Single Sign-On open source avec CAS (Central Authentication Service)

Vincent Mathieu

Pascal Aubry

Julien Marchal

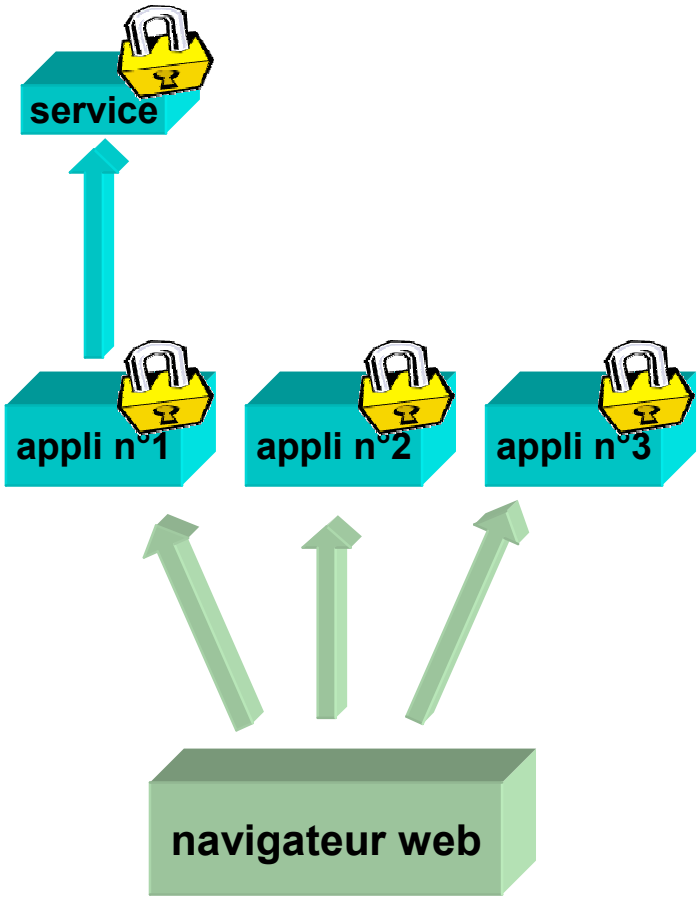
SSO open source avec CAS

- **Introduction**
  - Pourquoi le Single Sign-On ?
  - Principes du SSO sur le web
  - Le choix de CAS
- **Le mécanisme CAS**
- **L'authentification sous CAS**
- **CAS-ification d'une application**
  - applications web
  - applications non web
- **CAS aujourd'hui et demain**
- **Démonstration**

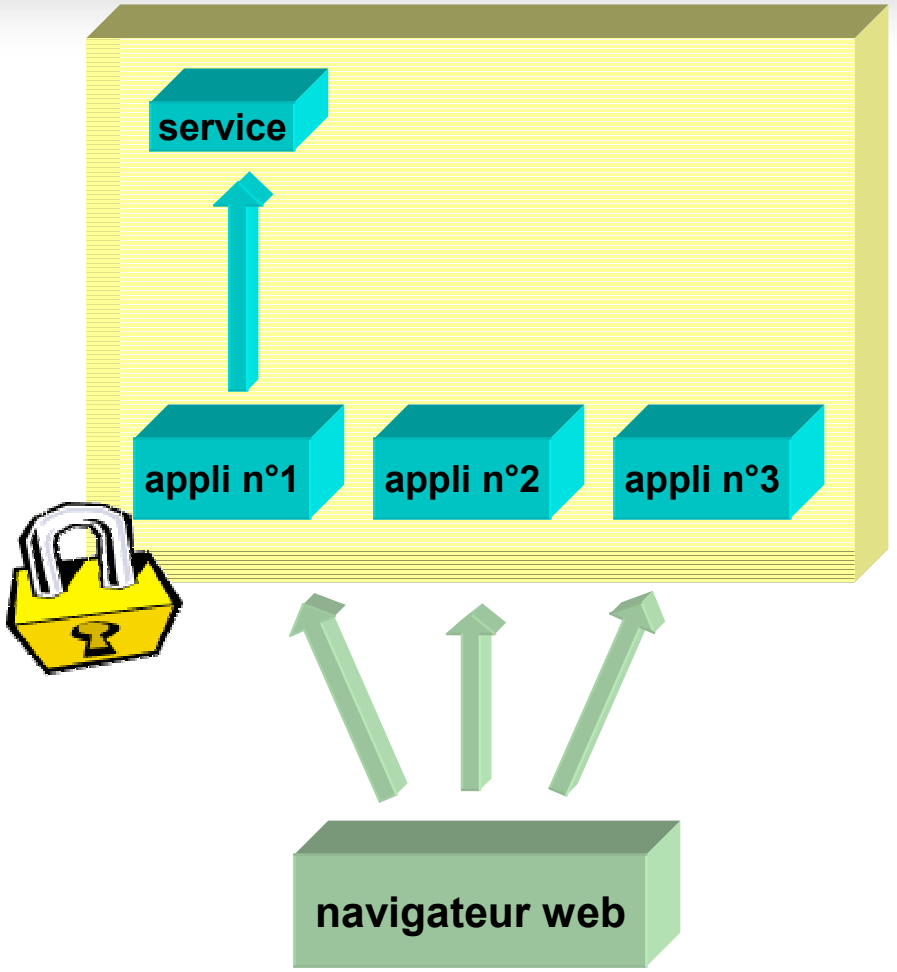
Pourquoi le Single Sign-On ?

- **Single Sign-On = Authentification unique et unifiée**
- **Authentifications multiples**
- **Sécurité**
  - Le vol d'un mot de passe unique est critique
    - Protéger le mot de passe
    - Ne pas le transmettre aux applications
      (simplification des applications et non délégation de la sécurité)
- **Différents mécanismes d'authentification**
  - Abstraction du mode d'authentification
    - LDAP, NIS, BDD, certificats X509, …

Pourquoi le Single Sign-On ?



sans le SSO



avec le SSO

Principes du SSO web

- **Centralisation de l'authentification**
  - Sur un serveur (d'authentification)
- **Redirections HTTP transparentes**
  - Des applications vers le serveur d'authentification
  - Du serveur d'authentification vers les applications
- **Passage d'informations lors de ces redirections**
  - Cookies
  - Paramètres CGI

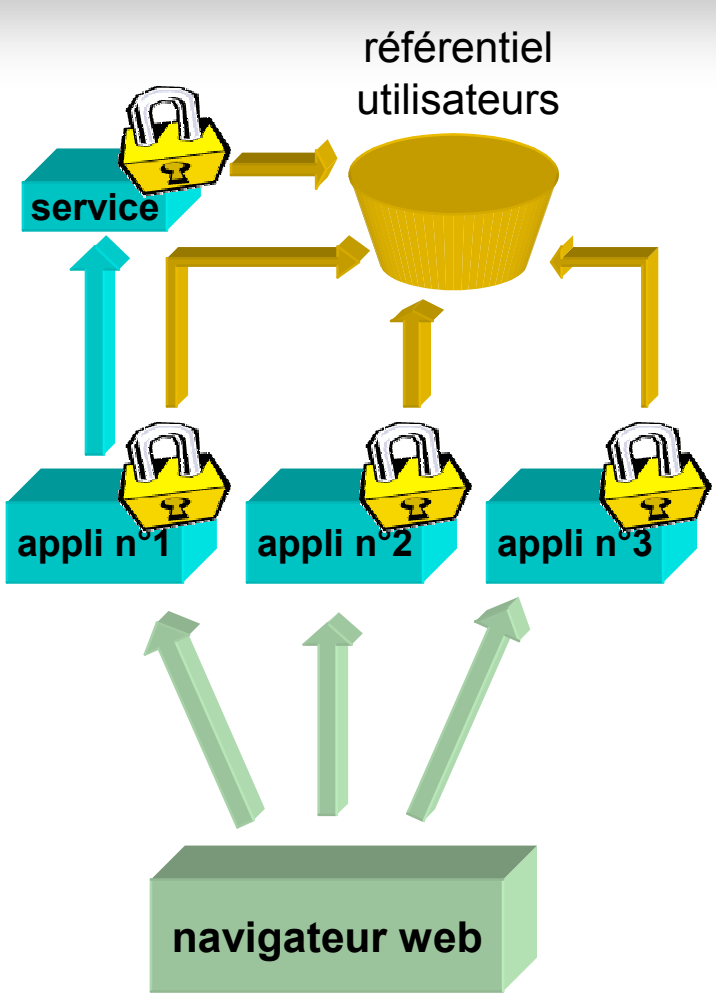
Le choix de CAS par ESUP-Portail (1)

- **Sécurité**
  - Le mot de passe n'est transmis qu'au serveur d'authentification
  - Utilisation de tickets « opaques » et à usage unique (à la *Kerberos*)
- **Mécanisme n-tiers**
  - Utilisation de services sans transmission du mot de passe
- **Portabilité (librairies clientes)**
  - Java, Perl, JSP, ASP, PHP, PL/SQL, modules apache et PAM
  - Adaptation aisée des applications

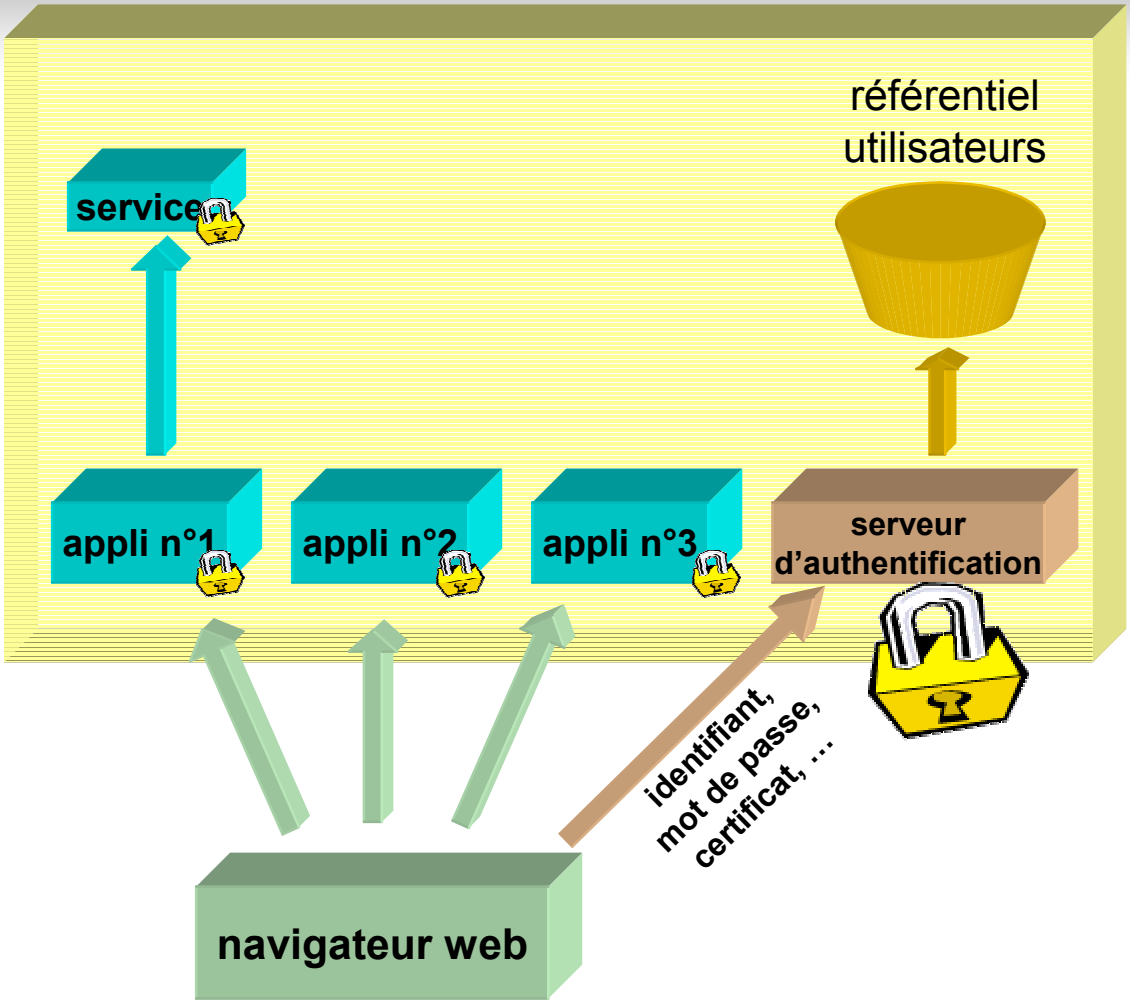
Le choix de CAS par ESUP-Portail (2)

- **Pérennité**
  - Développé par l'Université de Yale
  - En production dans les universités (américaines notamment)
- **Plateforme J2EE**
  - Code léger (un millier de lignes de code)
- **Open source**
- **Intégration uPortal**

Le choix de CAS



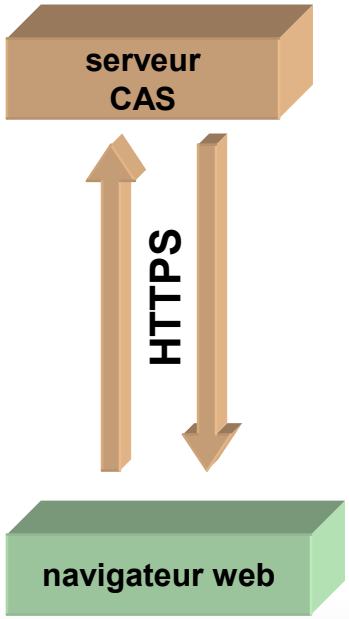
sans le SSO



avec CAS

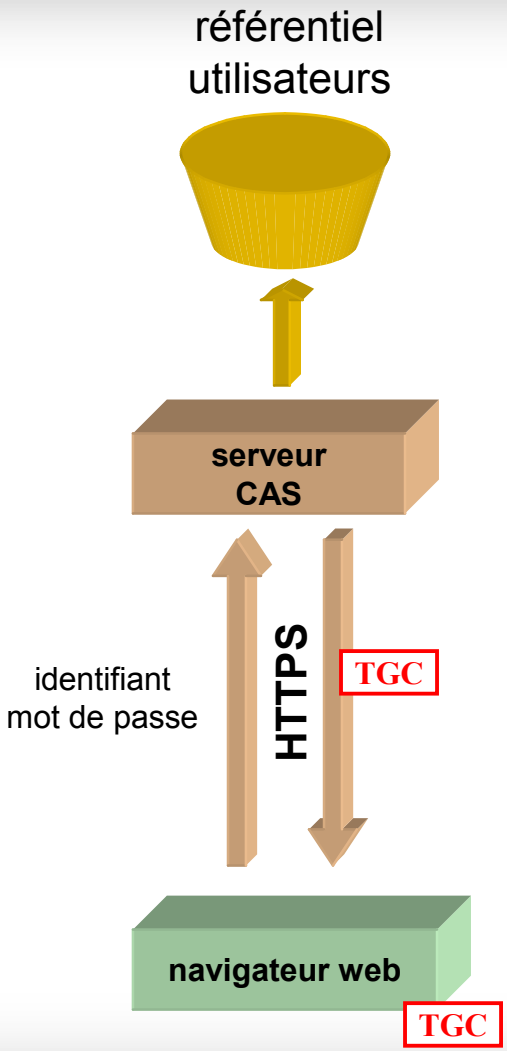
Mais comment ça marche ?

1ère authentification d'un utilisateur



formulaire
d'authentification :

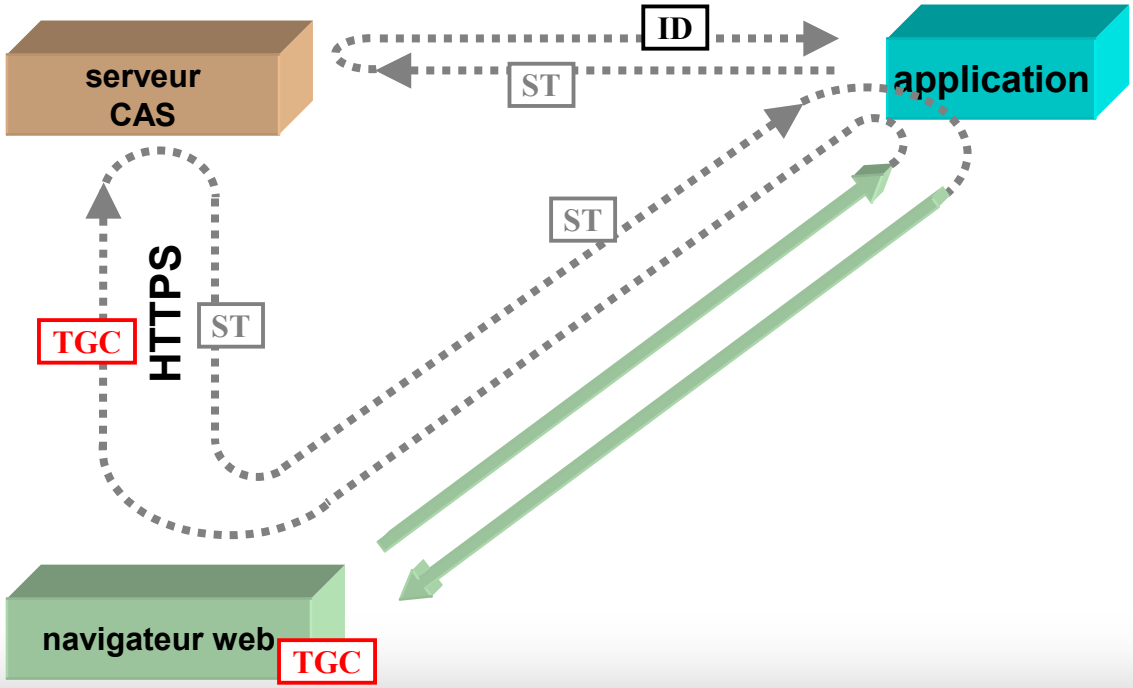
1ère authentification d'un utilisateur



- **TGC : Ticket Granting Cookie**
  - Passeport du navigateur auprès du serveur CAS
  - Cookie privé et protégé
    (le seul cookie utilisé dans CAS ;
    il n'est pas obligatoire)
  - Ticket opaque rejouable

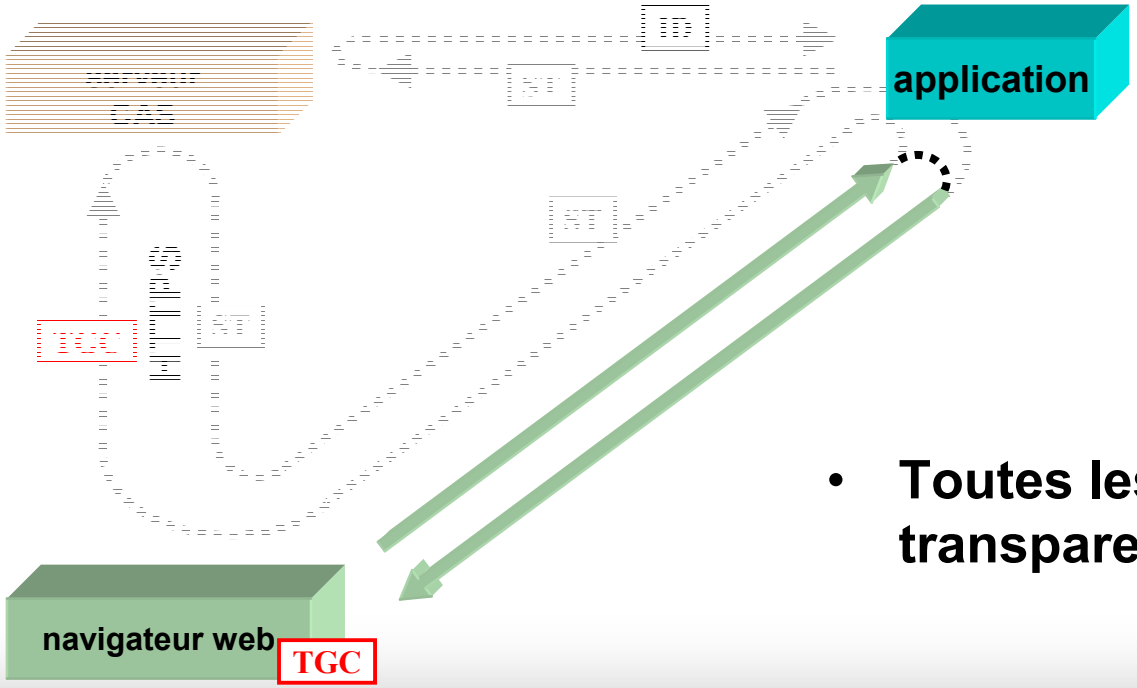
Accès à une application (après authentification)

- **ST : Service Ticket**
  - Passeport du navigateur auprès du client CAS
  - Ticket opaque non rejouable
  - Limité dans le temps



Accès à une application (après authentification)

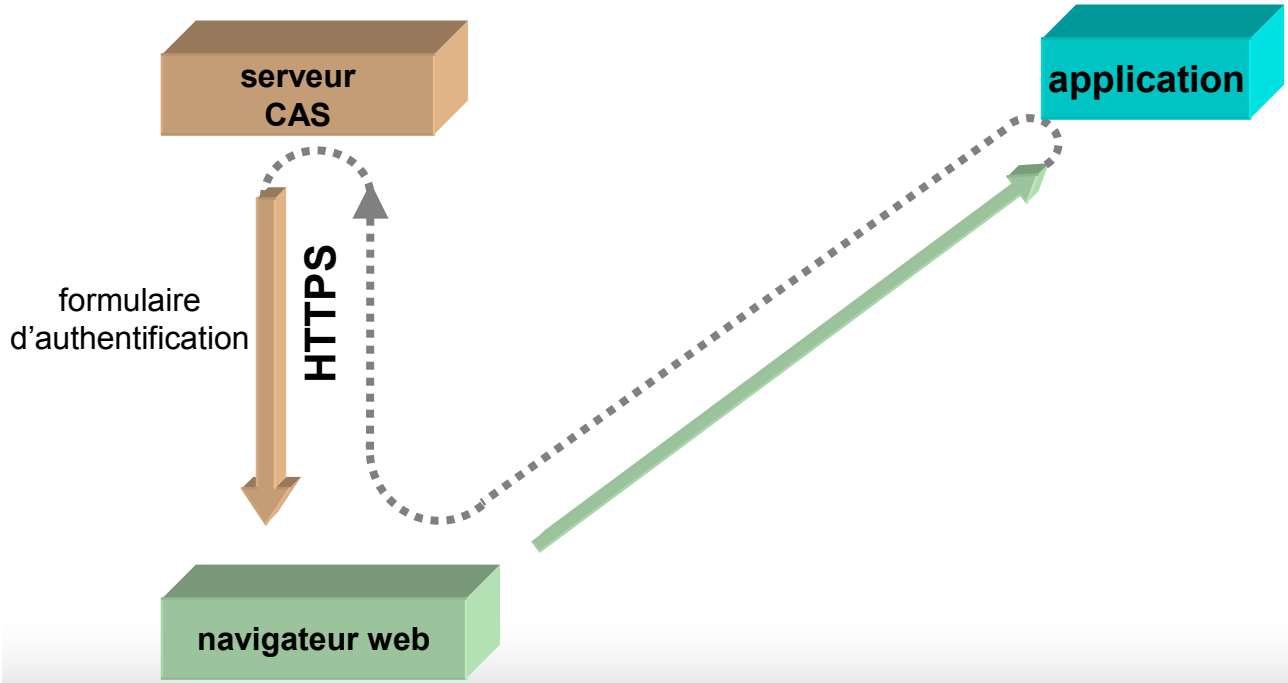
- **ST : Service Ticket**
  - Passeport du navigateur auprès du client CAS
  - Ticket opaque non rejouable
  - Limité dans le temps



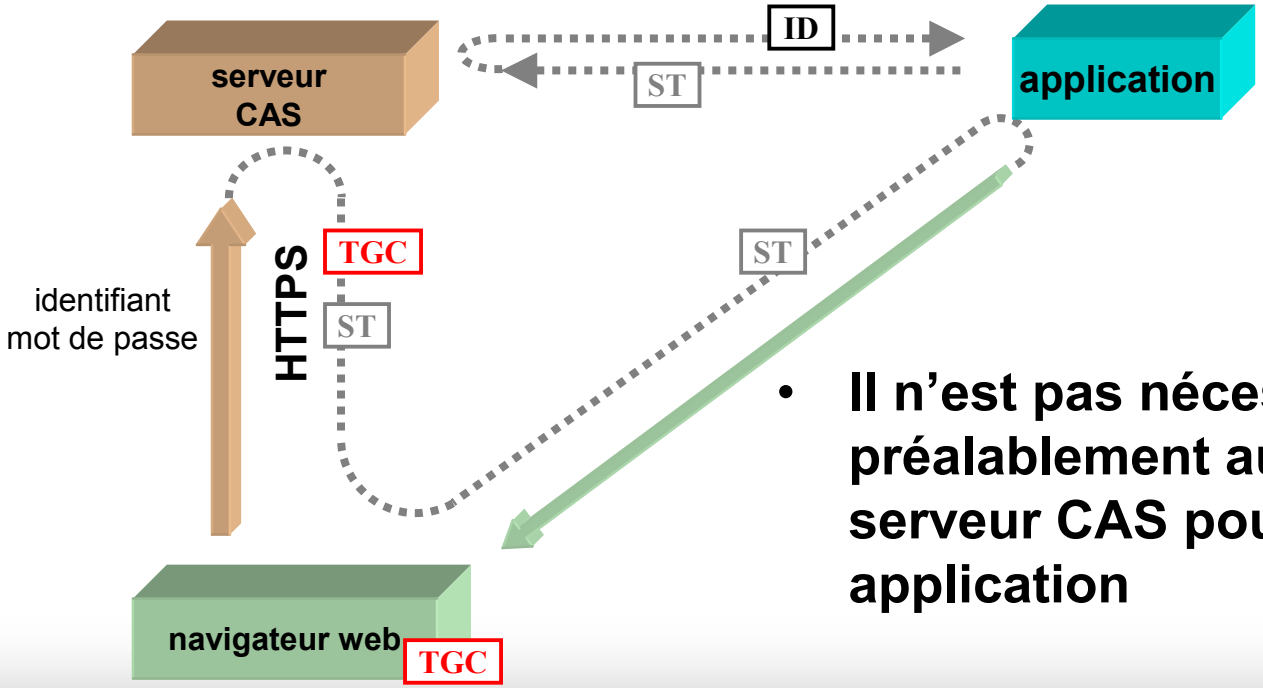
- **Toutes les redirections sont transparentes pour l'utilisateur**

Dans la pratique…

Accès à une application (avant authentification)



Accès à une application (avant authentification)



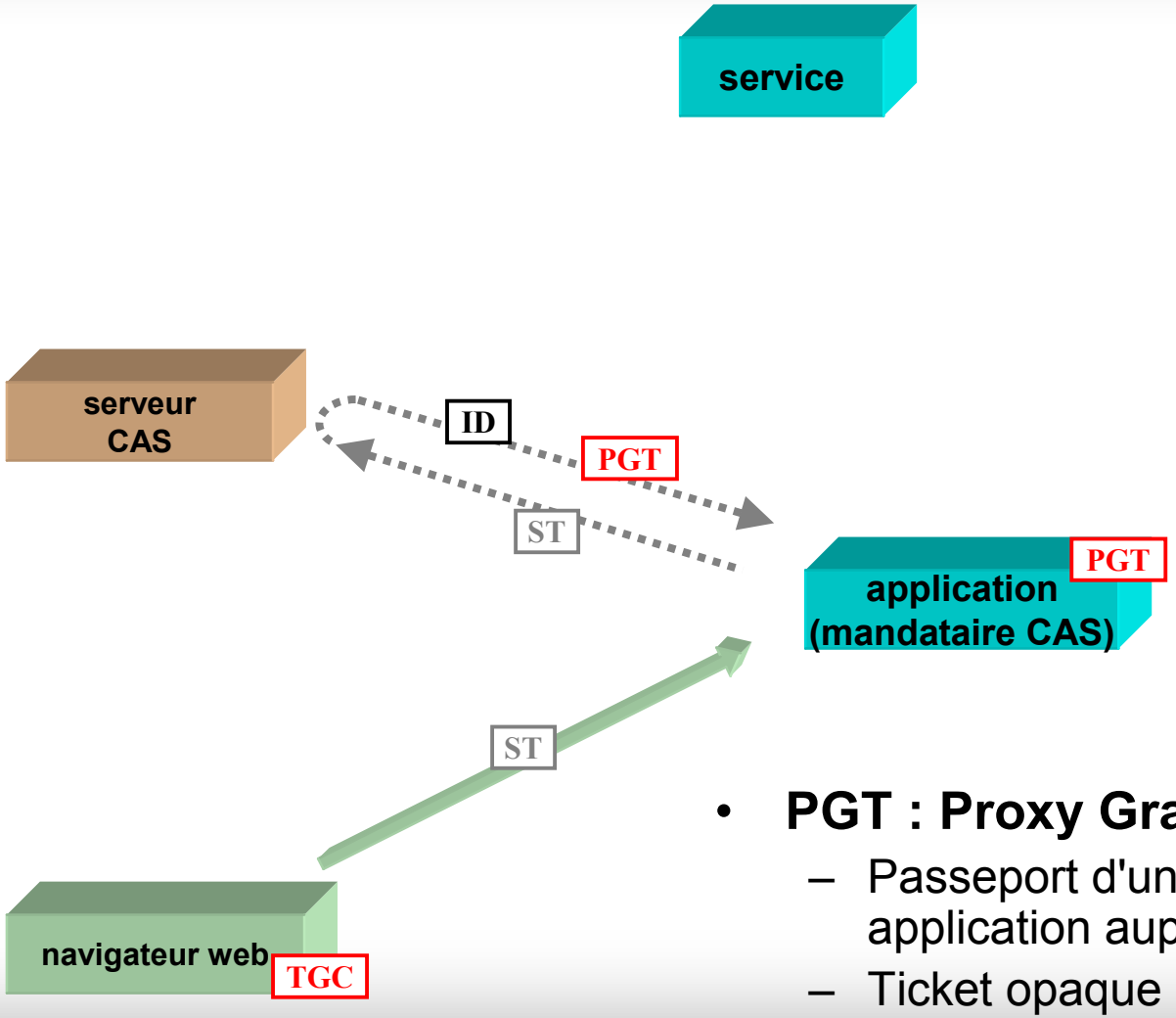
- **Il n'est pas nécessaire de s'être préalablement authentifié auprès du serveur CAS pour accéder à une application**

Quelques remarques…

Remarques

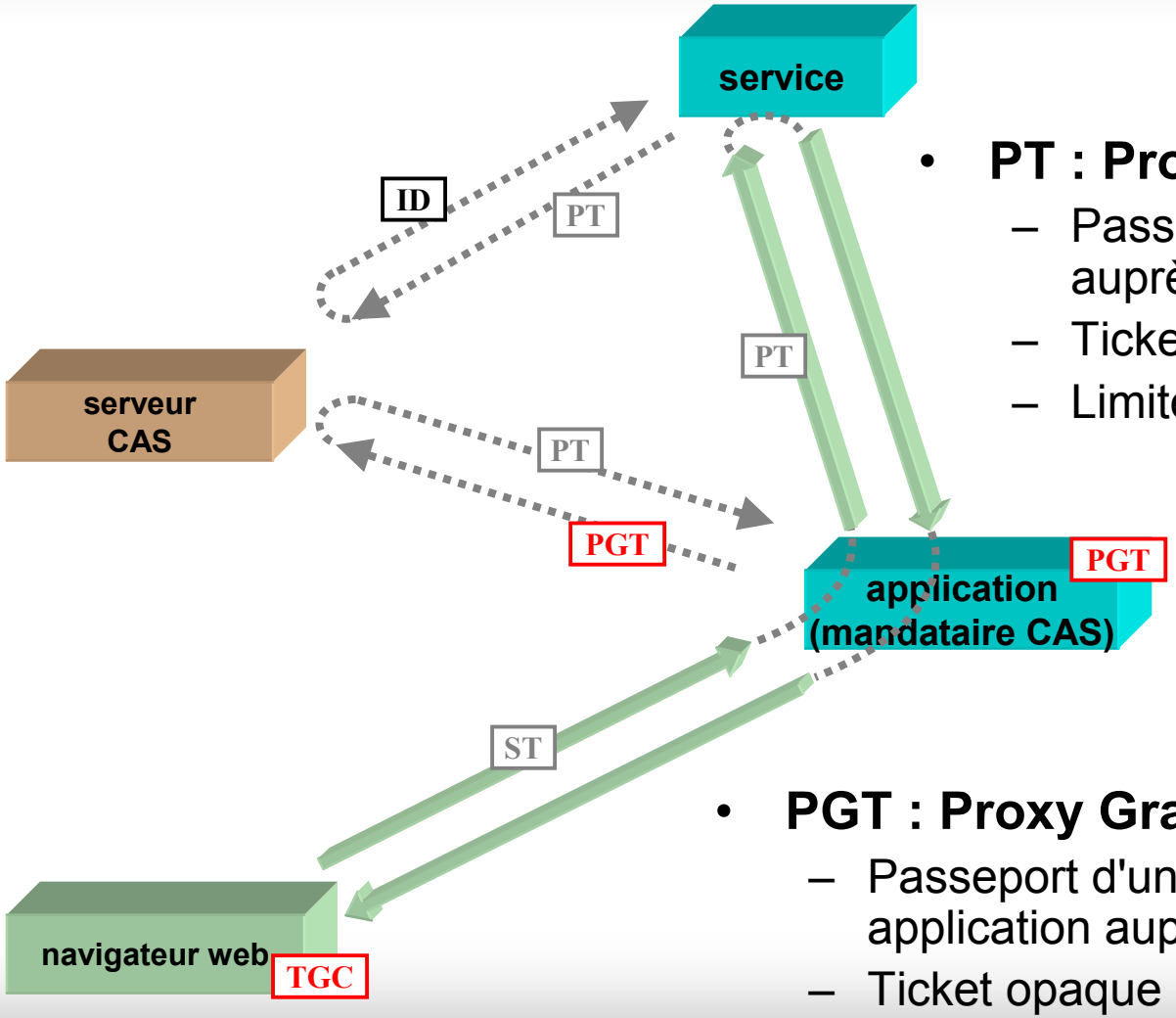
- **Une fois le TGC acquis, l'authentification devient transparente pour l'accès à toutes les autres applications CAS-ifiées**
- **Une fois authentifié pour une application, une session applicative est mise en place**

Fonctionnement n-tiers



- **PGT : Proxy Granting Ticket**
  - Passeport d'un utilisateur pour une application auprès du serveur CAS
  - Ticket opaque rejouable

Fonctionnement n-tiers

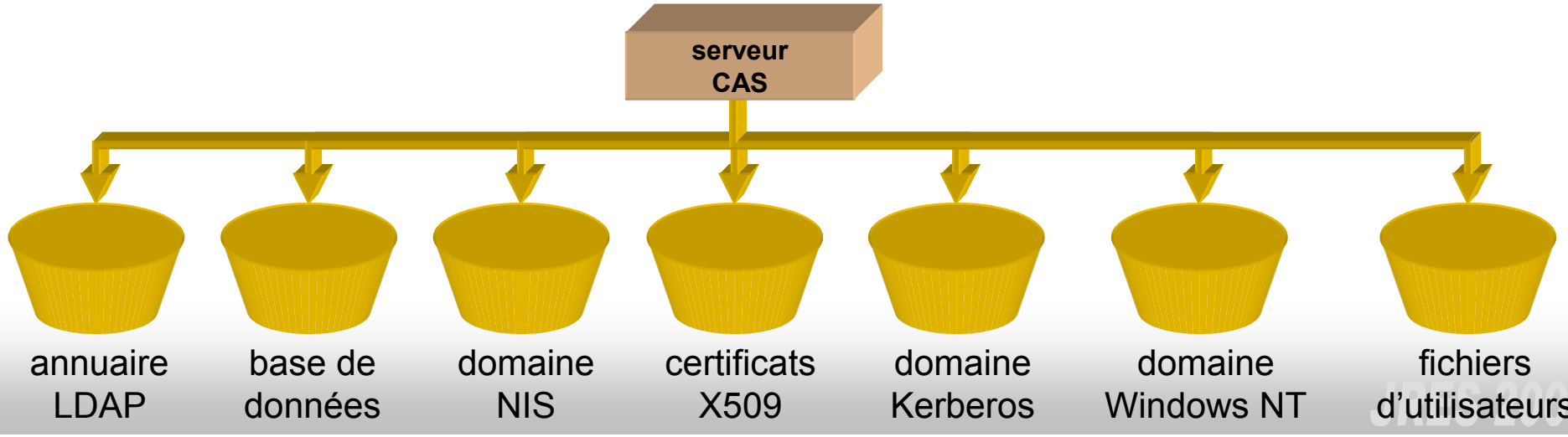


- **PT : Proxy Ticket**
  - Passeport d'un utilisateur auprès d'un service tiers
  - Ticket opaque non rejouable
  - Limité dans le temps

- **PGT : Proxy Granting Ticket**
  - Passeport d'un utilisateur pour une application auprès du serveur CAS
  - Ticket opaque rejouable

L’authentification sous CAS

- **Laissée à l’initiative de l’administrateur**
- **Développement ‘genericHandler’ par ESUP-Portail**
  - Possibilité d’utiliser plusieurs modes d’authentification
  - Configuration au format XML



CAS-ification d'une application web

- **Utilisation des librairies fournies**
- **Quelques lignes de code**
- **Cas-ification des applications proxy**
  - HTTPS nécessaire pour certaines URL
  - Complexité masquée par les librairies
- **Dans tous les cas, gérer des sessions applicatives**
- **Possibilité de mod_cas avec Apache**
  - Protection de documents statiques et/ou dynamiques

CAS-ification d'une application web

- **Exemple d'utilisation de phpCAS (ESUP-Portail)**

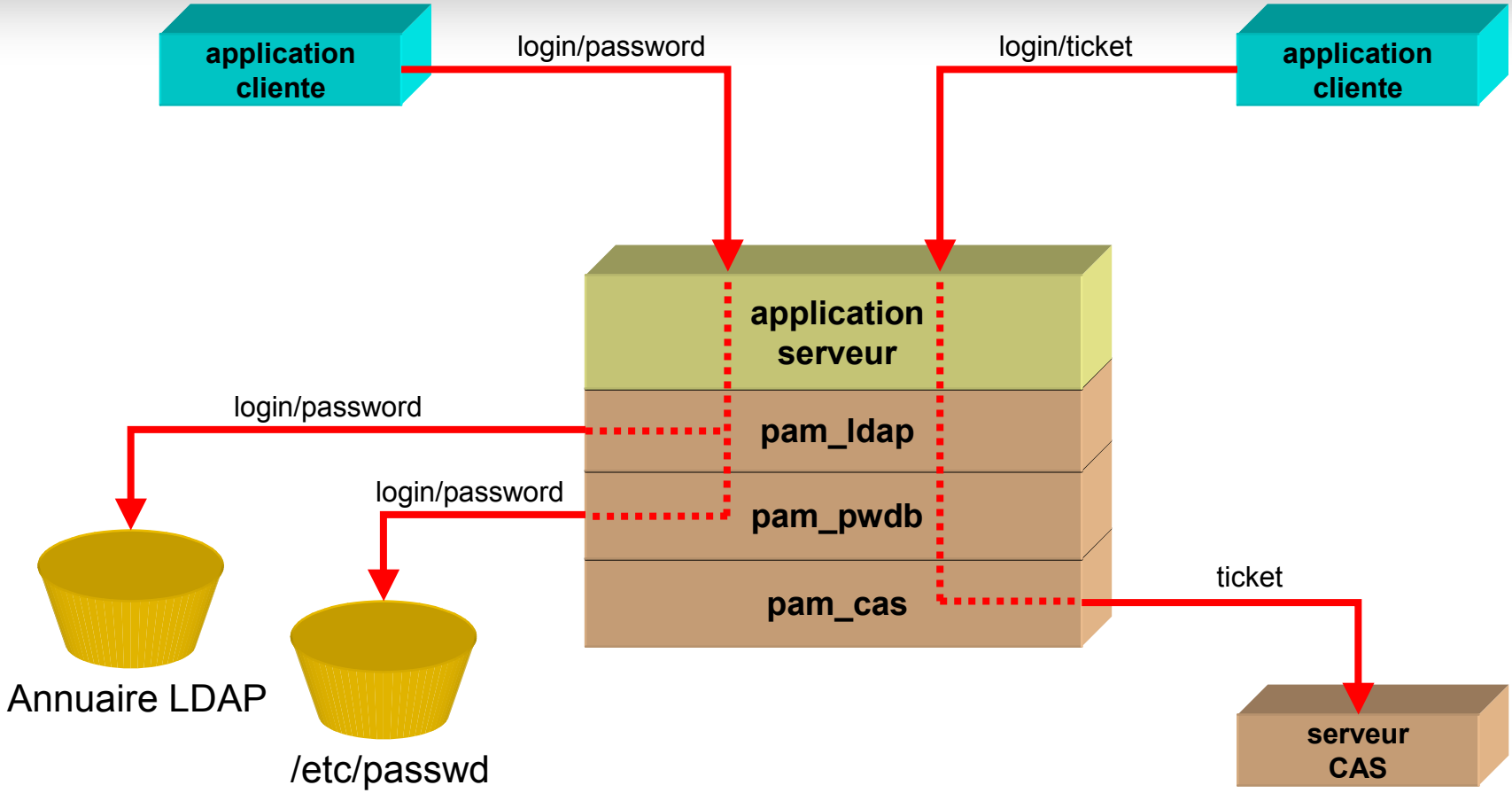
```
<?php
  // include phpCAS library
  include_once('CAS/CAS.php');

  // declare our script as a CAS client
  phpCAS::client(CAS_VERSION_2_0,'auth.univ.fr',443,'');

  // redirect to the CAS server if needed
  phpCAS::authenticateIfNeeded();

  // at this point, the user is authenticated
?>
<h1>Successfull Authentication!</h1>
<p>User's login: <?php echo phpCAS::getUser(); ?>.</p>
```

CAS-ification d'une application non WEB

- **Un des point forts de CAS**
- **Grâce au module pam_cas**
- **Exemple de configuration PAM**

```
auth sufficient /lib/security/pam_ldap
auth sufficient /lib/security/pam_pwdb.so shadow nullok
auth required /lib/security/pam_cas.so \
    -simap://mail.univ.fr \
    -phttps://ent.univ.fr/uPortal/CasProxyServlet
```

Le module pam_cas

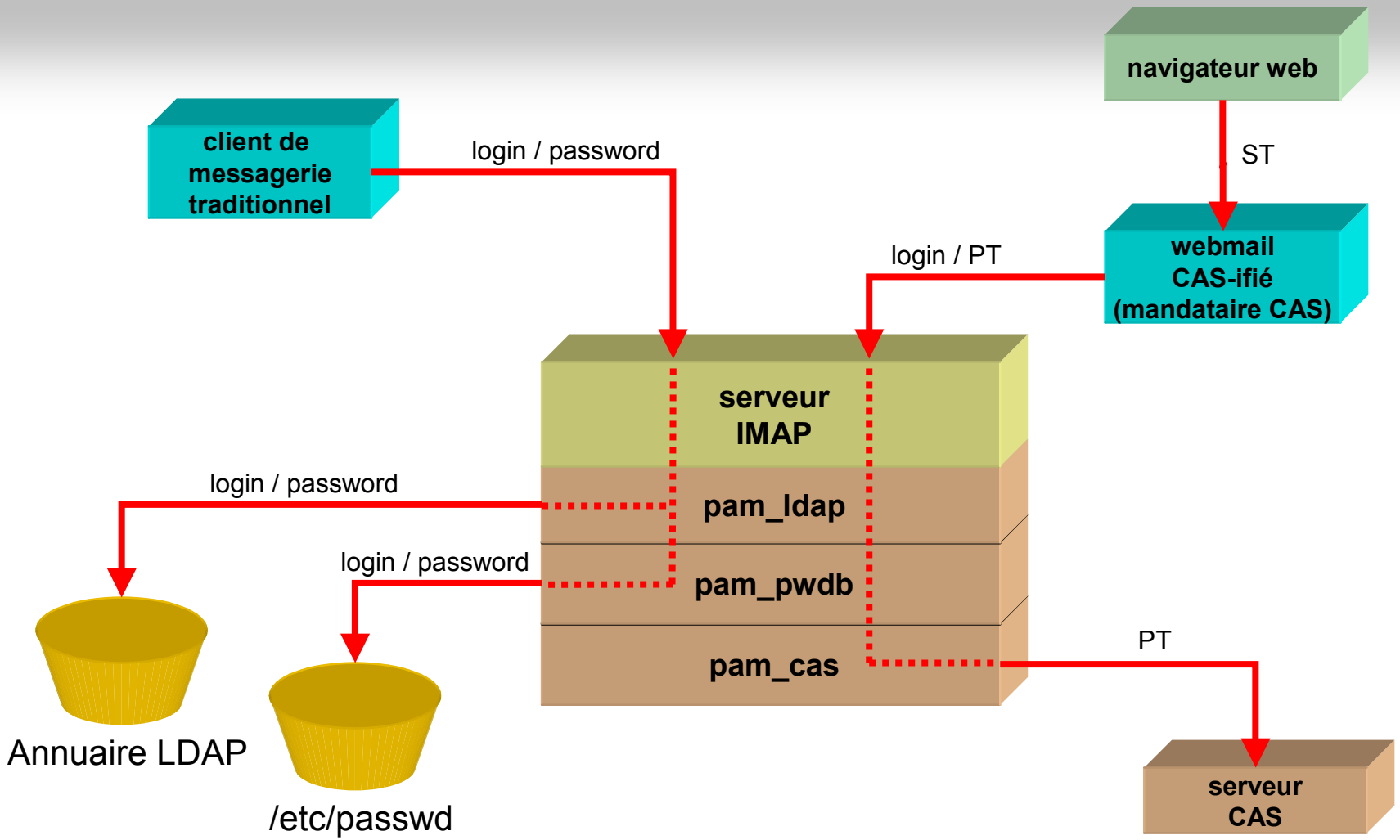


- **Pam_cas permet d'authentifier à partir d'un ticket CAS**

CAS-ification d'un serveur IMAP

- **Problématique**
  - Accéder à un serveur IMAP depuis une application web alors que l'on ne connaît pas le mot de passe de l'utilisateur connecté
  - Laisser la possibilité aux clients de messagerie traditionnels de s'authentifier « normalement » (avec un mot de passe)
  - Ne pas modifier le serveur IMAP
- **La solution : pam_cas :-)**

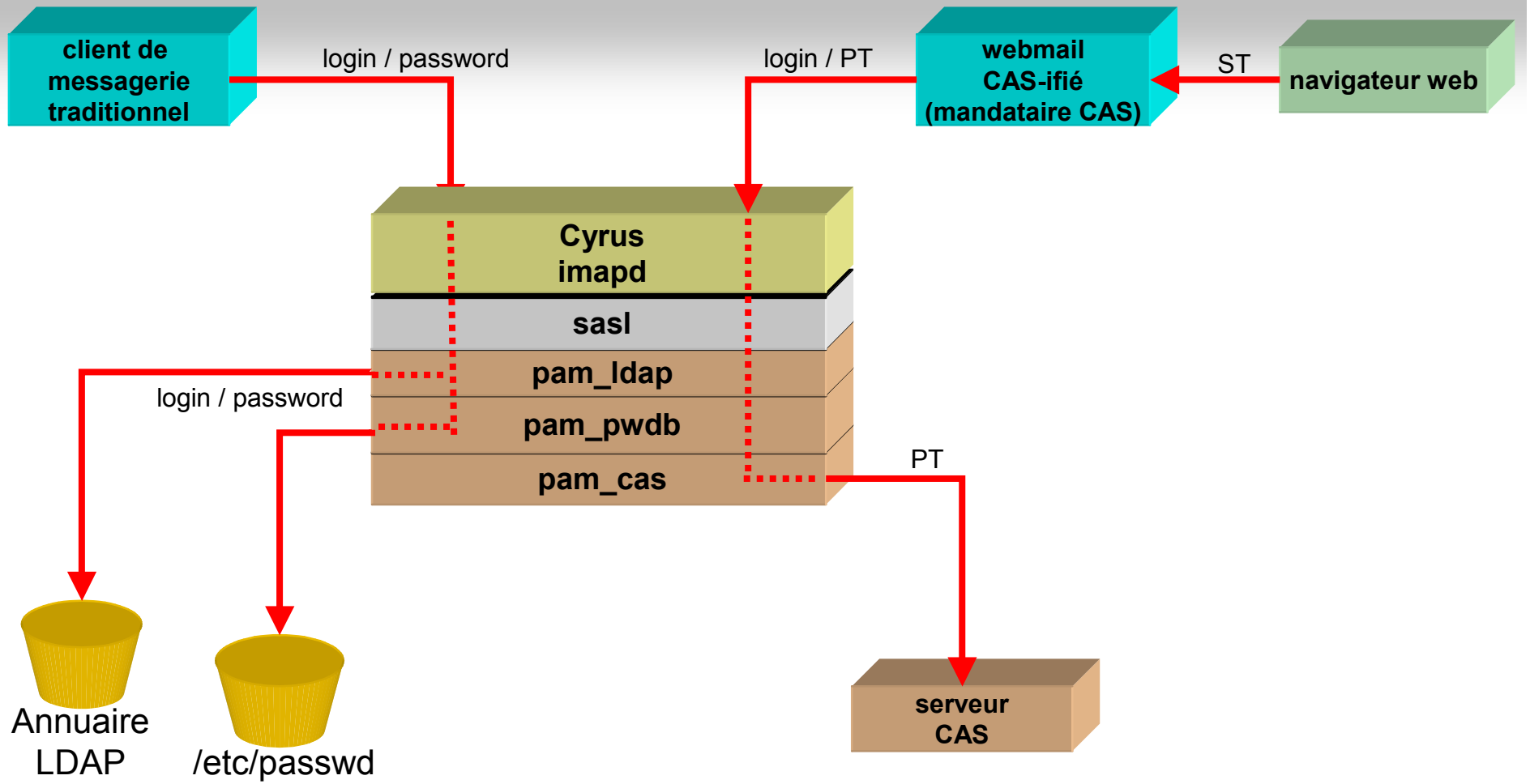
CAS-ification d'un serveur IMAP



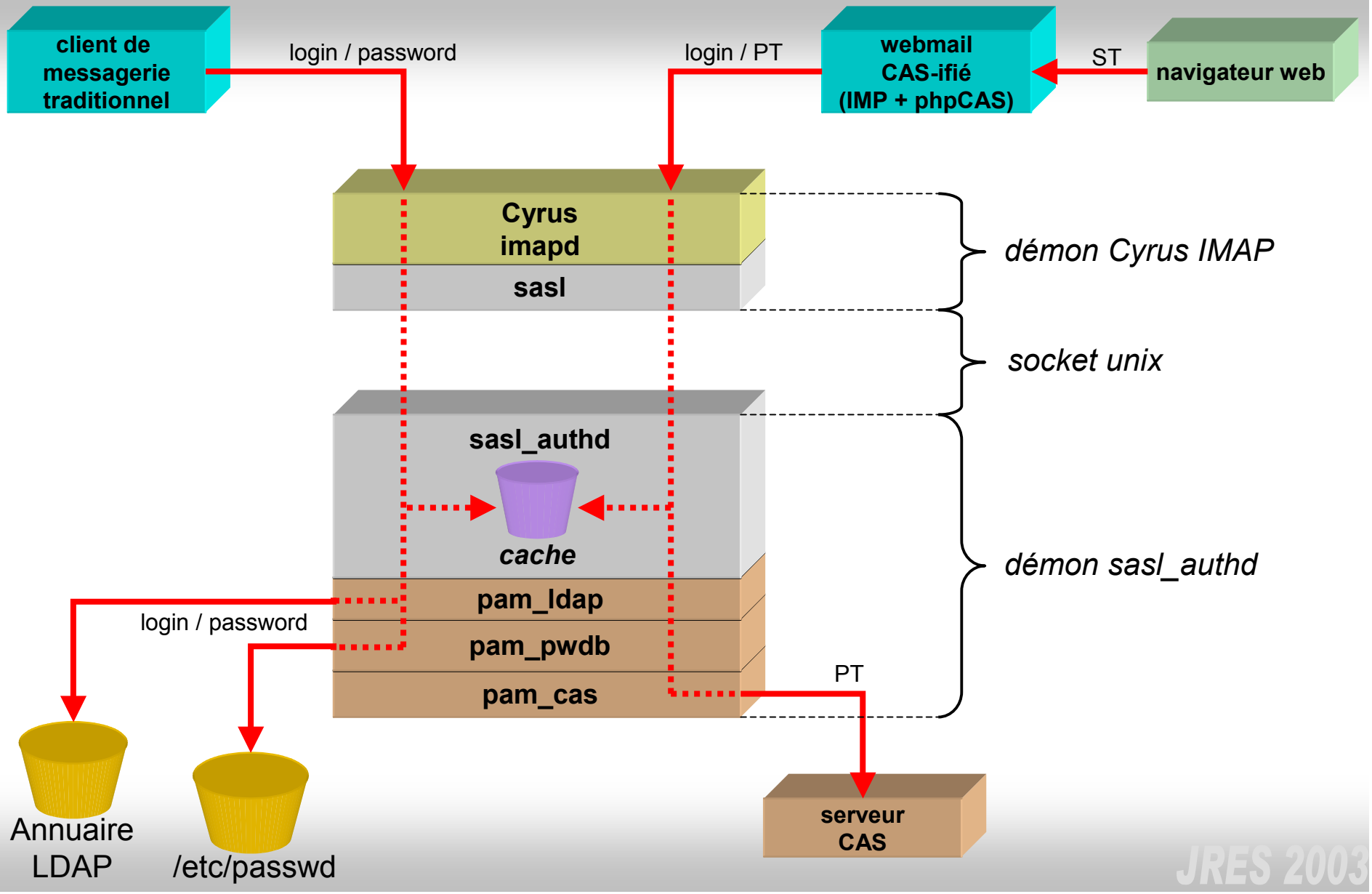
La problématique particulière d'IMAP

- **Les ouvertures de connexion vers un serveur IMAP sont très nombreuses**
  - Les clients Web ne gardent pas les connexions IMAP ouvertes (IMP)
  - Valider un ticket à chaque connexion est pénalisant pour le serveur CAS
- **Nécessité d'un cache**
- **Cyrus-IMAP propose un cache en standard (sasl_authd)**

La problématique particulière d'IMAP



La problématique particulière d'IMAP



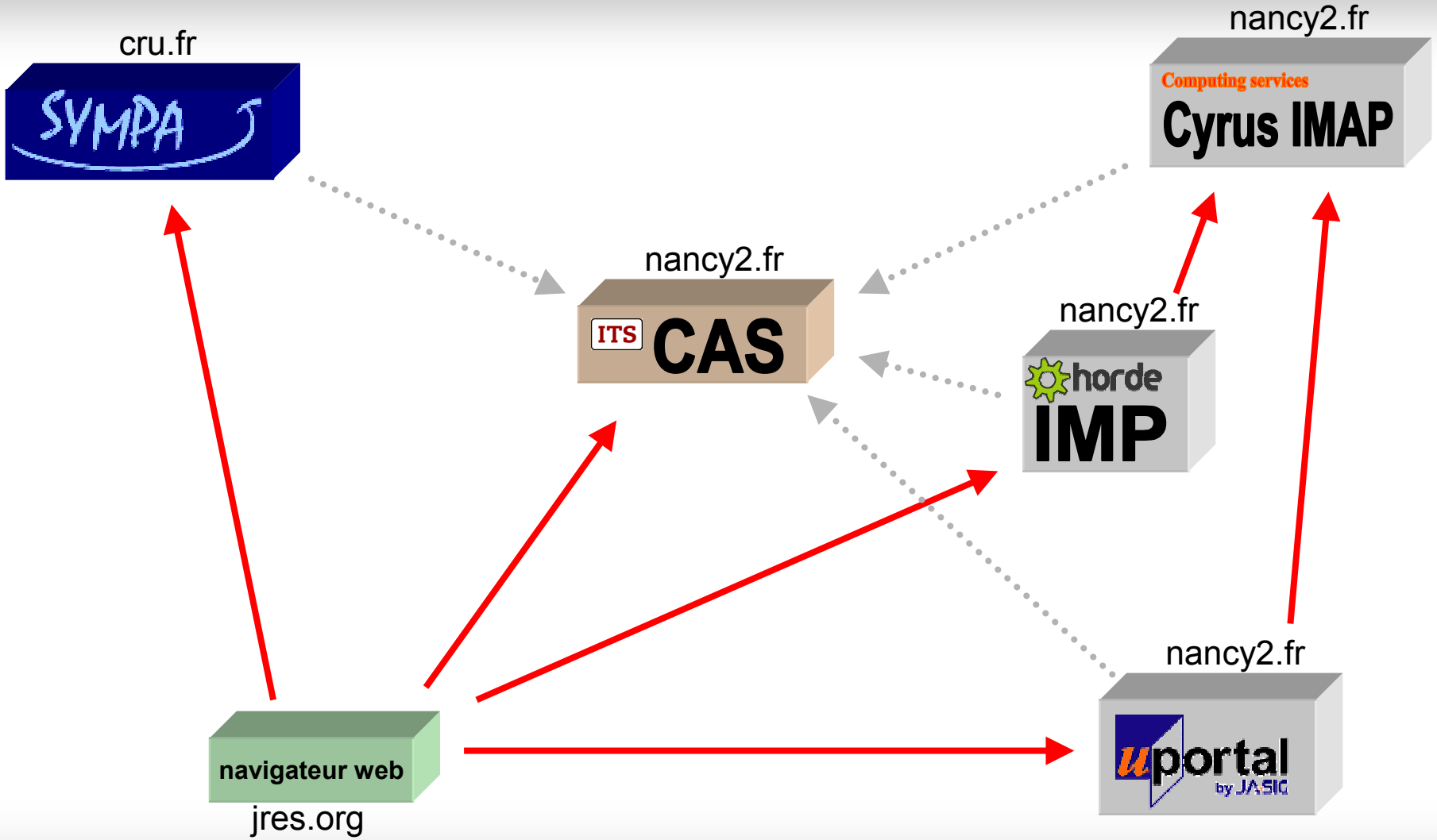
La problématique particulière d'IMAP

- **Le webmail est intégré dans le SSO de ESUP-Portail**
- **En production à l'Université de Nancy 2**
  - Efficacité du cache : 95%

CAS aujourd'hui et demain

- **CAS aujourd'hui**
  - Au sein du projet Esup-portail
  - D'une manière générale
- **Limitations et perspectives**
  - CAS traite l'authentification, pas les autorisations
  - Partage de charge et tolérances aux pannes

Démonstration rapide



Liens utiles

- **Home page de CAS :**
  http://www.yale.edu/tp/cas/
- **Archive liste CAS :**
  http://tp.its.yale.edu/pipermail/cas/
- **Archive uportal :**
  http://list.unm.edu/archives/jasig-portal.html
- **Documentations esup-portail :**
  http://www.esup-portail.org/consortium/espace/SSO_1B/
  http://www.esup-portail.org/consortium/espace/SSO_1B/cas/