CRU

# Authentification web, Single Sign-On et fédération d'identités

# L'importance de la plateforme web

- Le web propose des outils standards
- Le navigateur web est disponible sur tous les postes utilisateurs

- Les applications classiques client/serveur deviennent des applications web

- Il faut pouvoir sécuriser l'accès à ces applications

CRU

# Comment gérer le contrôle d'accès

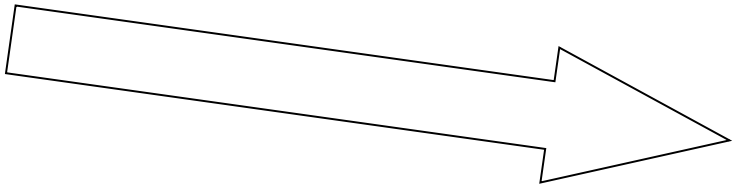
- Objectif
  - restreindre l'accès à une partie du serveur pour une population identifiée

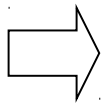
- Identification de l'utilisateur :
  1. Adresse IP du poste client (pas suffisant)
  2. Authentification de l'utilisateur

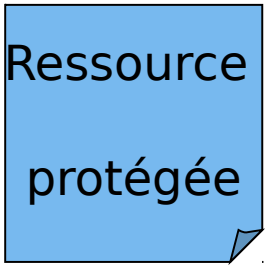
# Apache et les modules d'authentification

- On peut étendre les fonctionnalités du serveur web Apache en ajoutant des modules
- Authentification gérée par des modules :
  - mod_auth : utilisation des fichiers htpasswd et htgroup
  - mod_ssl : gestion SSL et authentification avec certificats
  - mod_authldap : authentification LDAP
  - mod_authZldap : authentification avec certificats puis autorisation basée sur LDAP
  - …beaucoup d'autres (http://modules.apache.org)

# Exemple 1
# Contrôle d'accès IP



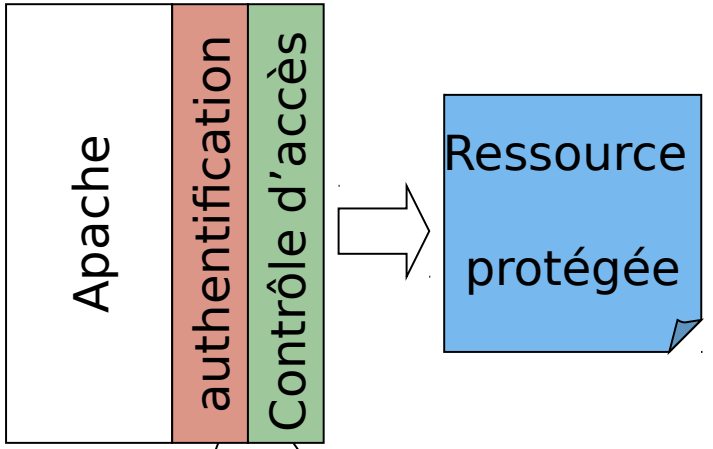





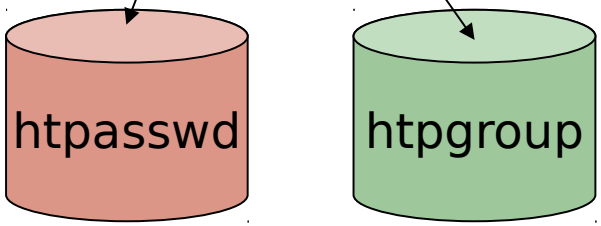

```
<File /mon_document.html>
Order deny,allow
Deny from all
Allow from .cru.fr
</File>
```

# Exemple 2
# htpasswd+htgroup

ID + mot de passe




```
<File /mon_document.html>
AuthType Basic
AuthUserFile /etc/httpd/conf/htpasswd
AuthGroupFile /etc/httpd/conf/htgroup
Require user dupont
Require group gestion
</File>
```

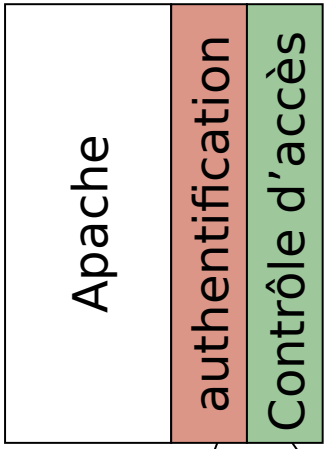


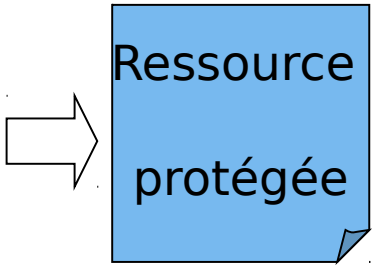
# Exemple 3
# certificats + base LDAP

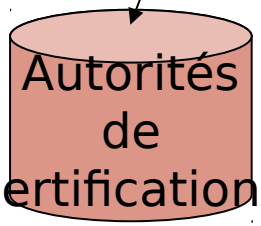
Certificat X509

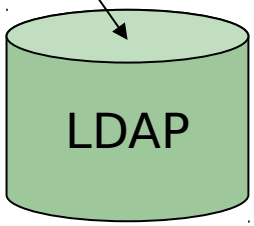
```
<File /mon_document.html>
SSLEngine on
SSLRequireSSL
…
SSLCACertificatePath /etc/httpd/ssl.crt

AuthzLDAPEngine on
AuthzLDAPServer "ldap.cru.fr"
…
Require filter (gestionMember=Yes)
</File>
```



CRU

# Authentification VS autorisation

- Deux processus disjoints
- Authentification
  - « prouver son identité »
- Autorisation (ou contrôle d'accès)
  - « déterminer les droits d'accès »

# Architecture d'un SSO les briques

- Le serveur d'authentification
  - Élément central
- Les agents d'authentification
  - Devant chaque ressource à protéger

# Le serveur d'authentification

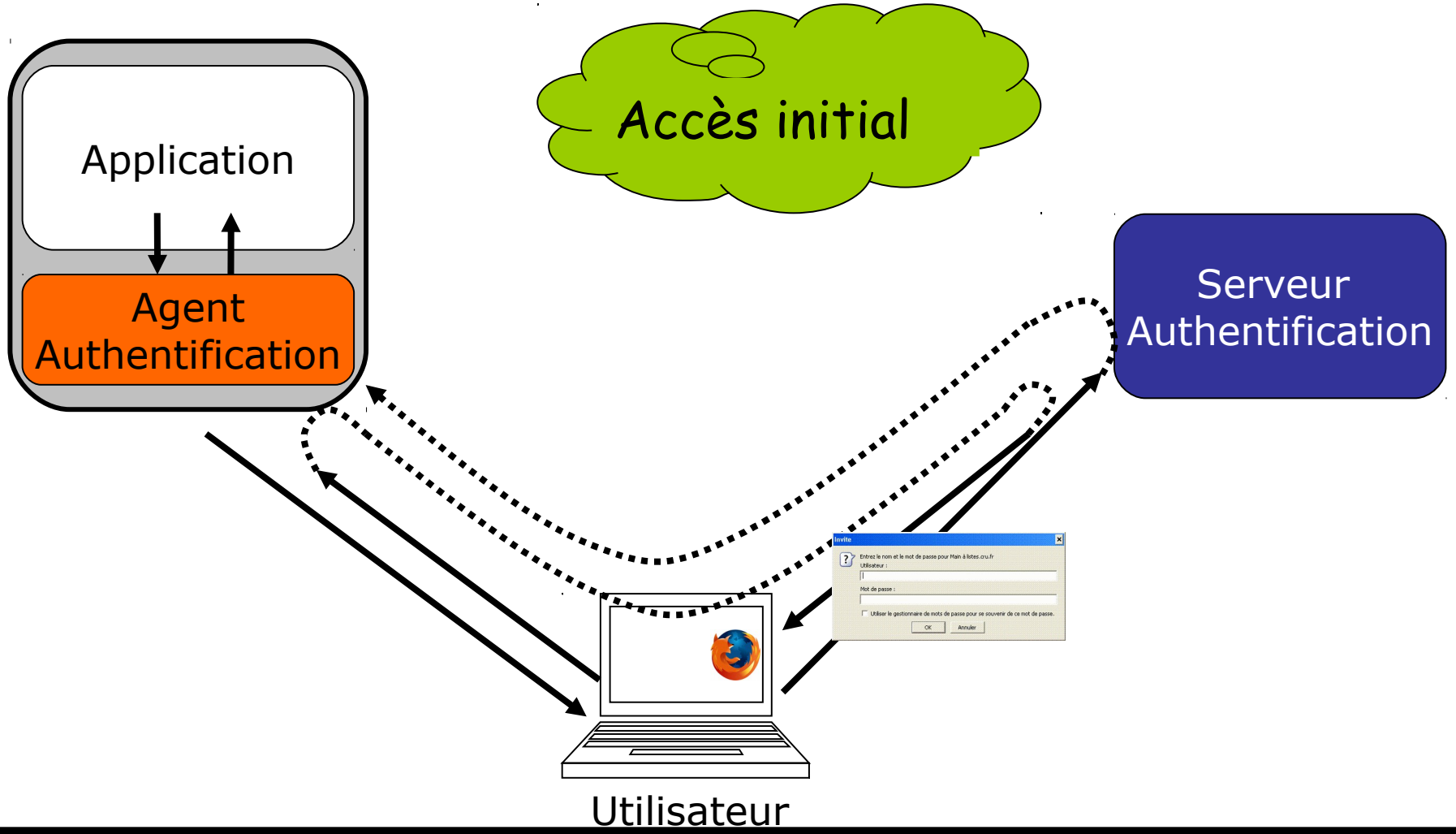
- Élément central du SSO :
  - Authentification utilisateur
  - Persistance connexion
  - Propagation identité vers applications

- Repose sur un ou des référentiels d'authentification (NIS,LDAP,…)

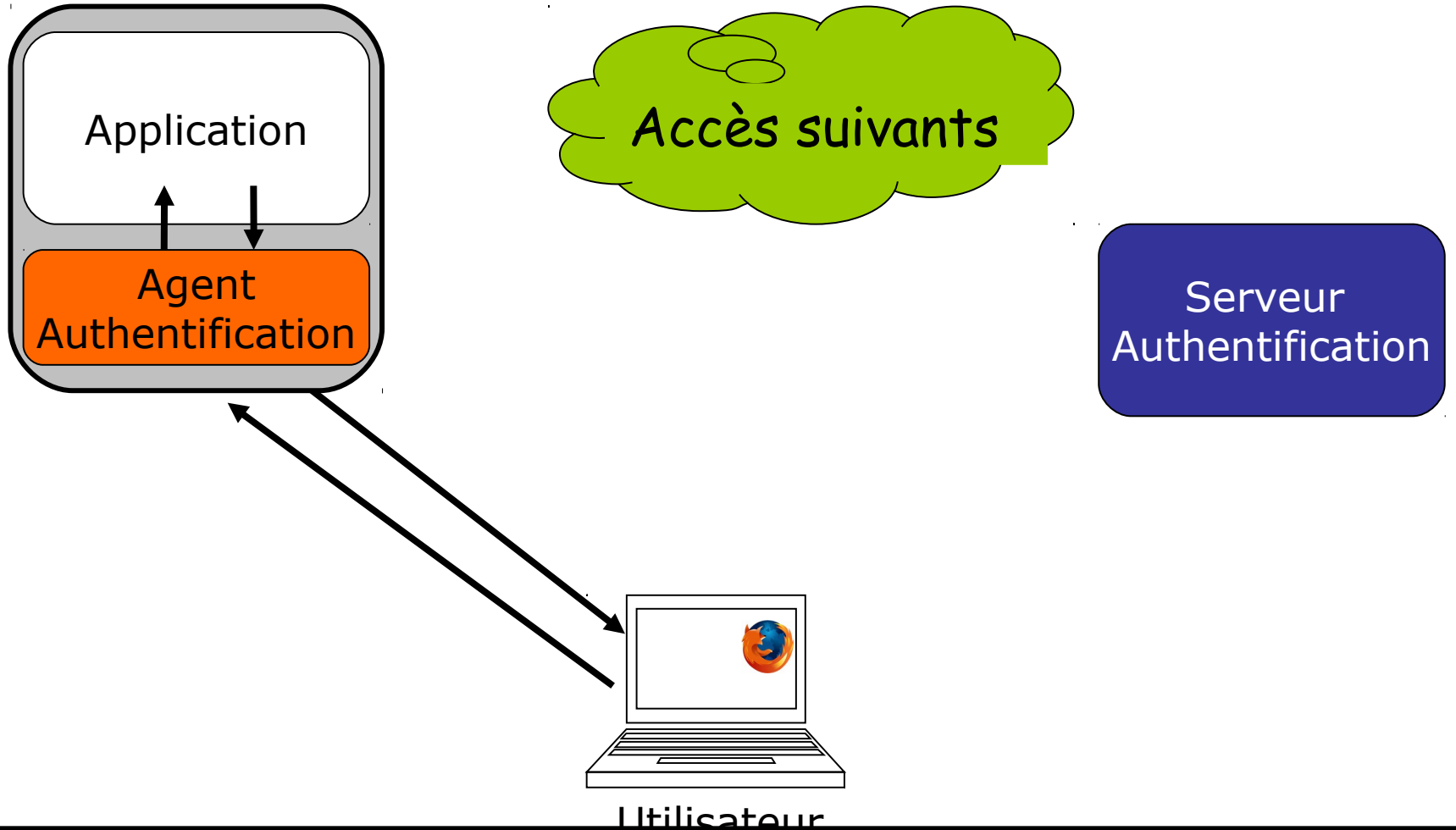
# L'agent d'authentification

- Interface entre serveur authentification et application :
  - Redirection de l'utilisateur vers le serveur d'authentification
  - Transmission de l'identité de l'utilisateur à l'application
- Implémenté sous plusieurs formes
  - Librairies clientes
  - Module intégré au serveur web

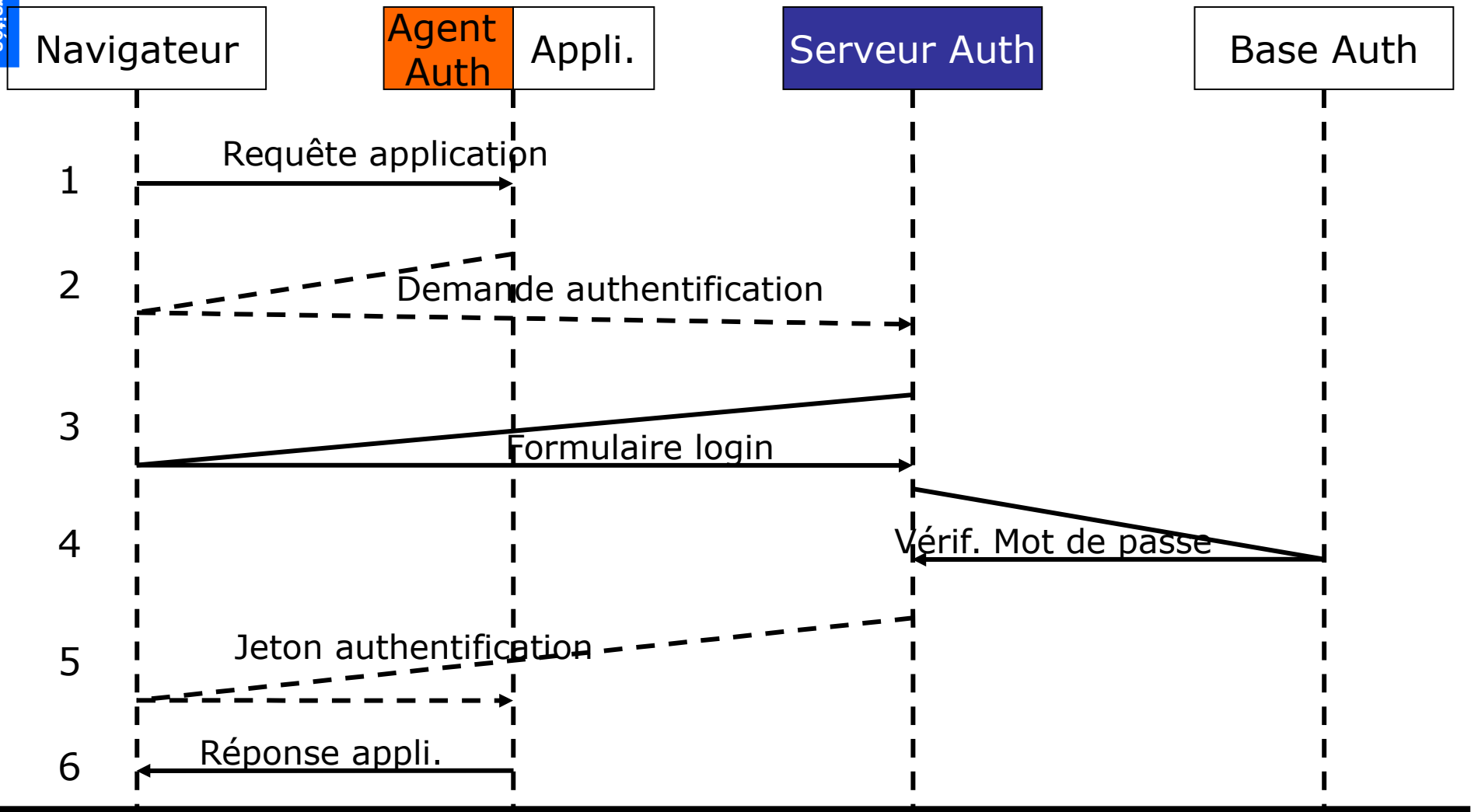
# Scénario d'authentification



# Scénario d'authentification



# Les flux


Navigateur
Agent Auth
Appli.
Serveur Auth
Base Auth
1
Requête application
2
Demande authentification
3
Formulaire login
4
Vérif. Mot de passe
5
Jeton authentification
6
Réponse appli.

CRU

# Single Sign-On
# Les techniques utilisées

- Communication via l'utilisateur utilisant :
  - Requêtes HTTP (GET) ou formulaires (POST)
  - Redirections HTTP
  - Javascript
- Persistence des sessions :
  - Cookies HTTP
- Protection des échanges :
  - SSL
  - Portée et durée de validité des cookies
  - Jetons non rejouables

# Gestion des sessions à deux niveaux

- Avec le serveur d'authentification
  - évite de se ré-authentifier à chaque application
- Avec l'agent d'authentification
  - évite de renvoyer l'utilisateur vers le serveur d'authentification à chaque requête

# Apports du SSO
# Ergonomie

- 1 seul mot de passe
- saisi 1 seule fois
- à 1 seul endroit

=> l'utilisateur est sensibilisé et ne doit fournir son mot de passe que sur la page de Login.

CRU

# Apports du SSO
# Sécurité

- Limite l'accès et la circulation des mots de passe (uniquement entre le navigateur et serveur d'authentification)
- Politique de gestion des mots de passe possible (Changement, complexité minimum, accounting)
- Extension des méthodes d'authentification envisageable

# Apports du SSO pour les applications

- Utilisation des même librairies d'authentification dans toutes les applications :
  - Meilleur intégration des applications dans l'environnement informatique
  - Rapidité de développement
  - Même niveau de sécurité partout

# Solutions de Single Sign-On l'embarras du choix…

- Pas de standardisation des techniques de SSO pour le web
- Besoins pressants dès lors que les applications web se multiplient
- Multiplication des solutions :
  - Libres ou propriétaires
  - Utilisant une terminologie propre
  - Solutions techniques très variables (plus ou moins sûres)

# CAS, un exemple de SSO

- CAS (**C**entral **A**uthentication **S**ervice)
  - SSO web développé par l'université de Yale(USA)
- Très utilisé par les universités françaises
- Une architecure simple et robuste
  - Code léger
  - L'identité de l'utilisateur est transmise via un ticket (à la Kerberos)
  - Permet un fonctionnement en mode proxy
    - Cas d'exemple : webmail, portail web.

# Le Single Logout…

- Si l'utilisateur s'est « logué » globalement, il doit pouvoir se « délogué » globalement
- Plus difficile que le SSO car chaque application gère sa propre session avec l'utilisateur.
- Il faut donc gérer une base centrale des applications qui ont une session d'authentification avec l'utilisateur et mémoriser une URL de déconnexion.
- Le single logout est complexe donc peu implémenté…

# Constat…

Les limites d'un service d'authentification / autorisation apparaissent lorsqu'on veut ouvrir les accès à des personnes extérieures…

Comment reconnaître localement l'identité et/ou les privilèges d'un utilisateur enregistré ailleurs ?

# Des cas d'utilisation

- Contrôle d'accès en fonction du profil de l'utilisateur
  - Exemple : étudiant en pharmacie
- Intranet pour un groupe de travail
  - Chercheur, étudiants, association, informaticiens
- Accès aux périodiques électroniques depuis un ENT
  - Science Direct, JSTOR, …

CRU

# Les solutions techniques de fédération d'identités

# SAML

- SAML=Security Assertion Markup Language
- Standard OASIS en 2002
- Répond à un besoin d'interopérabilité
- Echanges d'assertions de sécurité entre services
- Indépendant des mécanismes d'authentification

# Types d'assertions SAML

- Authentification
- Échange d'attributs
- Décisions d'autorisation

CRU

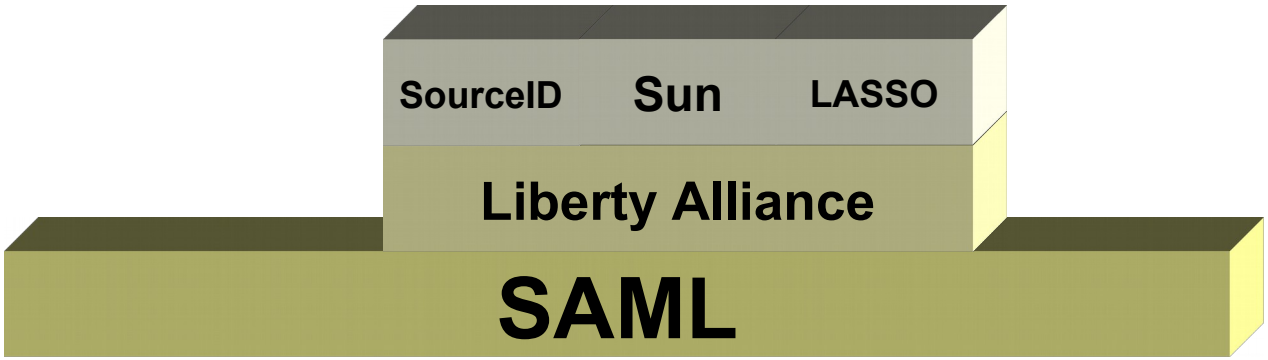
# Exemple d'assertion SAML

```
<saml:Assertion MajorVersion="1" MinorVersion="0"
    AssertionID="128.9.167.32.12345678"
    Issuer="Comite Reseau des Universites"
    IssueInstant="2002-03-21T10:02:00Z">
    <saml:Conditions NotBefore="2002-03-21T10:02:00Z"
        NotAfter="2002-03-21T10:07:00Z" />
    <saml:AuthenticationStatement
        AuthenticationMethod="password"
        AuthenticationInstant="2002-03-21T10:02:00Z">
    <saml:Subject>
        <saml:NameIdentifier
            SecurityDomain="www.cru.fr"
            Name="osalaun" />
    </saml:Subject>
</saml:AuthenticationStatement>
```

# Liberty Alliance

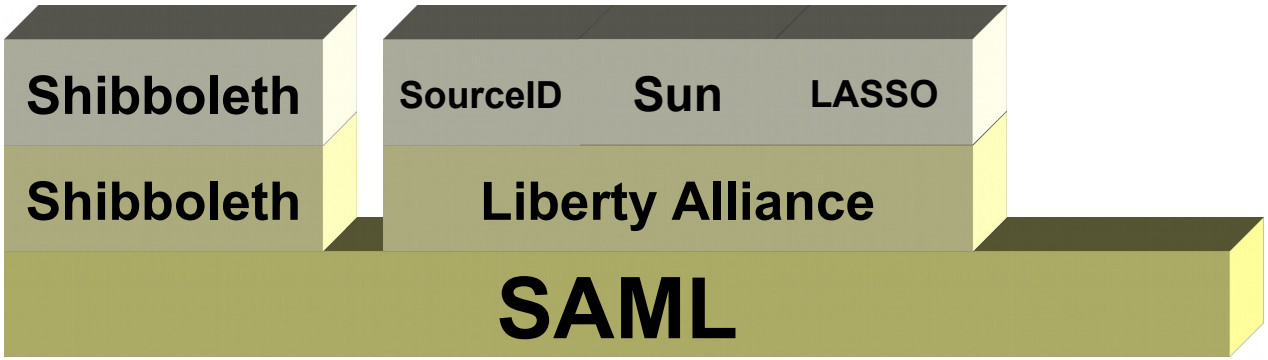
- Liberty Alliance n'est pas un produit

- Consortium d'industriels produisant des spécifications sur la gestion d'identités
- S'appuie sur SAML
- Implémenté dans de nombreux produits



# Shibboleth

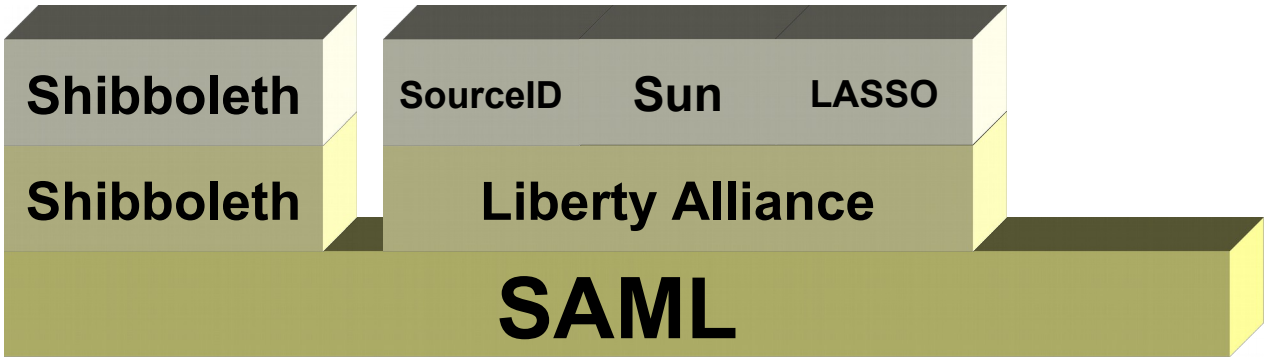
- Norme et produit développé par Internet2
- Open source
- Première version en 2002
- Basé sur SAML (bibliothèque OpenSAML)
- Utilisé surtout par des universités
  - en Allemagne, Danemark, Finlande, France, Grande-Bretagne, Suède, Suisse, USA, Australie, Chine



CRU

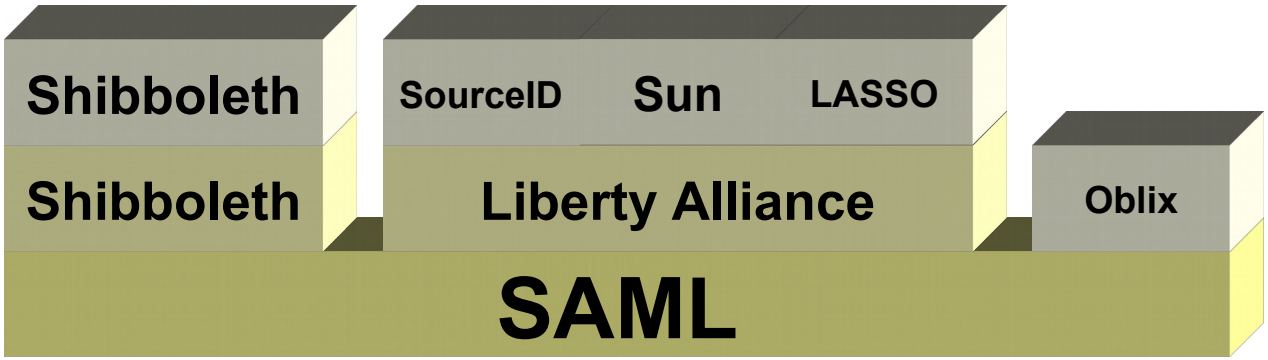
Shibboleth.

- Conçu pour interconnecter les SSO des universités
- Fonctionnalités
  - Délégation d'authentification
  - WAYF pour orienter l'utilisateur
  - Propagation des attributs utilisateur
  - Partage de méta données
  - Définition de règles de confiance



# WS-Federation

- *Draft* porté par Microsoft et IBM, 2003
- Basée sur les spécifications WS-*
  - WS-Security, WS-Trust, WS-Policy, WS-MetadataExchange
- Définit l'échange d'identités et d'attributs entre domaines de sécurité



# Autres initiatives

- CardSpace (Microsoft)
- OpenID
- OAuth
- Bandit (IBM et Novell)

# OpenID
# http://openid.net/

- Protocole de délégation d'authentification
- Origine : authentification sur les blogs
- Principe :
  1. Saisie de l'URI de son OpenID provider
  2. Redirection vers l'OpenID provider
  3. Authentification

– Développement rapide de la technologie
  – Nombreux services compatibles
  – Nombreuses implémentations

# Oauth
# http://oauth.net

- Standard récent (version 1.0, décembre 2007)
- Définit un standard permettant la délégation d'un utilisateur pour l'accès à un service
- Exemple d'utilisation
  - Un utilisateur autorise un service d'impression à accéder à ses photos sur un site de gestion de photos
- Oauth est complémentaire de OpenID

# Apports de la fédération d'identités

- Pour l'utilisateur
  - Il n'utilise que le mot de passe de son SSO
  - Adapté aux utilisateurs nomades
- Pour l'établissement de rattachement
  - Niveau de sécurité constant (SSO)
  - Meilleure maîtrise des données personnelles
- Pour les gestionnaires de ressources
  - Plus besoin de gérer des comptes utilisateurs
  - Accès à des données utilisateurs fiables