



**JOSY « Authentification Centralisée »
Paris, 6 mai 2010**

CASifier une application

Julien Marchal

CASifier une application

- **Introduction**
- **Moyens**
 - Module Apache CAS (mod_auth_cas)
 - Librairie PHP (phpCAS)
 - Filtre Java
 - Module PAM (Pam_cas)
- **Problèmes courant**

Introduction

- **Quelques principes**
 - Le serveur CAS doit être accessible
 - Les applications doivent maintenir des sessions
 - Les applications doivent pouvoir gérer les connexions SSL avec le CAS

Module Apache CAS (mod_auth_cas)

- **Module Apache 2.x**
- **L'identifiant utilisateur se retrouve dans REMOTE_USER**
- **Ne gère pas le logout**

Module Apache CAS (mod_auth_cas)

```
LoadModule auth_cas_module modulesN2/mod_auth_cas.so
```

```
<IfModule mod_auth_cas.c>
```

```
    CASVersion 2
```

```
    CASDebug Off
```

```
    CASValidateServer Off
```

```
    CASLoginURL https://auth.univ.fr
```

```
    CASValidateURL https://auth.univ.fr/serviceValidate
```

```
    CASCookiePath /var/cas/
```

```
    CASTimeout 7200
```

```
    CASIdleTimeout 3600
```

```
    CASCacheCleanInterval 1800
```

```
</IfModule>
```

Module Apache CAS (mod_auth_cas)

```
<IfModule mod_auth_cas.c>  
    AuthName "Authentification centrale"  
    AuthType CAS  
</IfModule>  
  
require user uid1 uid2 ...
```

Librairie PHP (phpCAS)

- **Module PHP > 4.2**
 - Extension curl
 - Extension openssl
 - Extension DOM (xml pour PHP4)
 - Extension Zlib (PHP4)
- **Gère le logout**
- **Gère la récupération d'attribut (SAML)**

Librairie PHP (phpCAS)

```
<?  
include_once('CAS.php');  
  
phpCAS::client(CAS_VERSION_2_0, 'auth.univ.fr',443,"");  
phpCAS::handleLogoutRequests(true, array("auth.univ.fr"));  
  
phpCAS::setNoCasServerValidation();  
phpCAS::forceAuthentication();  
  
echo phpCAS::getUser();  
>
```

Filtre Java

- **5 Filtres :**
 - Authentication Filter
 - Validation Filter
 - HttpServletRequest Wrapper Filter
 - Assertion Thread Local Filter
 - Single Sign Out Filter
- **1 Session Listener :**
 - SingleSignOutHttpSessionListener

Filtre Java

- **Authentication Filter**

- Va intercepter les clients et les renvoyer au serveur CAS si il ne sont pas déjà authentifiés

```
<filter>
```

```
<filter-name>CAS Authentication Filter</filter-name>
```

```
<filter-class>org.jasig.cas.client.authentication.AuthenticationFilter</filter-class>
```

```
<init-param>
```

```
<param-name>casServerLoginUrl</param-name>
```

```
<param-value>https://auth.univ.fr/login</param-value>
```

```
</init-param>
```

```
</filter>
```

Filtre Java

- **Validation Filter**

- Va intercepter les tickets et les valider auprès du CAS

```
<filter>
```

```
<filter-name>CAS Validation Filter</filter-name>
```

```
<filter-class>
```

```
org.jasig.cas.client.validation.Cas20ProxyReceivingTicketValidationFilter
```

```
</filter-class>
```

```
<init-param>
```

```
<param-name>casServerUrlPrefix</param-name>
```

```
<param-value>https://auth.univ.fr</param-value>
```

```
</init-param>
```

```
<init-param>
```

```
<param-name>serverName</param-name>
```

```
<param-value>http://application.univ.fr</param-value>
```

```
</init-param>
```

```
</filter>
```

Filtre Java

- **HttpServletRequest Wrapper Filter**

- Va mettre à disposition l'identifiant utilisateur dans `HttpServletRequest.getRemoteUser ()`

```
<filter>
```

```
  <filter-name>CAS HttpServletRequest Wrapper Filter</filter-name>
```

```
  <filter-class>
```

```
    org.jasig.cas.client.util.HttpServletRequestWrapperFilter
```

```
  </filter-class>
```

```
</filter>
```

Filtre Java

- **Assertion Thread Local Filter**

- Va mettre à disposition l'identifiant utilisateur dans un Thread local pour les classes ne pouvant pas accéder à HttpServletRequest

```
<filter>
```

```
  <filter-name>CAS Assertion Thread Local Filter</filter-name>
```

```
  <filter-class>
```

```
    org.jasig.cas.client.util.AssertionThreadLocalFilter
```

```
  </filter-class>
```

```
</filter>
```

Filtre Java

- **Single Sign Out Filter**

- Va intercepter les demandes de logout

```
<filter>
```

```
  <filter-name>CAS Single Sign Out Filter</filter-name>
```

```
  <filter-class>org.jasig.cas.client.session.SingleSignOutFilter</filter-class>
```

```
</filter>
```

- **SingleSignOutHttpSessionListener**

- Va surveiller les sessions utilisateurs afin de déloguer lors de l'expiration

```
<listener>
```

```
  <listener-class>
```

```
    org.jasig.cas.client.session.SingleSignOutHttpSessionListener
```

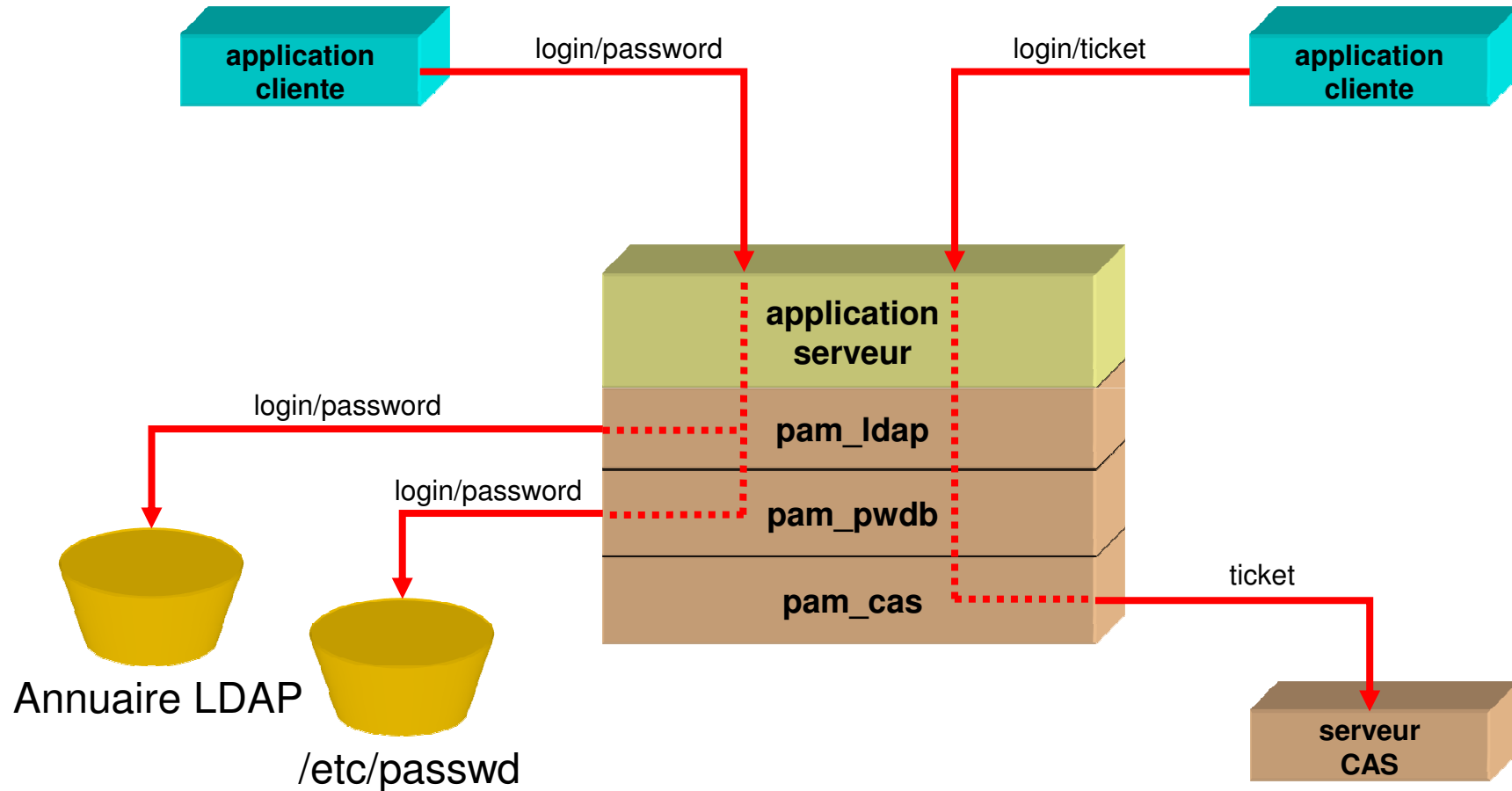
```
  </listener-class>
```

```
</listener>
```

Module PAM (Pam_cas)

- **Pluggable Authentication Modules**
- **Est utilisé dans les mecanismes n-tiers (imap, ftp, ...)**

Module PAM (Pam_cas)



Module PAM (Pam_cas)

- Pour un serveur imap par exemple :

/etc/pam.d/imap

```
auth sufficient /lib/security/pam_cas.so -simap://imap.univ.fr -f/etc/pam_cas.conf
auth sufficient /lib/security/pam_ldap.so
auth required /lib/security/pam_pwdb.so shadow nullok
account required /lib/security/pam_pwdb.so shadow nullok
```

Module PAM (Pam_cas)

- Pour un serveur imap par exemple :

/etc/pam.d/imap

```
auth sufficient /lib/security/pam_cas.so -simap://imap.univ.fr -f/etc/pam_cas.conf
auth sufficient /lib/security/pam_ldap.so
auth required /lib/security/pam_pwdb.so shadow nullok
account required /lib/security/pam_pwdb.so shadow nullok
```

Module PAM (Pam_cas)

/etc/pam_cas.conf

host auth.univ.fr

uriValidate /proxyValidate

ssl off

debug off

trusted_ca /Cert/mycert.pem

Des questions ?

- <http://www.ja-sig.org/wiki/display/CASC/Home>
- http://www.ja-sig.org/wiki/display/CASC/mod_auth_cas
- <http://www.ja-sig.org/wiki/display/CASC/phpCAS>
- http://www.esup-portail.org/consortium/espace/SSO_1B/tech/cas/cas_pam.html