Introduction

aux architectures web

de Single Sign-On

Single Sign-on…

- Authentifier 1 seule fois un utilisateur pour accéder à un ensemble d'applications

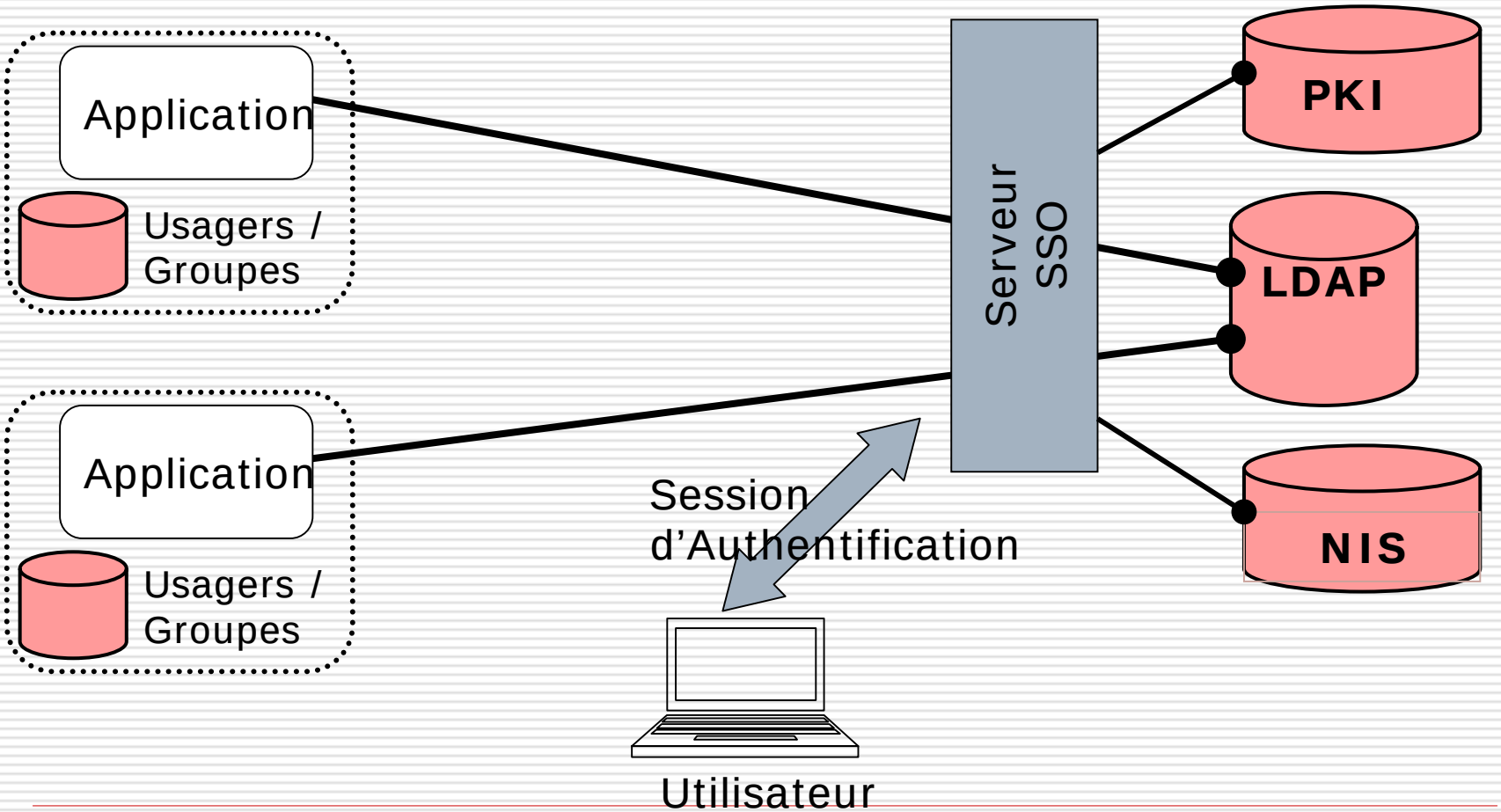
...contexte web

- Nombre croissant d'applications ayant chacune son système d'authentification
- Au mieux les applications utilisent un « backend » d'authentification commun (NIS, LDAP,...)
- => Besoin de définir un service d'authentification utilisé par toutes les applications web

Plan de la présentation

- Apports d'un SSO web
- Architecture classique
- Problématique multi tiers
- Déploiement d'un SSO
- Vers un SSO inter-établissement
- Quelques produits universitaires
- Etudes et normalisation

Apports d'un SSO
Architecture simplifiée



Apports d'un SSO

Pour l'utilisateur

- Session d'authentification partagée par toutes les applications web
- Apport ergonomique
  - 1 seul mot de passe
  - Saisi 1 seule fois
  - À 1 seul endroit
- Contexte portails applicatifs (mosaïque d'applications devenant un service)
  - Localisation unique
  - « Look » commun
  - **sso**

Apports d'un SSO

Pour l'administrateur

- Sécurité
  - Limiter l'accès et la circulation des mots de passe
  - Politique de gestion des mots de passe possible (rappel, changement, accounting)
  - Extension des méthodes d'authentification
- Gestion des comptes
  - Plus de gestion multiple des comptes
  - Limite les risques d'inconsistances
- Les applications
  - Meilleur intégration avec le SI
  - Rapidité de développement

Les concepts de Kerberos

- Applications déchargées de l'authent.
- Applications ne recueillent pas les éléments d'authentification
- Le serveur d'authent. délivre des tickets aux clients et aux applications
- Relation de confiance entre le serveur d'authent. et les applications

Architecture classique

Les briques

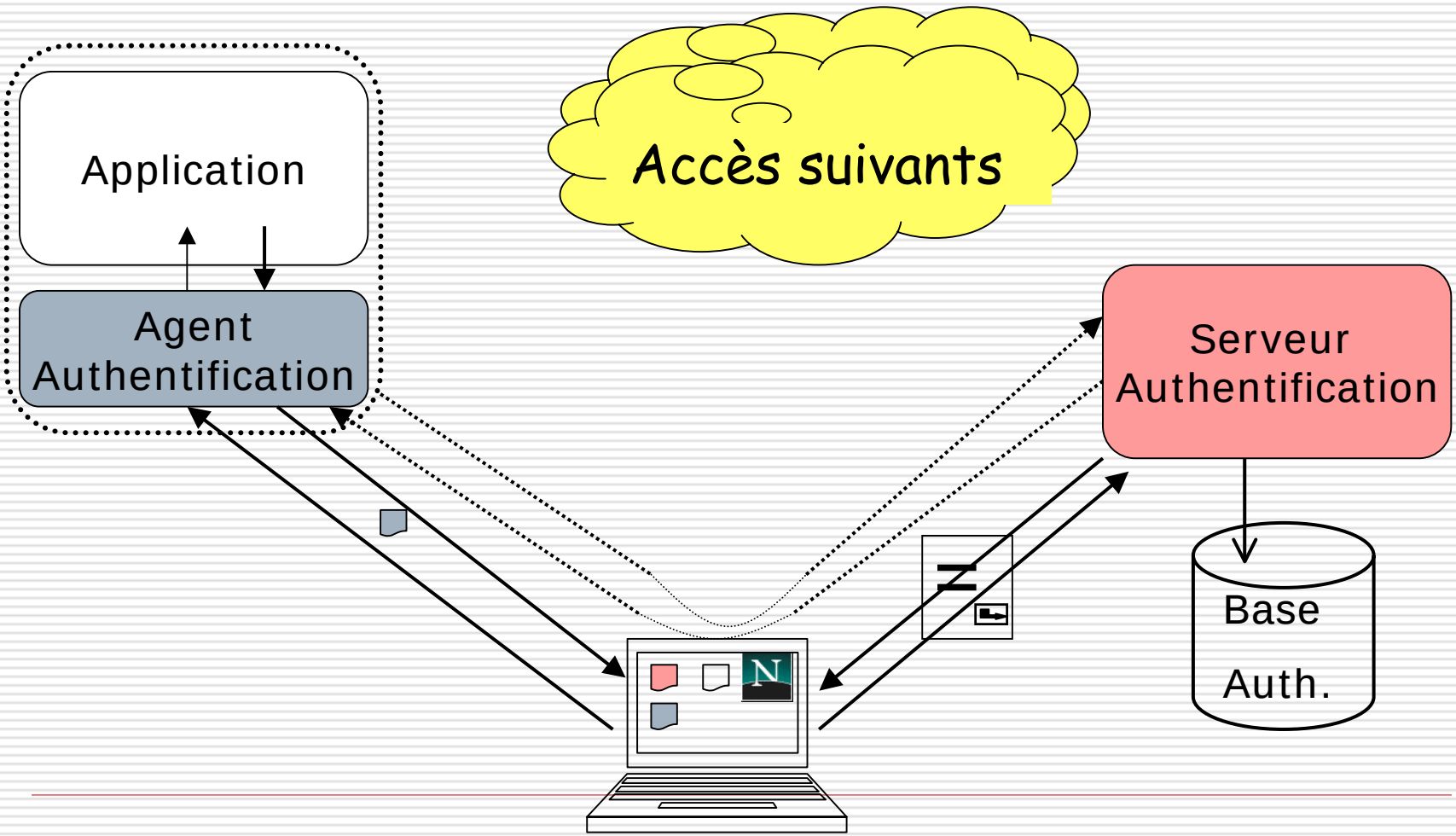
- Le serveur d'authentification :
  - Élément central du SSO :
    - Authentification utilisateur + maintient session
    - Propagation identité vers applications
  - Extension possible des méthodes d'authentification et du « backend »
- L'agent d'authentification :
  - Interface serveur authentification ⇔ applications
    - Redirection de l'utilisateur vers le serveur d'authentification
    - Transmission de l'identité de l'utilisateur à l'application
  - Transmission de l'identité directement (identifiant) ou indirectement (jetons)

Les techniques utilisées

- Client = navigateur web
- Communication directes ou indirectes via l'utilisateur (navigateur) utilisant :
  - Requêtes HTTP (GET) ou formulaires (POST)
  - Redirections HTTP
  - Javascript
- Persistance des sessions :
  - Cookies HTTP
- Protection des échanges :
  - SSL
  - Portée et durée de validité des cookies
  - Cookies non rejouables

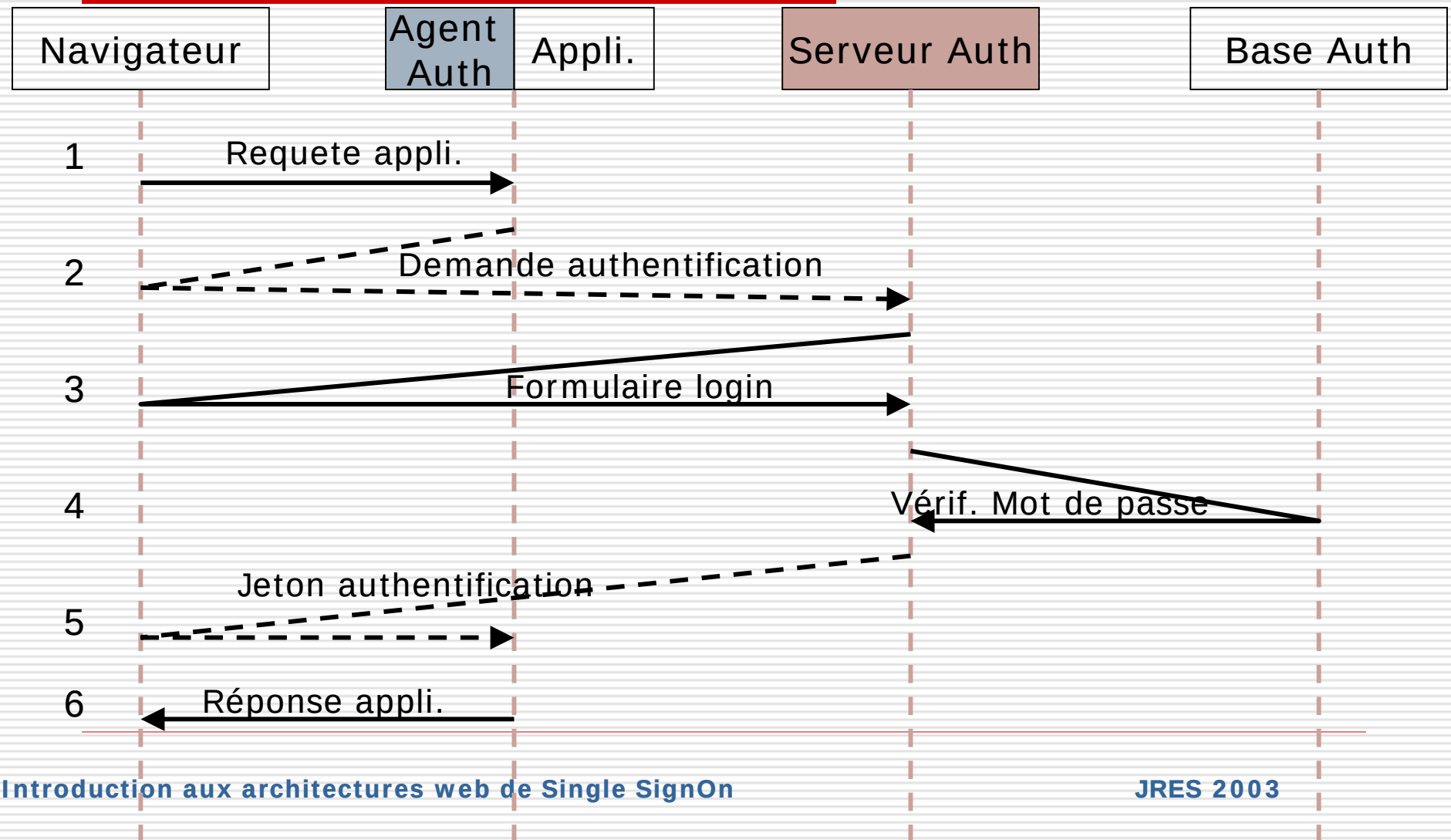
Architecture classique

Scénario d'authentification



Architecture classique

Les flux

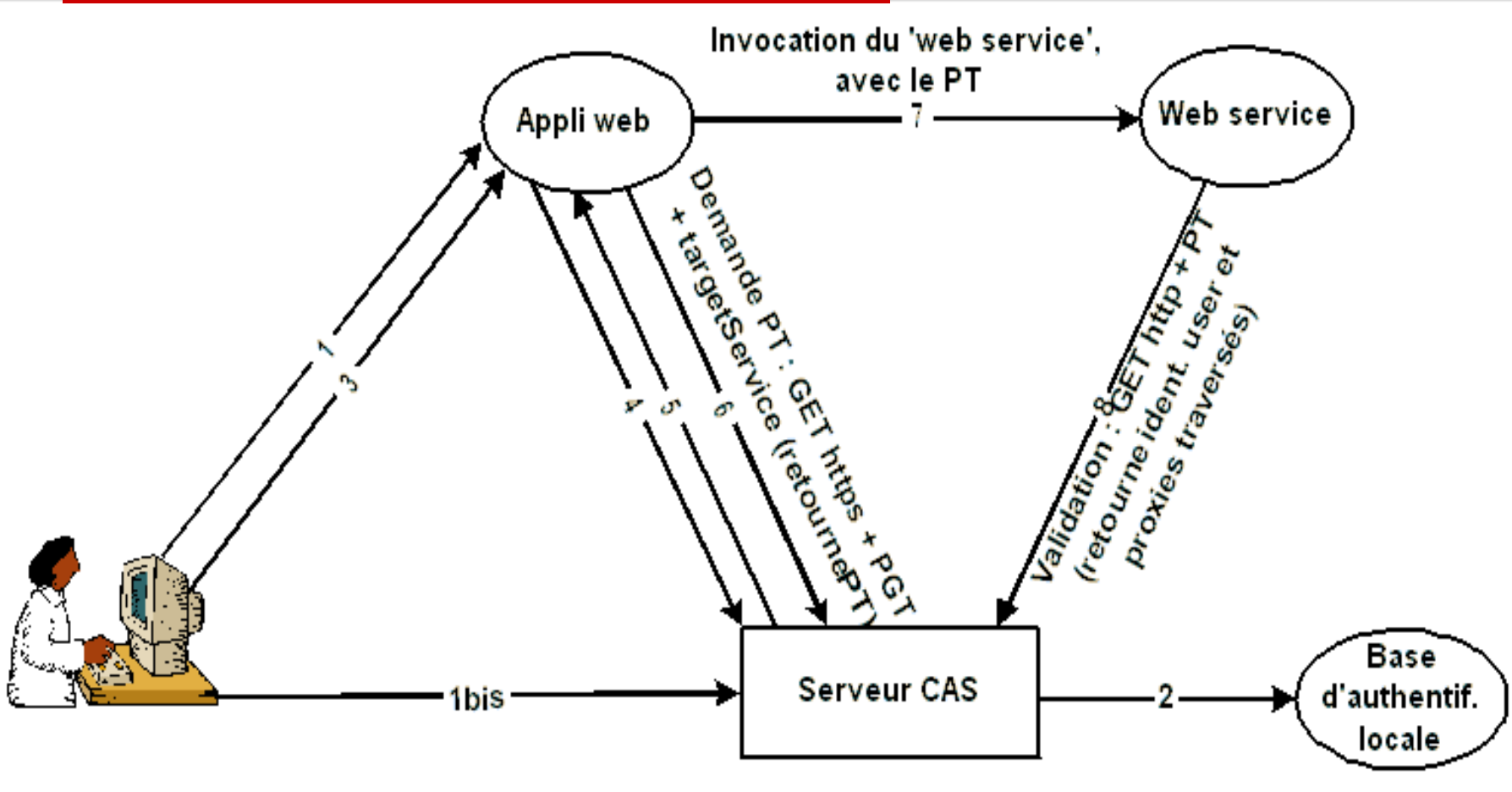


Problématique multi tiers

- Certaines applications plus compliquées à “SSO-ifier”…
- Problématique d’une architecture N-tiers :
  - Serveur auth / Portail / Canal applicatif
  - Serveur auth / Webmail / Serveur IMAP
- Architecture de + en + courante (web services)
- Des solutions peu satisfaisantes…
  - Les mots de passe transitent par l’intermédiaire
  - Confiance requise en l’application intermédiaire

Problématique multi tiers

L'architecture CAS



La solution CAS

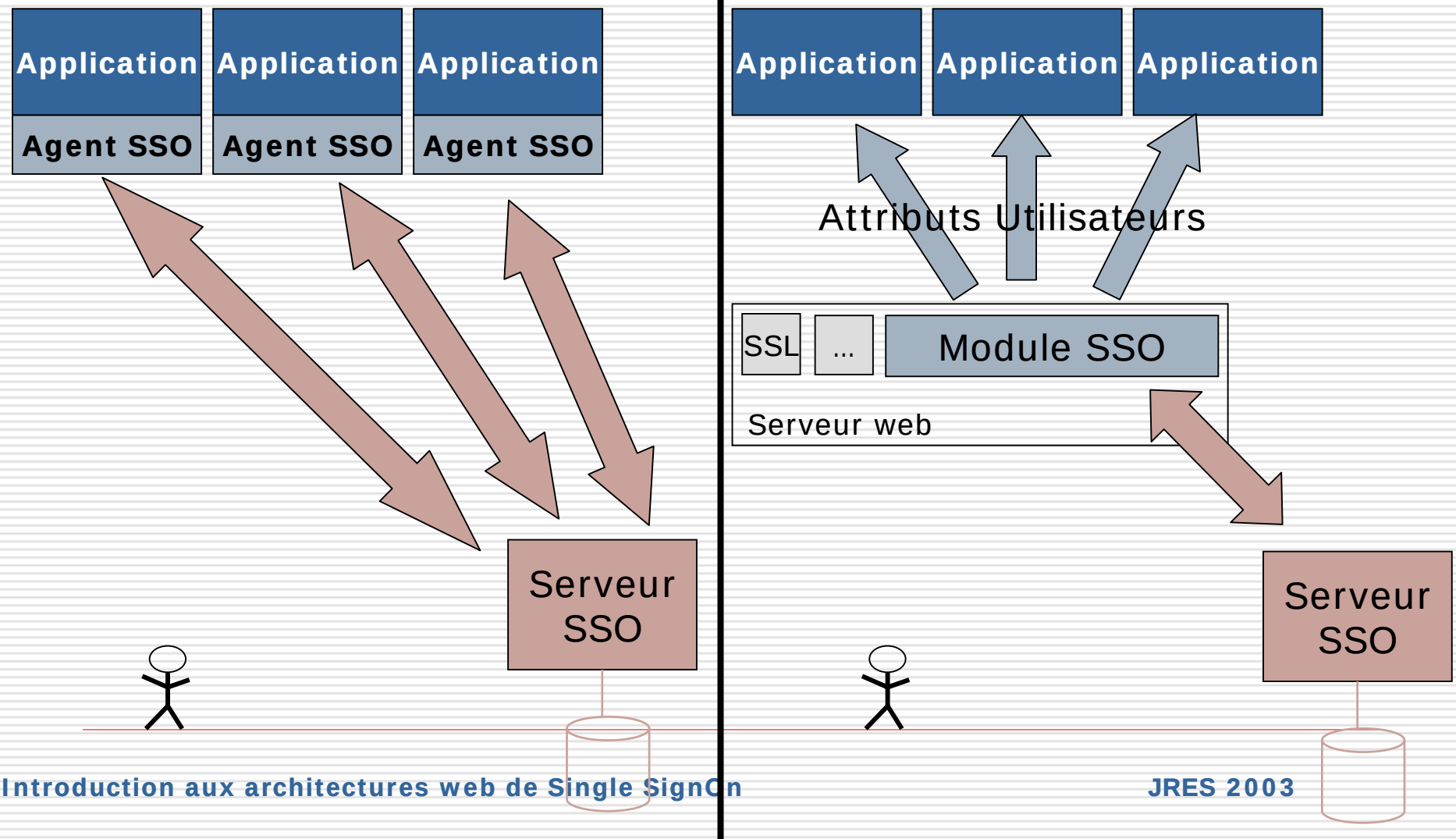
- CAS, Université de Yale
- Méthode de “Proxied credentials” :
  - Le proxy aquiert un “Proxy Granting Ticket”
  - Le proxy demande un “Service Ticket” grace à son “Proxy Granting Ticket”
  - Le “Proxy Ticket” est transmis à l’application
  - L’application échange le “Proxy Ticket” contre l’identité de l’utilisateur
- Permet la circulation de tickets opaques via plusieurs intermédiaires
- Permet de contrôler le chaînage des proxies

Intégration aux applications

- Critère essentiel dans le choix d'un SSO : quelles sont les implications au niveau des applications ?
- Pour les applications modifiables
  - Modifier l'application pour y intégrer un agent d'authentification
  - Utiliser les services d'un reverse-proxy
- Pour les applications non modifiables
  - Simuler une authentification "native"
  - Ne rien faire : tolérer des applications non SSO-ifiées

Déploiement

Le “reverse-proxy”



Vers un SSO inter-établissement

Les 3 approches

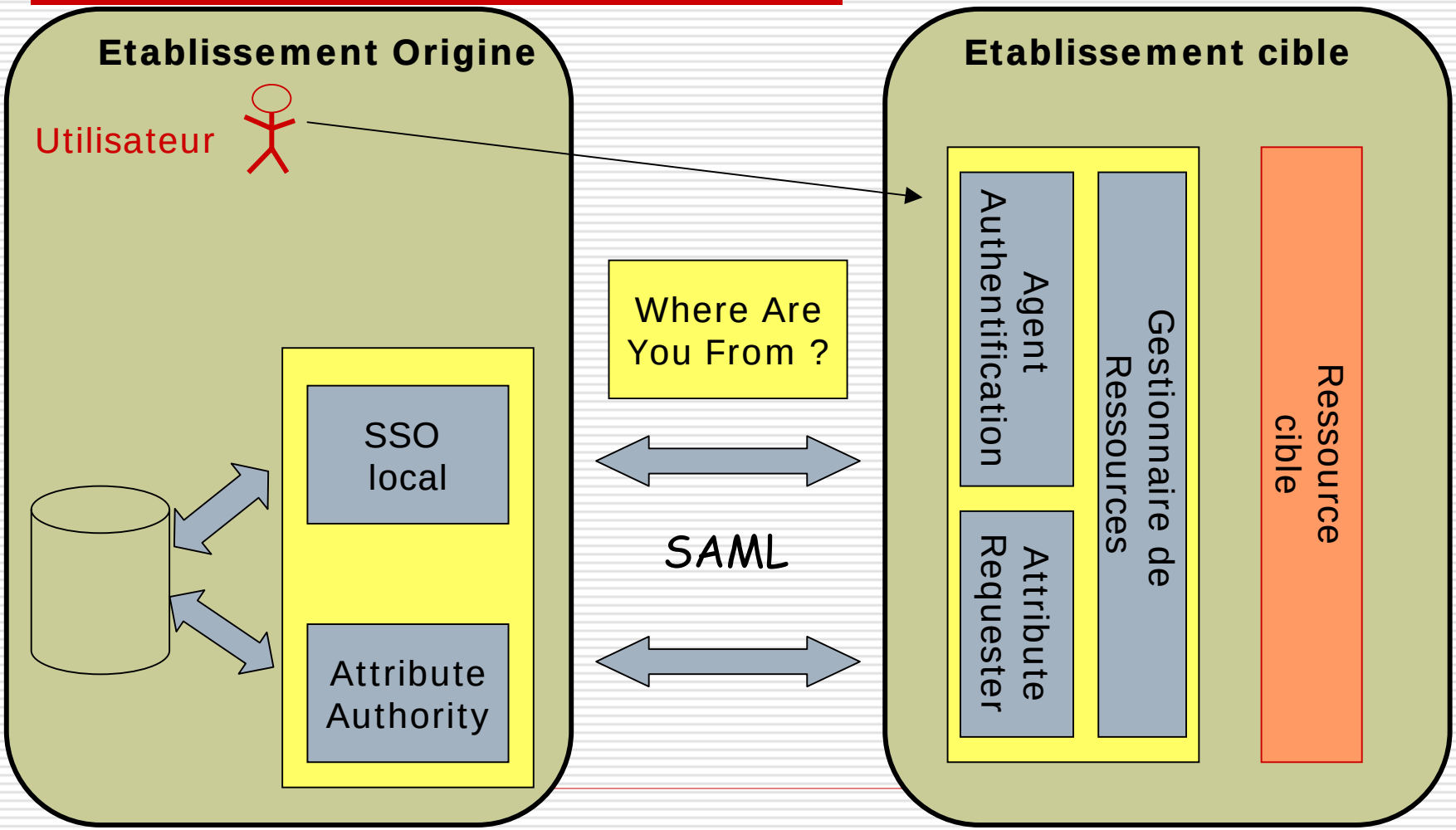
- Approche centralisée (Passport):
  - Base des utilisateurs globale
  - => pas adapté aux Universités
- Approche fédérative (Liberty Alliance) :
  - L'utilisateur doit avoir plusieurs comptes
  - Les partenaires s'échangent des informations sur l'utilisateur
  - => modèle adapté au commerce
- Approche coopérative (Shibboleth):
  - L'établissement d'origine gère l'authentification et fournit des attributs de l'utilisateur
  - Le fournisseur de ressources gère le contrôle d'accès
  - => modèle adapté au monde universitaire

Vers un SSO inter-établissement

Shibboleth

- Objectifs :
  - Contrôle d'accès à des ressources universitaires en ligne (bibliothèques)
  - Accès à des ressources hors communauté
  - Protection des données personnelles
- Fonctionnalités :
  - Connecteur entre SSO d'établissements
  - Regroupement d'établissements au sein d'un « club » ; relation de confiance (PKI)
  - Définition d'un vocabulaire commun d'attributs, basé sur eduPerson (schéma d'annuaires)

Vers un SSO inter-établissement
Architecture de Shibboleth



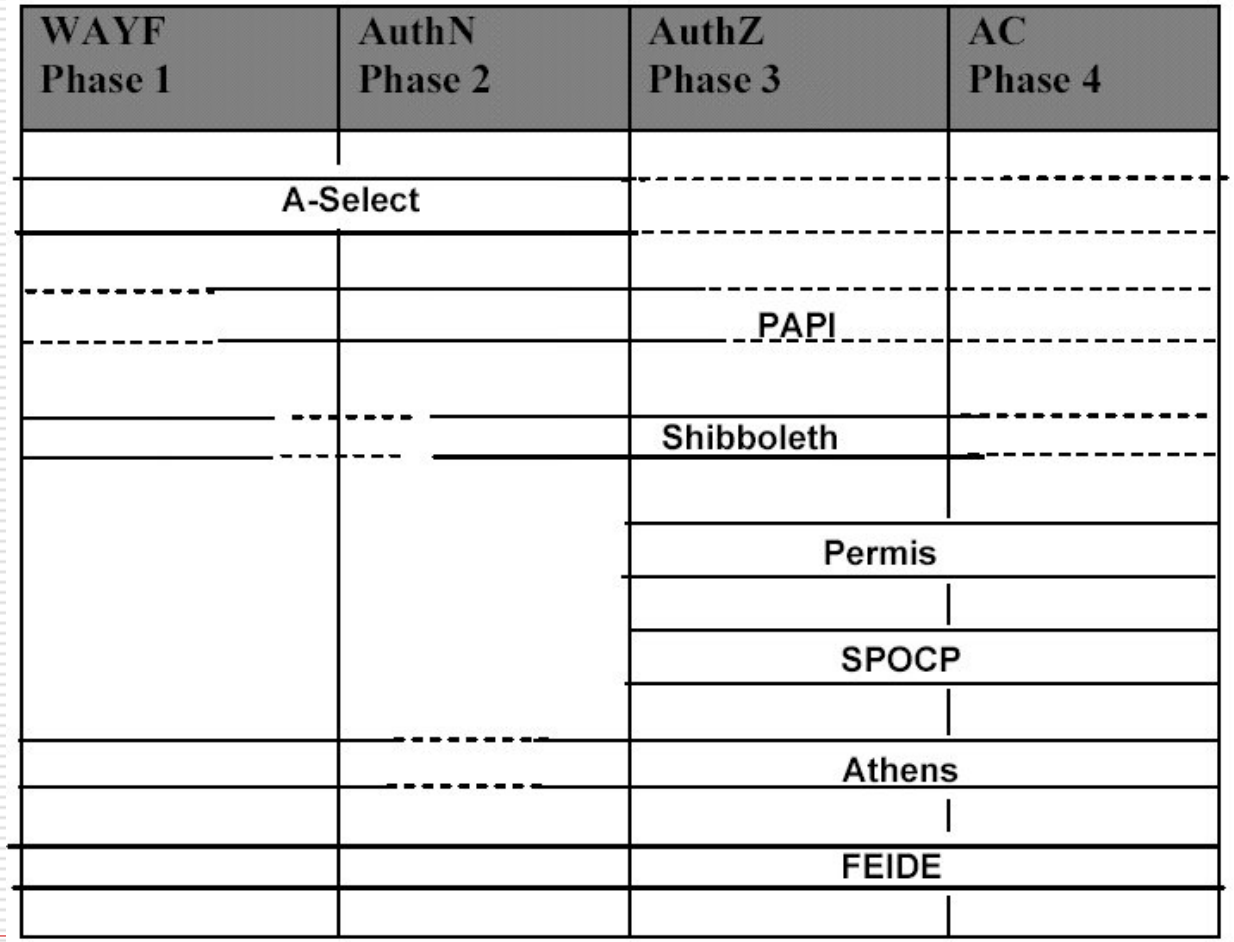
Etudes et normalisation

- Normalisation
  - WebISO (Internet 2)
  - SAML (OASIS)
  - Liberty Alliance
- TF-AACE (Terena)
  - inter- and extra-institutional A&A
- Campus numériques (SDET)
  - Groupe AAS
  - Esup-portail, CAS

Quelques produits universitaires
http://www.cru.fr/sso/

- Etats-Unis : PubCookie, C.A.S, Shibboleth
- Espagne : PAPI
- Grande-Bretagne : Athens, Sparta, Permis
- Suisse : GASPAR, TeQuiLa
- Pays-Bas : A-Select
- Suède : SPOCP
- Norvège : Moria (FEIDE)
- Finlande : FEIDHE

Quelques produits
Spectre fonctionnel

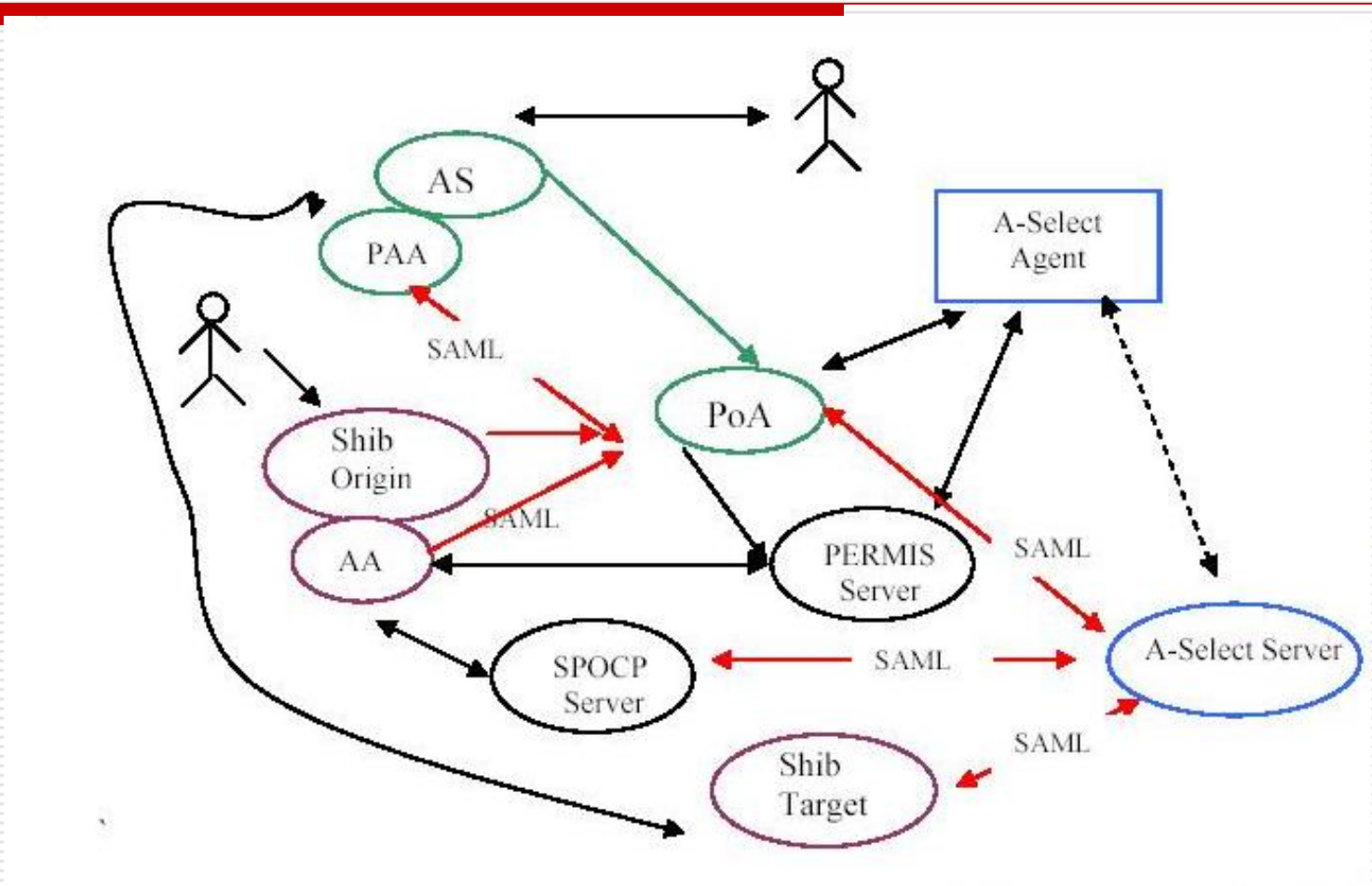


Groupe AAS

- Contexte ENT
- AAS=Authentification Autorisation SSO
- Recommandations V 1.0 (juillet 2003)
  - Service SSO : propagation des identités
  - Identifiant institutionnel (~adresse email)
  - Niveaux d'authentification
  - Propagation des identités, pas des mots de passe !
  - Autorisations gérées dans les applications ou globalement
  - SAML pour les échanges inter-établissements

Etudes et normalisation

Terena / SAML



Conclusion

- SSO, Suite logique du déploiement de LDAP
- SAML pour l'échange d'identités mais pas d'API standard SSO/applications
- SSO devient un service de base pour envisager d'autres services:
  - Délégation de l'authentification entre établissements (Shibboleth)
  - Gestion globale des comptes et des habilitations (E-provisioning)
  - Définition globale des politiques d'accès (XACML)