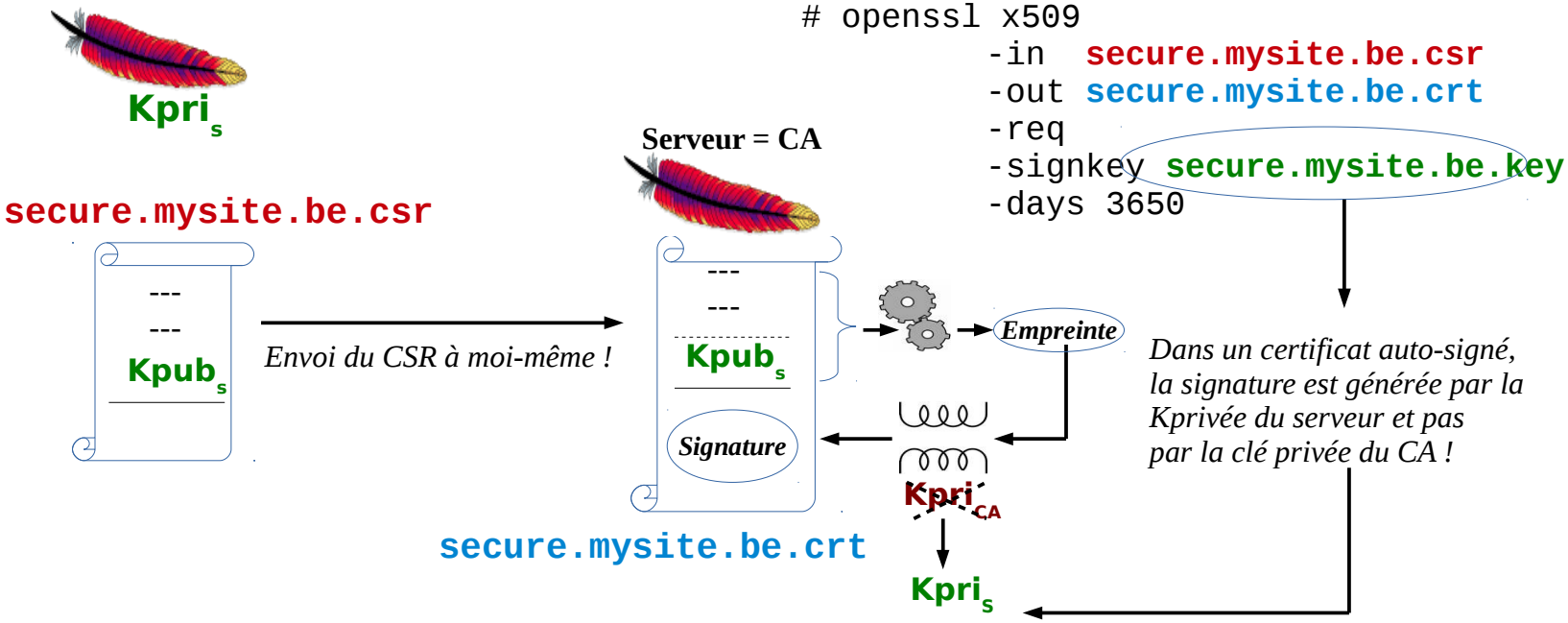
Les certificats

- **Implémentation**

  Implémentation d'un certificat auto-signé (suite)

  Etape 4: Création du certificat

```
# cd /tmp

# openssl x509
        -in  secure.mysite.be.csr
        -out secure.mysite.be.crt
        -req
        -signkey secure.mysite.be.key
        -days 3650
```



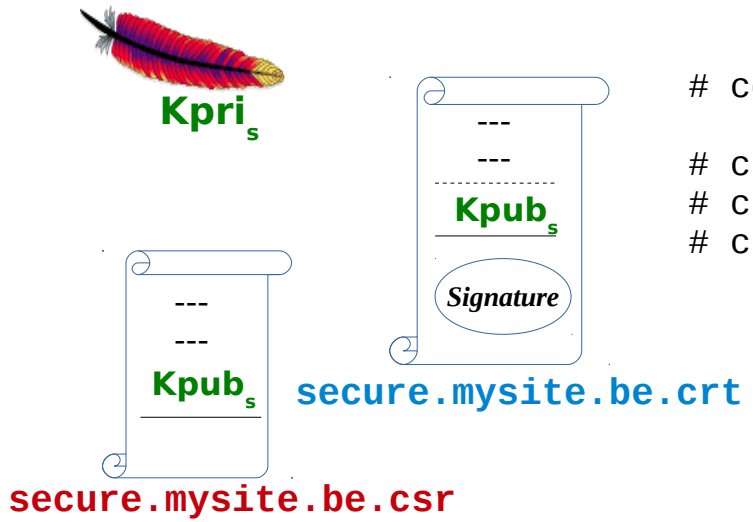
Les certificats

HELHa
Haute École
Louvain en Hainaut

- **Implémentation**

  Implémentation d'un certificat auto-signé (suite)

  Etape 5: Copie des clés et des certificats au bon endroit



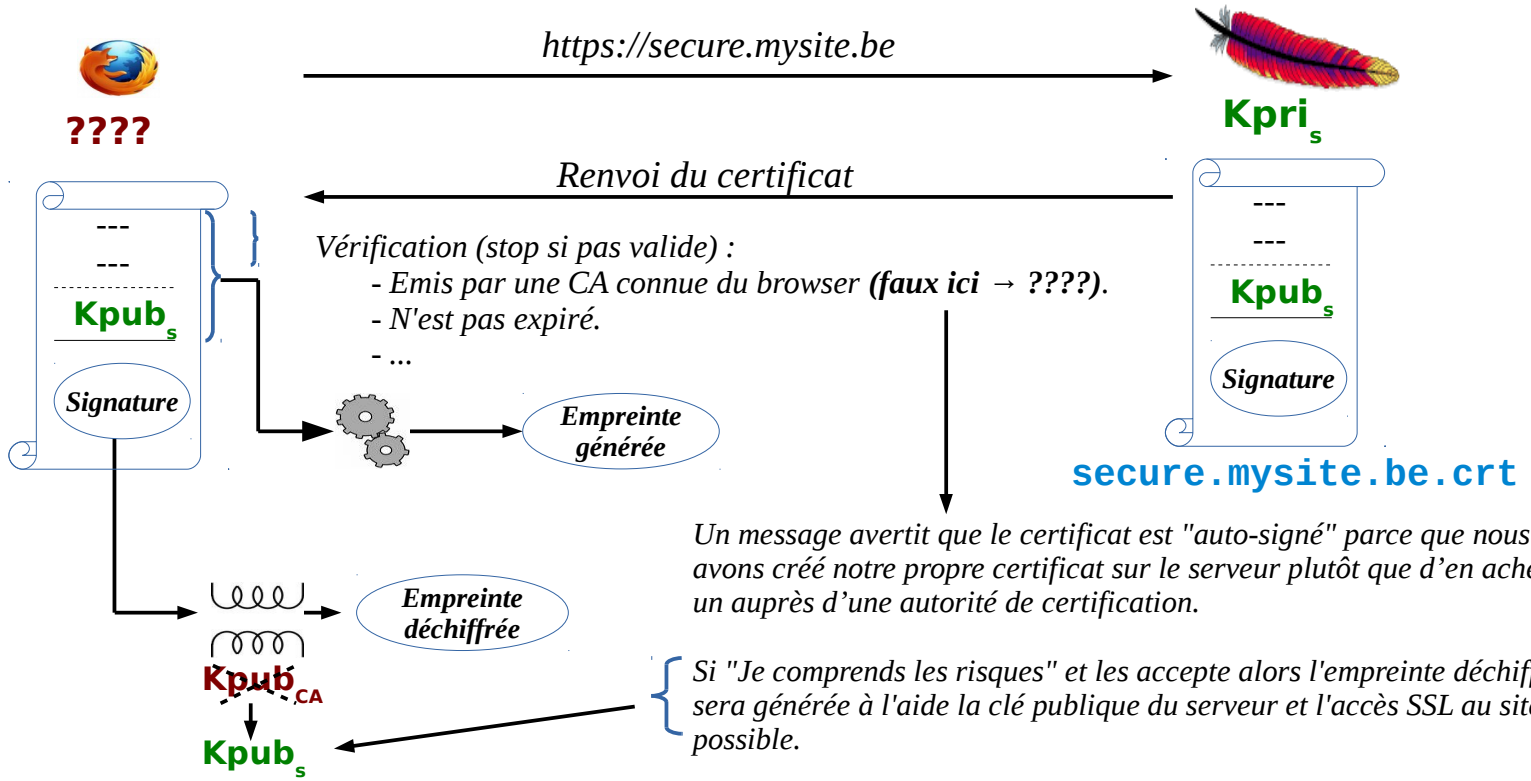

```
# cd /tmp

# cp secure.mysite.be.crt /etc/pki/tls/certs
# cp secure.mysite.be.csr /etc/pki/tls/private
# cp secure.mysite.be.key /etc/pki/tls/private
```

Les certificats

- **Implémentation**

  Implémentation d'un certificat auto-signé (suite)

  Etape 6: Connexion SSL



Les certificats

HELHa
Haute École
Louvain en Hainaut

- **Implémentation**

  Remarques

  - Pour ne pas devoir "Comprendre les risques" lors de chaque connexion, il faut importer le certificat (.crt) du serveur dans le navigateur client.

  - Lire le contenu d'un certificat: `# openssl x509 -noout -text -in mycertif.crt`

  - Soumettre son .csr à un CA:

    . Choisir un CA reconnu par tous les navigateurs (ex. Verisign). On disposera alors de sa clé publique.

    . Se connecter sur la page d'enregistrement du CA.

    . Copier/Coller son .csr dans le formulaire de cette page.

    . Répondre à quelques questions supplémentaires.

    . Réaliser le paiement.

    . Le certificat sera envoyé par mail dans les jours qui suivent.

Les certificats

HELHa
Haute École
Louvain en Hainaut

- **Implémentation**

  Configuration d'Apache

  - Soit: . `https://secure.mysite.be` sur `192.168.1.100` (hôte virtuel)
    . Page d'accueil dans `/var/www/html/secure`

  - Configurer `/etc/httpd/conf.d/01-ssl.conf`

```
<VirtualHost 192.168.1.100:443>
   ServerName secure.mysite.be
   DocumentRoot /var/www/html/secure
   SSLEngine on → Pour spécifier que SSL doit être utilisé pour un hôte virtuel et pas pour le serveur.
   SSLCertificateFile /etc/pki/tls/certs/secure.mysite.be.crt
                                  → Nom complet du certificat pour ce site.
   SSLCertificateKeyFile /etc/pki/tls/private/secure.mysite.be.key
                                  → Nom complet de la clé privée pour ce site.
</VirtualHost>
```

Les certificats

HELHa
Haute École
Louvain en Hainaut

- **Implémentation**

  Configuration d'Apache

  - Vérifier dans `/etc/httpd/conf.modules.d/00-ssl.conf`

    …
    `LoadModule ssl_module modules/mod_ssl.so` → *Nécessaire aux certificats.*

  - Configurer `/etc/httpd/conf.d/01-ssl.conf`

    …
    `Listen 443` → *Apache écoutera également sur le port 443 de toutes ses interfaces.*
    …
    `SSLPassPhraseDialog builtin` → *Une passphrase éventuelle sera demandée sur l'entrée standard.*
    `SSLSessionCacheTimeout 300` → *Timeout pour les sessions SSL (300 secondes).*
    …

    → Toutes ces directives générales seront exécutées au démarrage d'Apache (ne rien changer).

    Certaines d'entre elles sont redéfinies dans l'hôte virtuel (voir ci-avant)

Les certificats

HELHa
Haute École
Louvain en Hainaut

- **La redirection SSL**

Pourquoi ?

- Obliger une navigation sécurisée.
- Un des facteurs de référencement de Google.

Exemple 1

Toute requête à destination de la ressource `/secure` se trouvant dans `www.mysite.be` sera redirigée vers la page d'accueil se trouvant dans `/var/www/html/secure`. A partir de ce moment, la navigation SSL sera initiée.

```
<VirtualHost 192.168.1.1:80>
   ServerName www.mysite.be
   DocumentRoot /var/www/html
   Redirect permanent /secure https://www.mysite.be/secure
</VirtualHost>

<VirtualHost 192.168.1.1:443>
   ServerName www.mysite.be
   DocumentRoot /var/www/html/secure
   SSLEngine on
   ...
</VirtualHost>
```

La redirection est 'permanente' c'est-à-dire que le client recevra un code de retour 301 lui indiquant que la ressource a été déplacée définitivement (il devrait donc accéder à cette URL via le nom `www.mysite.be/secure`).

Les certificats

HELHa
Haute École
Louvain en Hainaut

- **La redirection SSL**

Exemple 2

```
<VirtualHost 192.168.1.1:80>
   ServerName www.mysite.be
   DocumentRoot /var/www/html
   Redirect permanent / https://www.mysite.be
</VirtualHost>

<VirtualHost 192.168.1.1:443>
   ServerName www.mysite.be
   DocumentRoot /var/www/html
   SSLEngine on
   ...
</VirtualHost>
```

Toute requête à destination de n'importe quelle ressource du site sera initiée en SSL.

Exercice 6

HELHa
Haute École
Louvain en Hainaut

Continuez la configuration d'Apache pour que:

- la page d'accueil de `www.mysite.be` offre en plus la possibilité d'accéder à un secret via un lien `"Cliquez ici pour voir le secret"`.
- un clic sur ce lien nous redirige vers la page d'accueil de `secure.mysite.be` d'un hôte virtuel SSL hébergé par Ip sur `192.168.27.100`. Cette page sera enregistrée dans `/var/www/html/websecure` et contiendra le secret.
  - . Importez le certificat dans Firefox de façon à ne plus avoir à "Comprendre les risques" lors de l'accès au site SSL.
  - . Tracez à l'aide d'un renifleur toute la conversation de `www.mysite.be` jusque `secure.mysite.be`.
- les requêtes `http://vh3.mysite.be` et `https://vh3.mysite.be` nous offrent, de manière sécurisée, la page d'accueil du site.
  - . générez un nouveau certificat auto-signé pour ce site
  - . les 2 requêtes seront envoyées sur l'Ip `192.168.27.2`
  - . vérifiez par les logs de `access_log.vh3` que:

    `http://vh3.mysite.be` génère 2 logs. Pourquoi ?
    `http://vh3.mysite.be` génère 1 log. Pourquoi ?

Références

HELHa
Haute École
Louvain en Hainaut

WEBOGRAPHIE

http://christian.caleca.free.fr/http
http://www.linux-france.org/prj/edu/archinet/systeme
http://www.commentcamarche.net/contents/crypto
http://ww2.ac-creteil.fr/reseaux/systemes/linux/index.html
http://httpd.apache.org/docs/2.4/

BIBLIOGRAPHIE

Guide de référence 'Apache 2' - JM CULOT - OEM Eyrolles

Apache 2.0 - Guide de l'administrateur Linux - C. AULDS - Eyrolles