Applications Web

Authentification

Jean-René Rouet

plan

- A quoi sert l'authentification dans les serveurs WEB
- Authentifications simples
- WSSO
  - modèle
  - implémentation
  - fonctionnement
  - solutions
- Et après…

à quoi ça sert ?

- Contrôler l’accès à l’information
- Attacher des rôles applicatifs à un utilisateur
  - administrateur
  - visiteur

authentification simple

- le couple utilisateur/mot de passe est fourni par le navigateur web
- fonctionnalité HTTP
- ce couple est présenté à chaque requête
- la vérification est faite par apache ou un module ou l'application web elle-même

authentification simple

- apache
  - fichier user/mot de passe crypté (crypt)
- modules
  - mod_auth_kerb
  - mod_auth_ldap
  - modXLDAPAuth

authentification simple

- mod_auth_kerb

```
fichier .htaccess :

AuthType Kerberos
AuthName "SSO test for kerberos password"
KrbMethodK4Passwd on
KrbVerifyKDC on
Require valid-user
```

authentification simple

- mod_auth_ldap

```
fichier .htaccess :

AuthName "CC Staff only"
AuthType Basic
LDAP_Server ldap.in2p3.fr
LDAP_Port 389
Base_DN "ou=cc,o=in2p3,c=fr"
UID_Attr_Alt uid
require valid-user
```

authentification simple

- modXLDAPAuth

```
fichier .htaccess :

XLDAPAuthoritative          on
XLDAPAuthServer             ldap.in2P3.fr
XLDAPAuthSuffix             "ou=cc,o=in2p3,c=fr”
XLDAPAuthFilter             "(&(mail=%
{SSL_CLIENT_S_DN_Email})(CN=%{SSL_CLIENT_S_DN_CN}))"
XLDAPAuthScope              Sub
XLDAPAuthRemoteUserAttr cn
```

authentification simple

- HTTPS
  - le navigateur web présente un certificat client
  - la vérifications est faite par apache et/ou par l’application web

authentification simple

HTTPS

```
fichier .conf :

<Directory /usr/local/apache2/htdocs/secure/area>
  SSLVerifyClient       require
  SSLVerifyDepth        5
  SSLCACertificateFile conf/ssl.crt/ca.crt
  SSLCACertificatePath conf/ssl.crt
  SSLRequireSSL
  SSLRequire            %{SSL_CLIENT_S_DN_OU} eq "cc"
  SSLOptions            +
</Directory>
```

authentification simple

- couple formulaire de connexion/session
- génère un cookie de session
  - (ou est passé dans l’url à chaque fois)
- l’application web vérifie les valeurs saisies

ça marche ?

- oui, ça marche
- surtout si c’est couplé à HTTPS
- …
- WSSO
- …
- qu’est-ce ça apporte, alors ?

limites

- chaque application demande une authentification supplémentaire (généralement différente)
- euh, on pourrait pas en avoir qu'une

limites

- chaque application vérifie des authentifications
  - possibilités de bogues découvrant des failles
  - si c'était centralisée et unique, on pourrait se focaliser sur la sécurité
  - même être un peu paranoïaque
  - l'application web ne se préoccupe plus de ce problème

limites

- C’est un peu le chaos
- plusieurs mots de passe, …
- rationaliser les applications et la gestion des comptes

WSSO

- On doit pouvoir définir un modèle décrivant ce type d'application
- http://middleware.internet2.edu/webiso/

un modèle

- le modèle doit fonctionner
- avec tous les navigateurs Web du marché
- avec les systèmes d’authentification déjà intégrés

un modèle

- il ne donne pas le mot de passe à l'application web
- juste un ticket de session
- et l'identifiant de l'utilisateur

un modèle

- il fournit une authentification unique
- il est facile à ajouter aux applications existantes

les composants du modèle

- le service d'authentification par page web
  - propose un page d'identification
  - traite la réponse

les composants du modèle

- le service d'authentification par page web
  - vérifie les informations vis à vis d'un système déjà intégré
    - kerberos
    - ldap
    - certificat X509
- le destinataire du résultat de l'authentification est l'agent d'authentification

les composants du modèle

- l’agent d’authentification
  - fournit l’intégration du système dans l’applications existante
  - il génère les requêtes au service d’authentification
  - traite le résultat
  - fournit l’identifiant de l’utilisateur à l’application web

les composants du modèle

- l’agent d’authentification
  - il peut être implémenté sous forme
    - d’un librairie incluse à l’application web
    - d’une extension au serveur web (module apache)
    - ou autre…

les composants du modèle

- l'application web
  - c'est le client de l'infrastructure WSSO, qui doit connaître ses besoins en terme d'identification d'utilisateur
    - cela va de l'accès à de la page web statique
    - à l'application qui gère des privilèges d'accès sur des fonctions et des données

les composants du modèle

- le service de vérification
  - c'est un service généralement externe au WSSO qui valide l'information fournie au service d'authentification
  - généralement, un couple utilisateur/mot de passe
  - le but est d'utiliser un service existant
    - ldap
    - kerberos

les composants du modèle

- le navigateur web
  - WSSO nécessite des fonctionnalités standards
    - redirection d'url
    - SSL/TLS
    - formulaire
    - cookie
    - (javascript/ECMAscript)

les composants du modèle

- l'annuaire de l'organisation
  - informations professionnelles sur les utilisateurs
    - adresse électronique
    - groupes d'appartenance
    - droits
- informations sur l'organisation
- informations sur les applications

les fonctionnalités du modèle

- Authentification unique
  - on s’identifie une fois par session
  - pour toutes les applications

les fonctionnalités du modèle

- possibilité de terminer sa session (logout)

les fonctionnalités du modèle

- chaque application a sa politique d’authentification
- et peut nécessiter un réauthentification

les fonctionnalités du modèle

- peut-être utiliser sur une borne kiosque
- dans ce cas pourquoi pas réduire la durée de session par exemple
- désactiver l'accès aux fonctionnalités d'administration

les fonctionnalités du modèle

- préférences utilisateur pour une session
  - je suis sur une borne internet (par exemple)

les fonctionnalités du modèle

- on peut ignorer ou abandonner l’authentification en cours

les fonctionnalités du modèle

- utilisation non interactive du service d’authentification
  - dans ce cas, on vérifie juste l’existence d’une session active et valide

les fonctionnalités du modèle

- personnalisation de la page d’authentification possible en fonction de l’application cliente

implémentations

- le service d'authentification est une application web standard correctement sécurisée
  - elle utilise HTTPS
  - elle utilise les technologies de session et de cookie

implémentations

- l'agent d'authentification est :
  - une librairie incluse dans l'application web cliente
    - dans ce cas elle est écrite dans le même langage
    - il est indépendant du serveur web

implémentations

- l'agent d'authentification est :
  - un module du serveur web
    - dans ce cas, il est écrit pour le serveur web
    - nécessite la configuration du serveur web pour prendre en compte WSSO pour l'application cliente

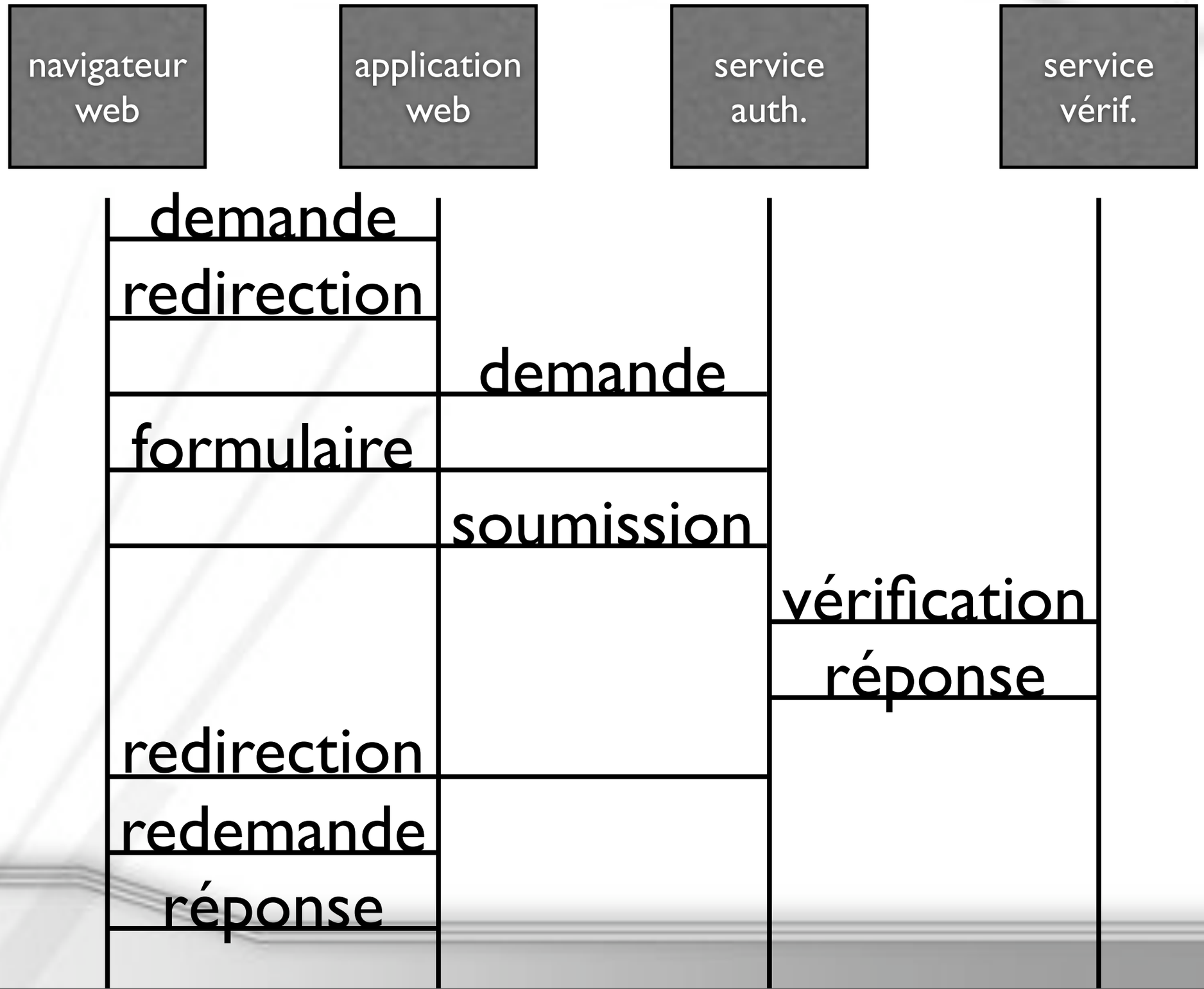
implémentations

- l'agent d'authentification est :
  - un wrapper
    - dans ce cas, il est écrit dans n'importe quel langage
  - nécessite la configuration du serveur web pour prendre en compte WSSO pour l'application cliente
  - attention ceci rajoute de la charge cpu
    - le wrapper va traiter tous les requêtes vers l'applications web

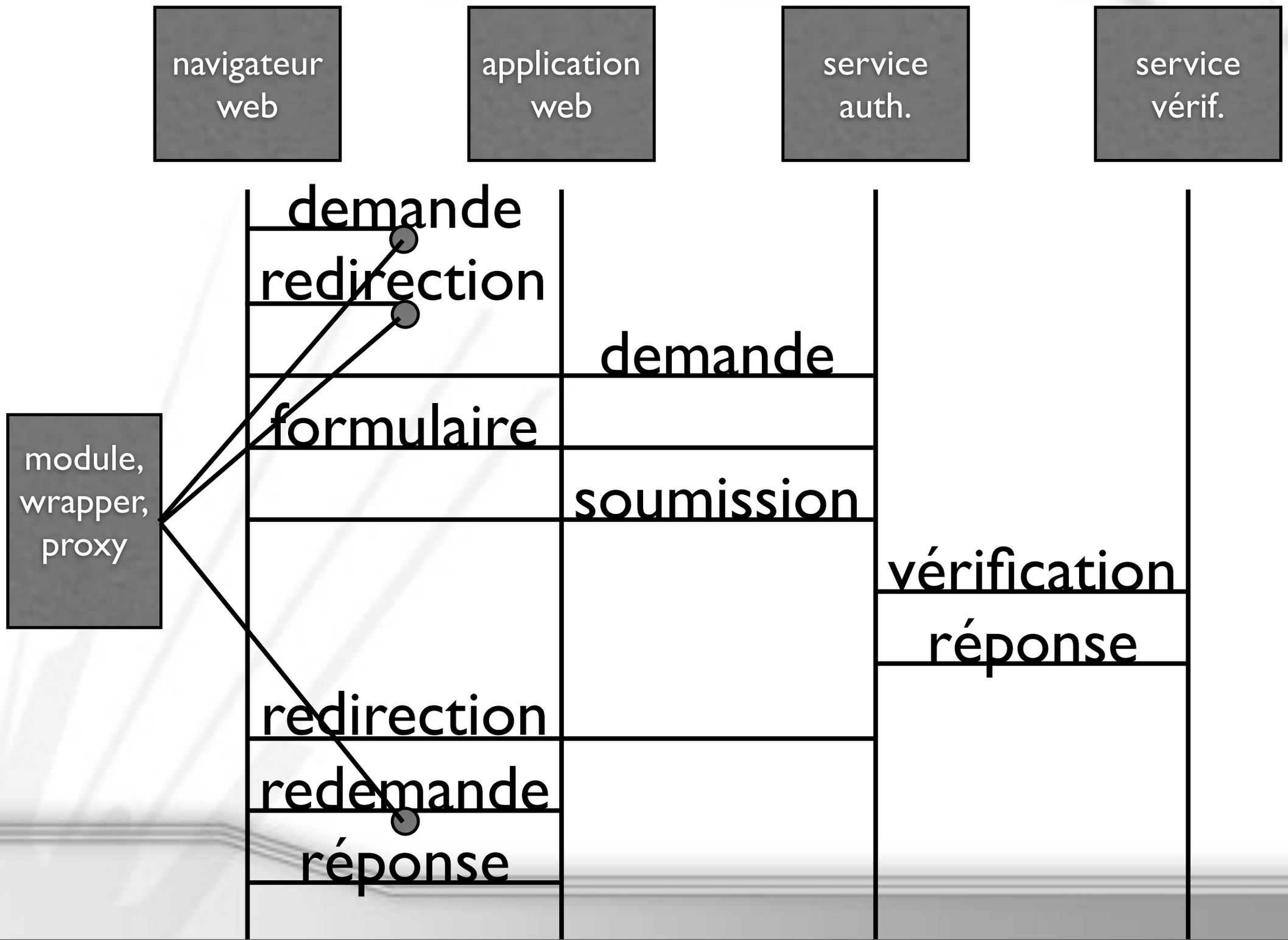
implémentations

- l’agent d’authentification est :
  - un proxy logiciel ou matériel

fonctionnement



fonctionnement



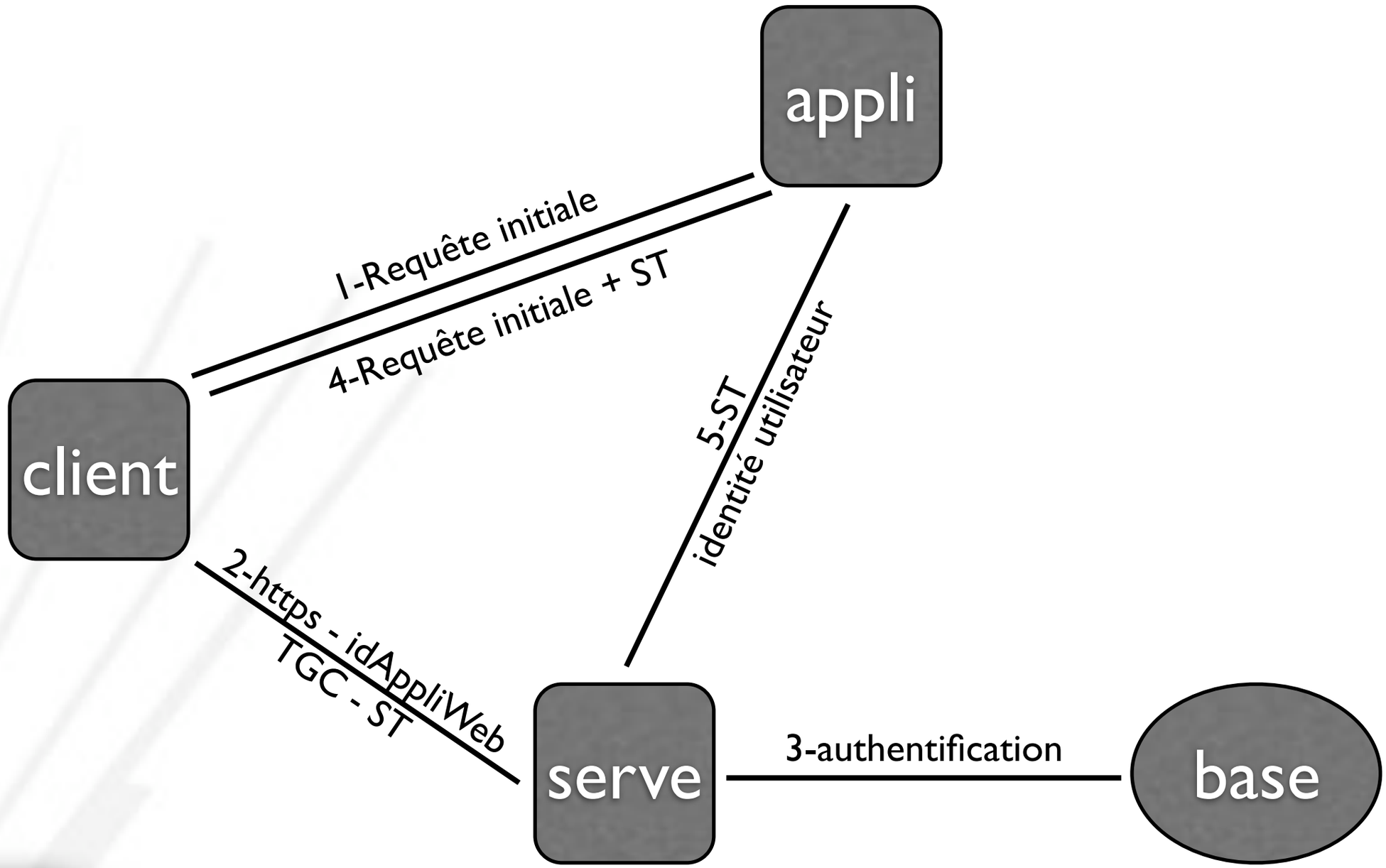
solutions

- des dizaines de solution existent
  - gratuites ou payantes
- plusieurs ajoutent la fonctionnalité d’authentification inter royaume

solutions

- CAS (Yale university)
- wsso
- complet (tous les clients sont disponibles)
- authentification proxy

solutions


appli
1-Requête initiale
4-Requête initiale + ST
client
5-ST
identité utilisateur
2-https - idAppliWeb
TGC - ST
serve
3-authentification
base

solutions

- pubcookie
  - wsso
  - fonctionnalités complètes (module apache)

solutions

- moria (FEIDE)
- Brown's web authentication system
- Stanford WebAuth
- A-Select
- PAPI

solutions

- moria (FEIDE)
- Brown's web authentication system
- Stanford WebAuth
- A-Select
- PAPI

A2C2

- Urec

Et après…

- Gestion de l’identité numérique intersite
- microsoft passport
- shibboleth
- liberty alliance

Références

- http://middleware.internet2.edu/webiso/
- http://www.pubcookie.org/
- https://clearinghouse.ja-sig.org/wiki/display/CAS/Home
- http://www.esup-portail.org/consortium/espace/SSO_1B/index.html
- http://www.cru.fr/sso/
- https://cctools.in2p3.fr/sso/