Shibboleth

David Verdin

-

JOSY "Authentification centralisée pour les applications web"

-

Paris - 4 février 2010

Plan de l'exposé

- Position du problème
- L'architecture de Shibboleth
- Shibboleth en action
- Do it yourself : shibboliser une application
- Sécurité
- Conclusion

Plan de l'exposé

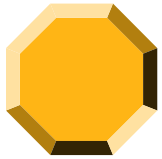
- **Position du problème**
- L'architecture de Shibboleth
- Shibboleth en action
- Do it yourself : shibboliser une application
- Sécurité
- Conclusion

Université 1

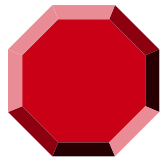
Université 2

Université 3

Université 4

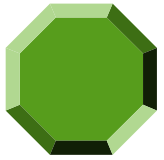


Ressource 1

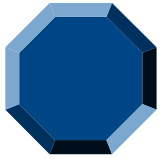


Ressource 2

Ressource 3



Ressource 4



Ressource 5

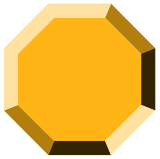
Université 1

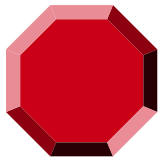
Université 2

Université 3

Université 4

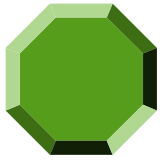
Ressource 1



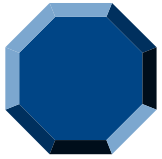


Ressource 2

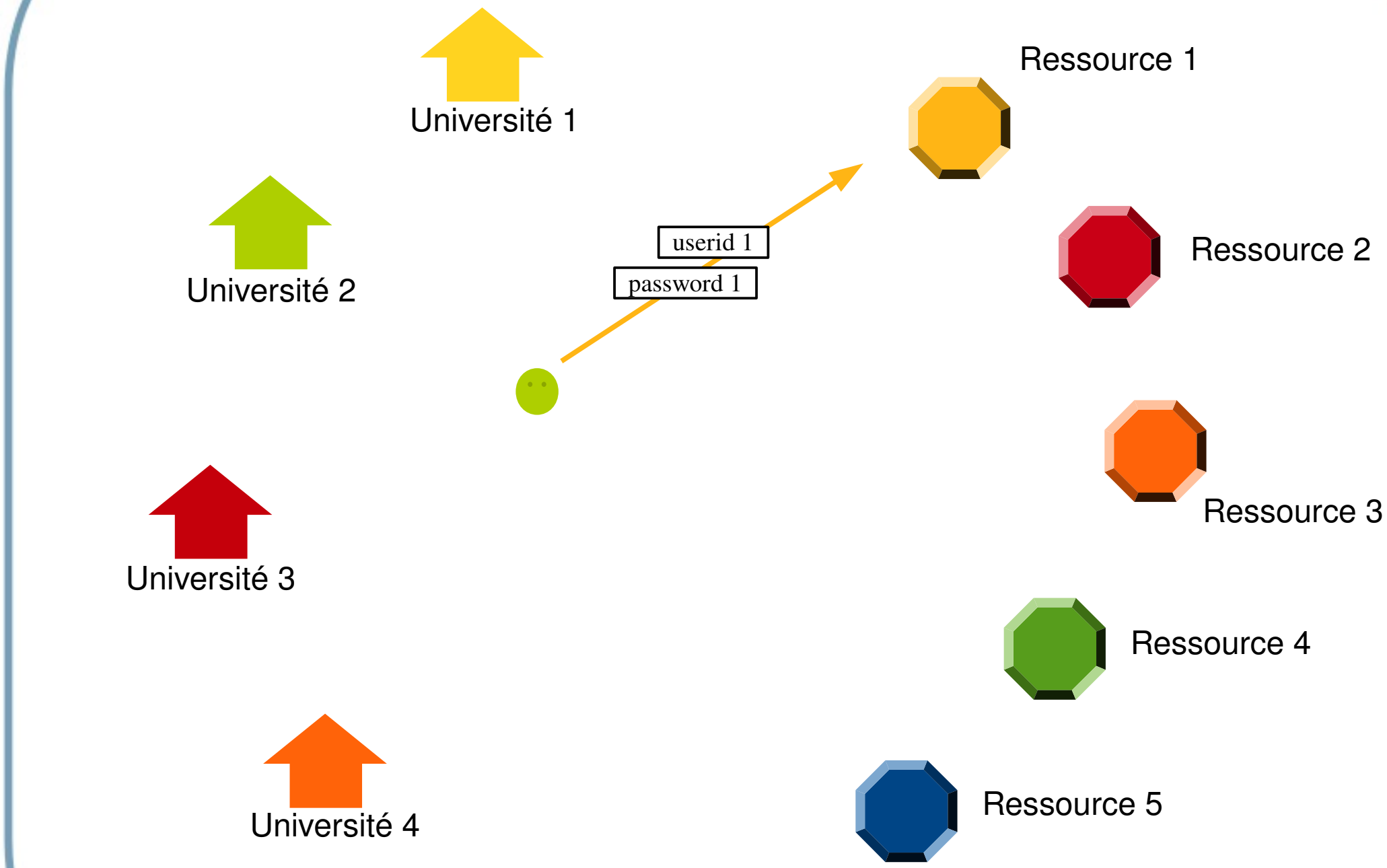
Ressource 3

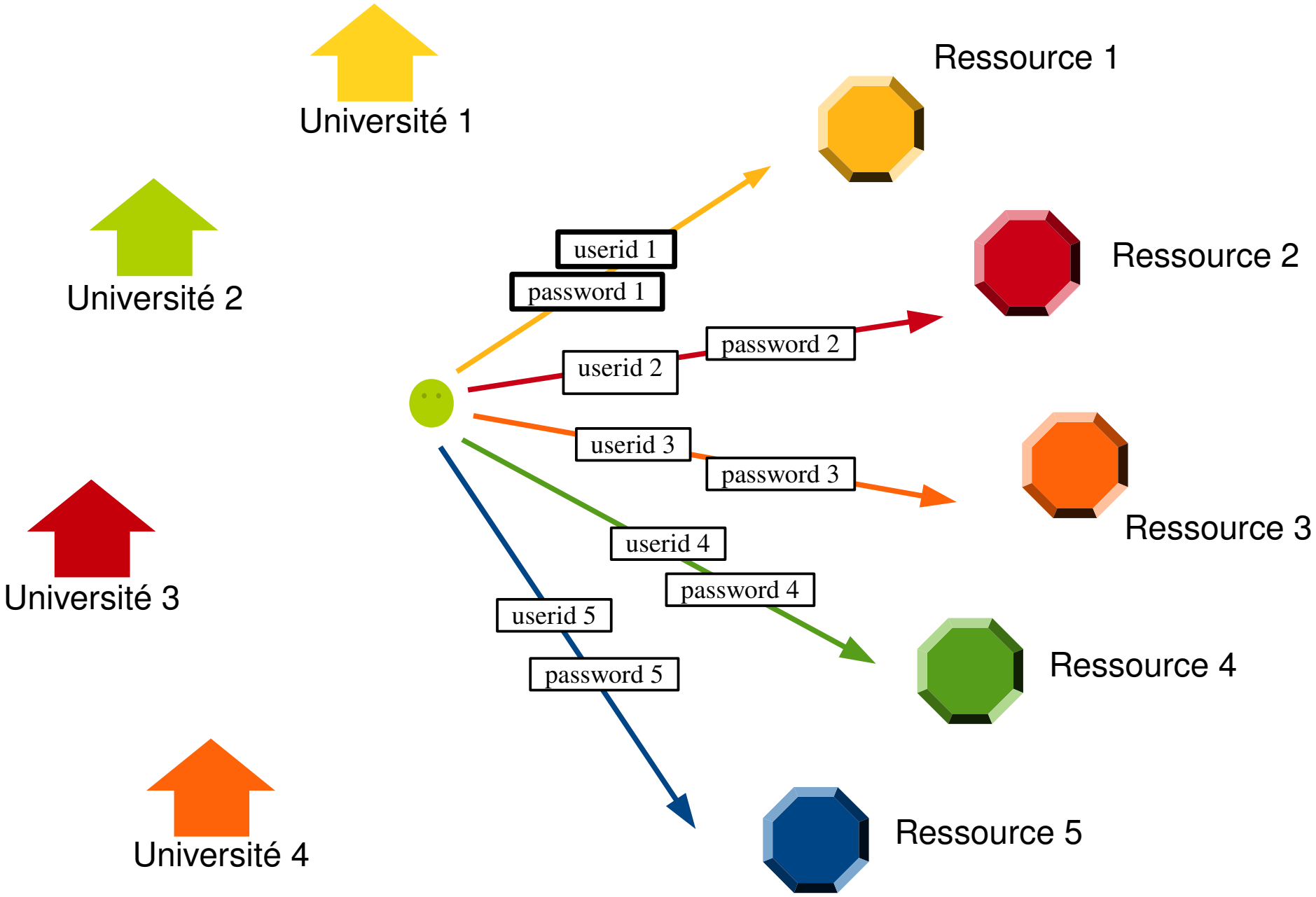


Ressource 4

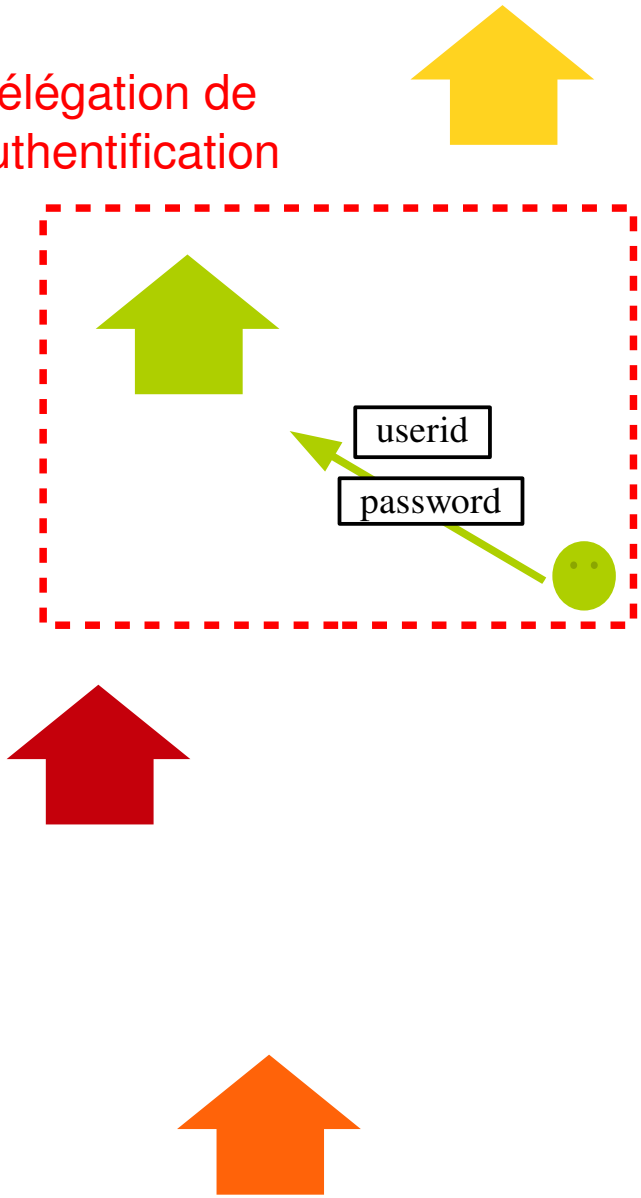


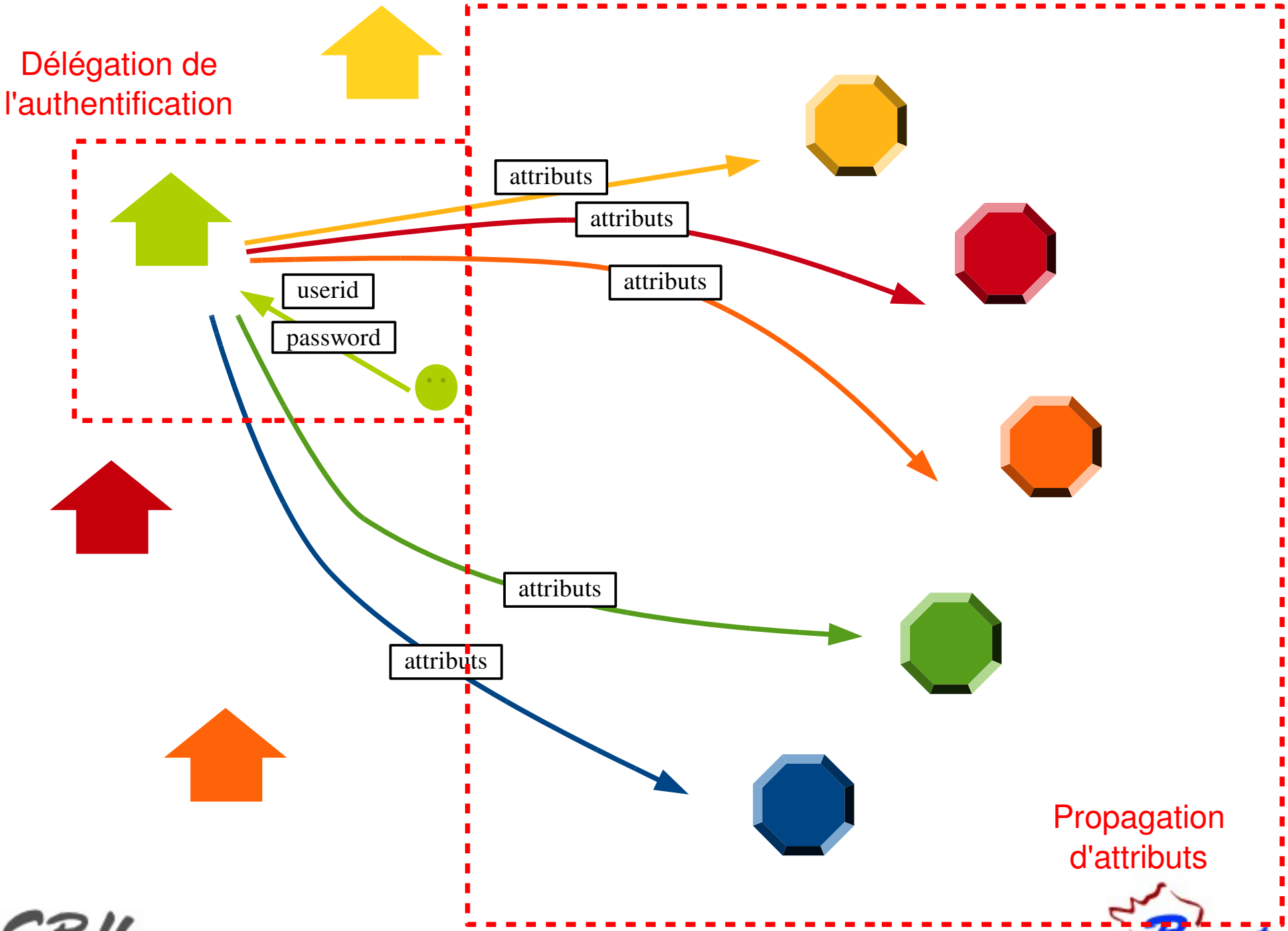
Ressource 5


Université 1
Université 2
Université 3
Université 4
Ressource 1
Ressource 2
Ressource 3
Ressource 4
Ressource 5
userid 1
password 1


Université 1
Université 2
Université 3
Université 4
Ressource 1
Ressource 2
Ressource 3
Ressource 4
Ressource 5
userid 1
password 1
userid 2
password 2
userid 3
password 3
userid 4
password 4
userid 5
password 5

Délégation de l'authentification



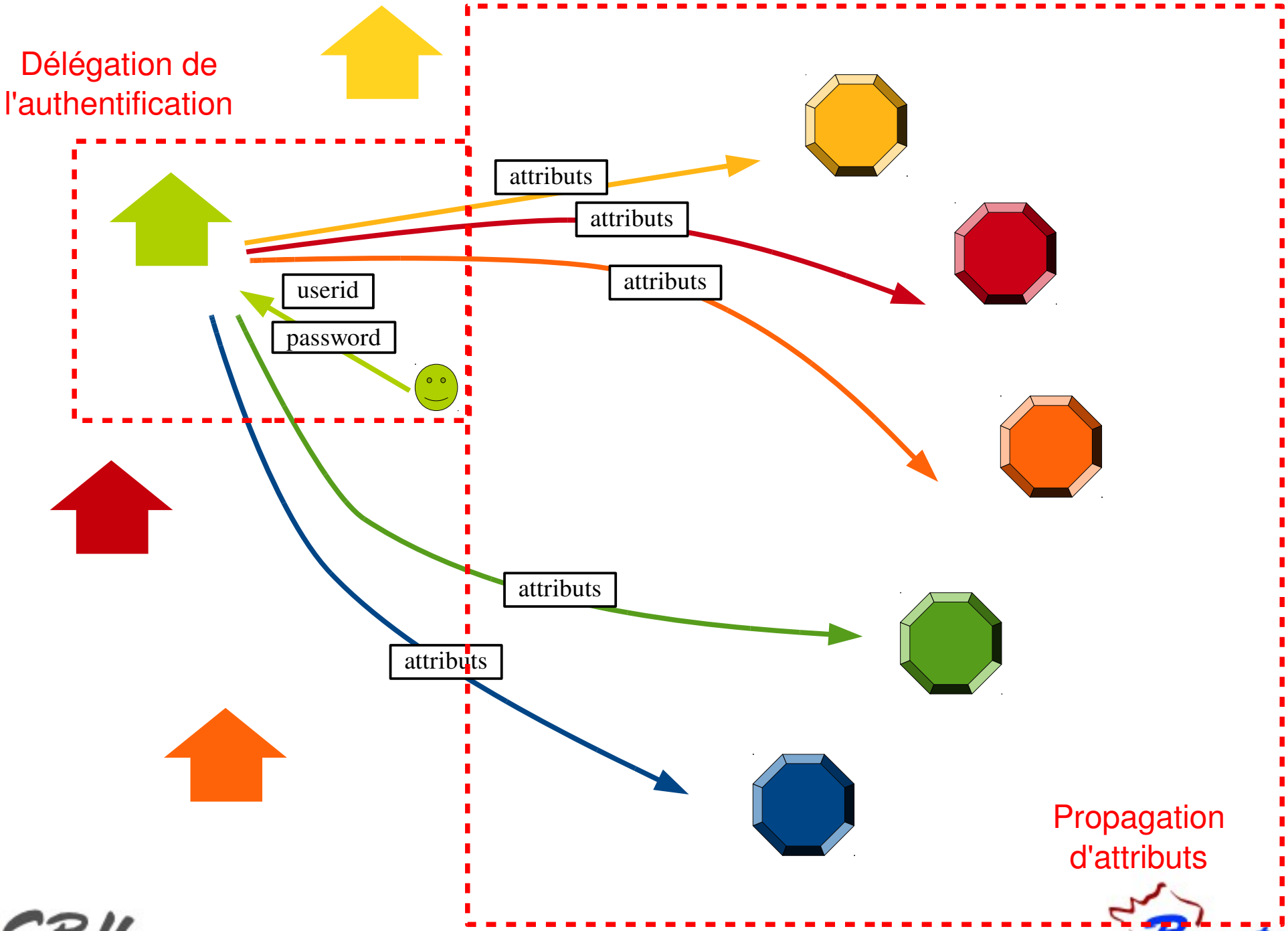

Délégation de
l'authentification
attributs
attributs
attributs
userid
password
attributs
attributs
Propagation
d'attributs

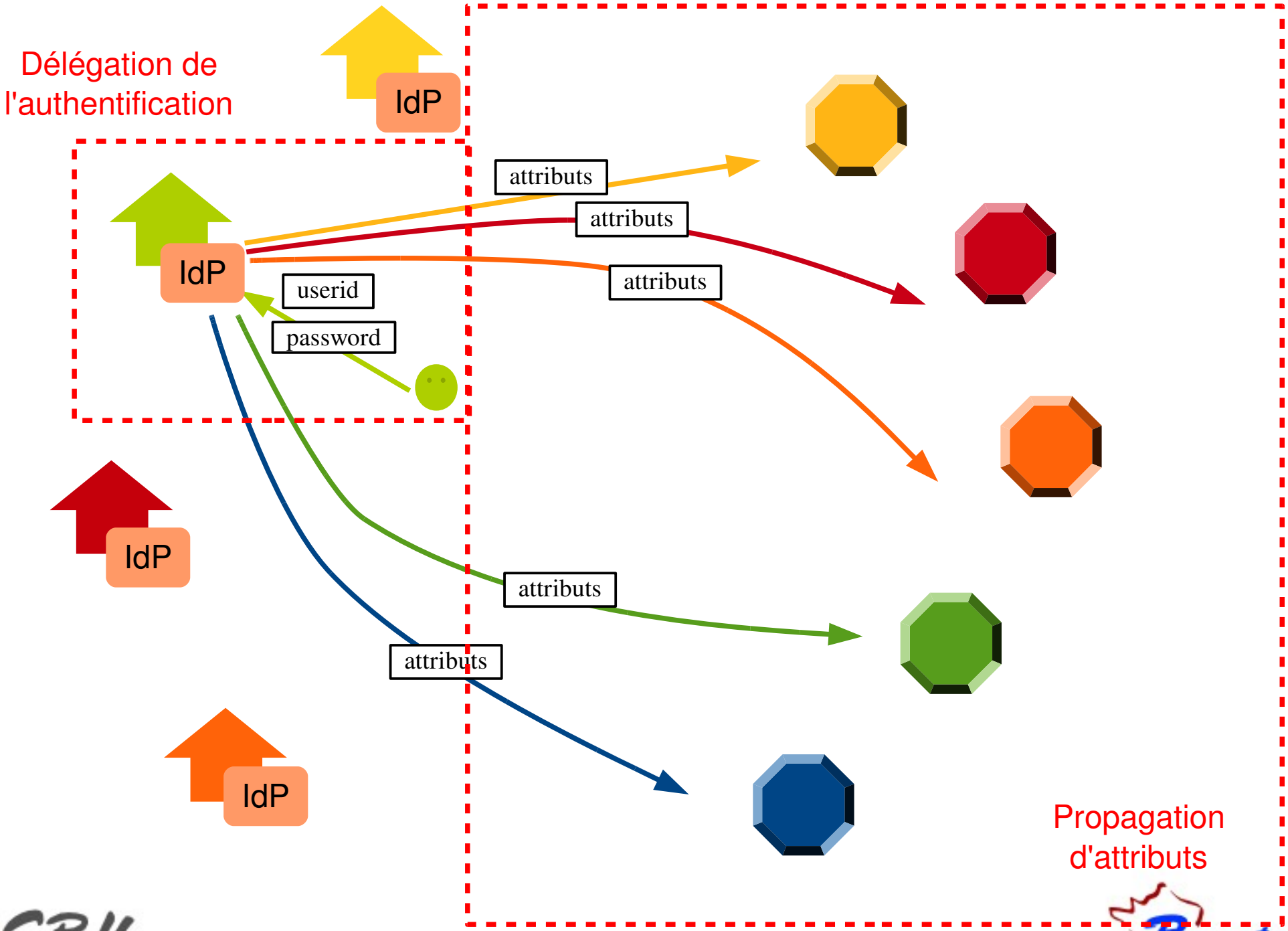
SAML

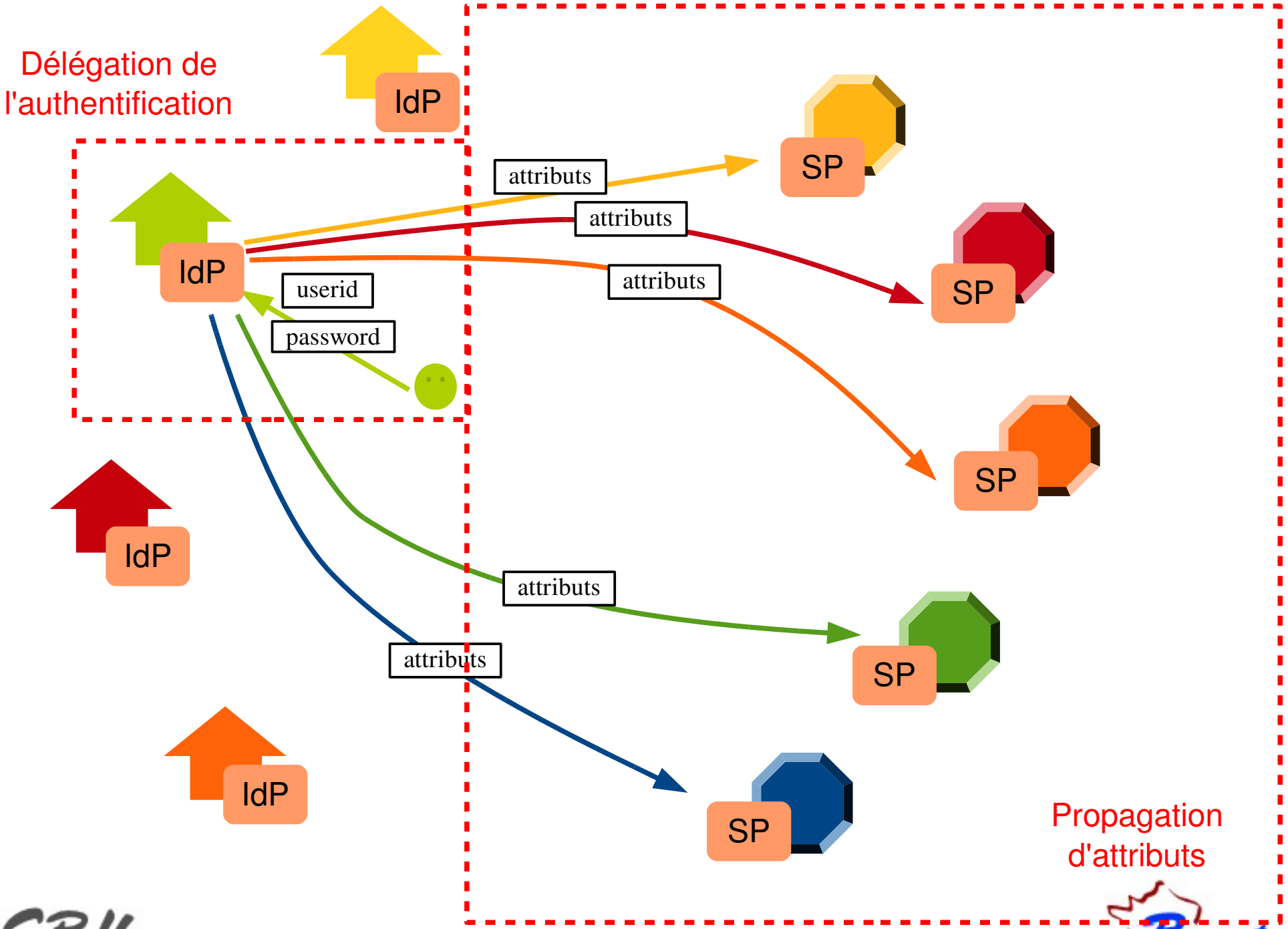
- Security assertion markup language
- Langage XML formalisant les échanges entre membres d'une fédération d'identité
- Standard Oasis depuis 2001
- Communication entre membres d'une fédération sous forme d'**assertions SAML** : fragments XML diffusés via HTTP.
- Haut niveau d'abstraction : plusieurs profils dérivés
- Implémentations :
  - Liberty Alliance
  - SimpleSAMLPHP
  - **Shibboleth**

Plan de l'exposé

- Position du problème
- **L'architecture de Shibboleth**
- Shibboleth en action
- Do it yourself : shibboliser une application
- Sécurité
- Conclusion


Délégation de
l'authentification
attributs
attributs
attributs
userid
password
attributs
attributs
Propagation
d'attributs


Délégation de
l'authentification
IdP
IdP
userid
password
attributs
attributs
attributs
IdP
attributs
attributs
IdP
Propagation
d'attributs


Délégation de
l'authentification
IdP
IdP
IdP
IdP
userid
password
attributs
attributs
attributs
attributs
attributs
SP
SP
SP
SP
SP
Propagation
d'attributs

SP et IdP : qu'est-ce que c'est ?

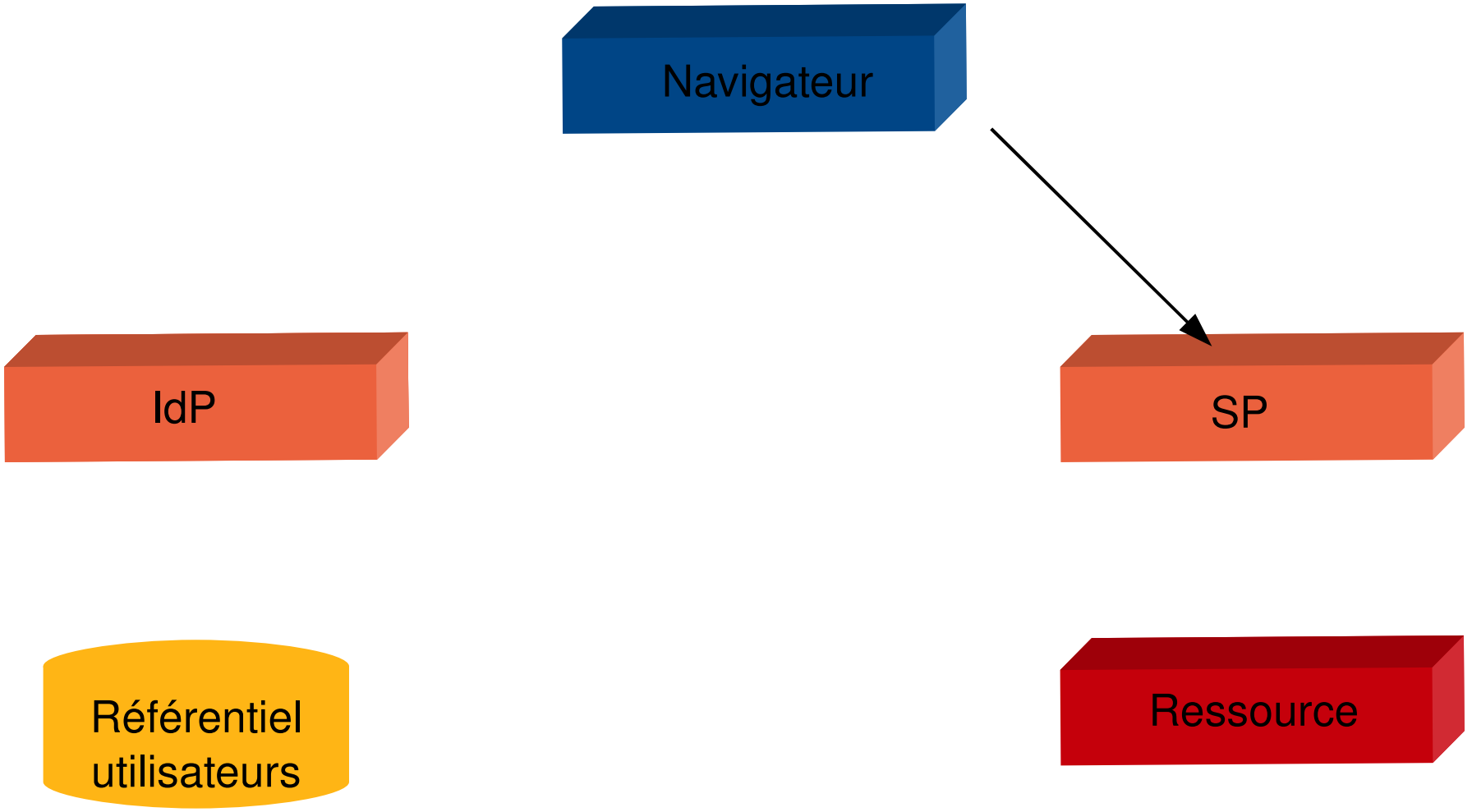
- **Service provider** (SP) = brique côté ressource
  - module Apache + démon shibd
  - Interfacé avec une application web

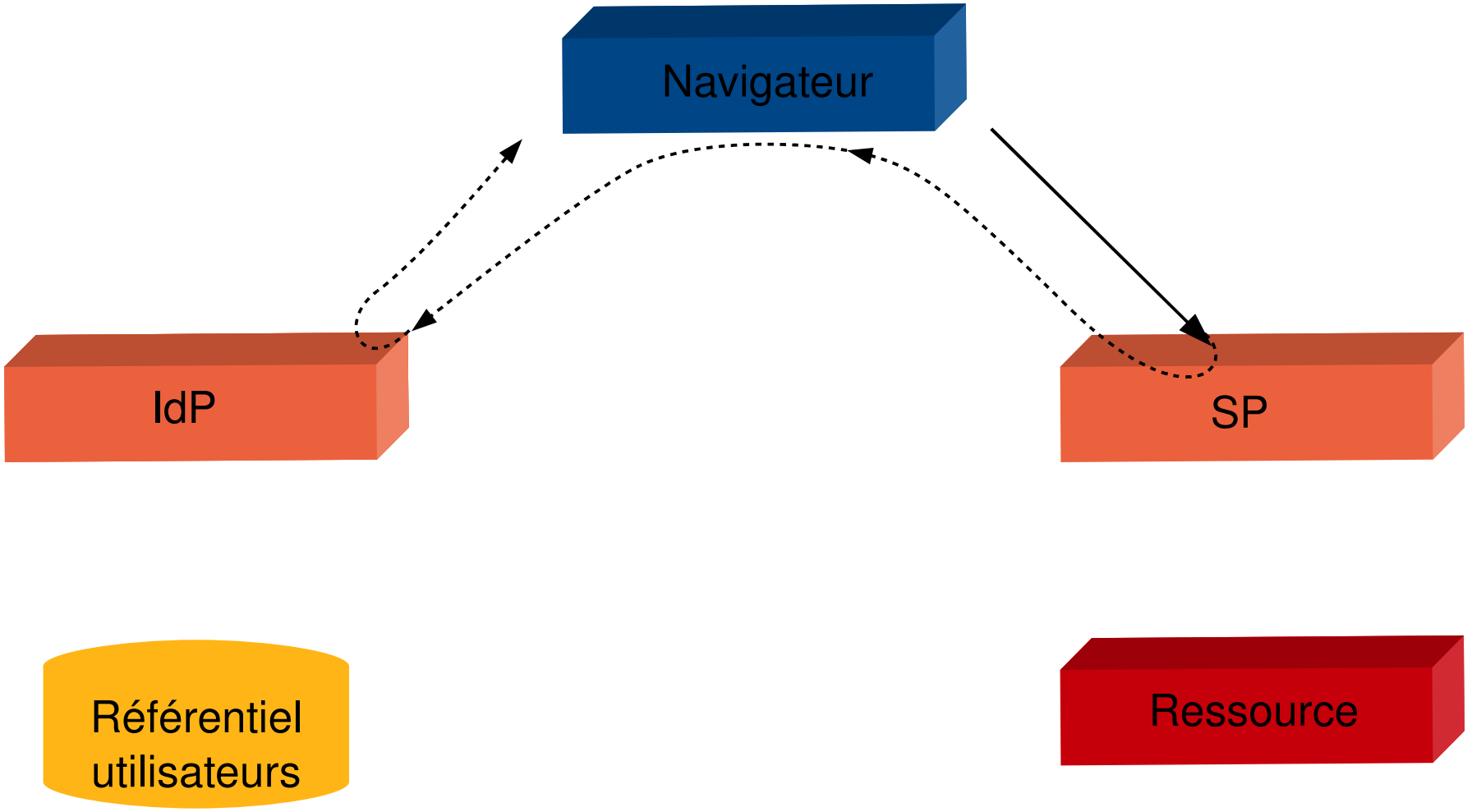
- **Identity provider** (IdP) = brique côté authentification
  - servlet java (Tomcat)
  - Interfacé avec le SI

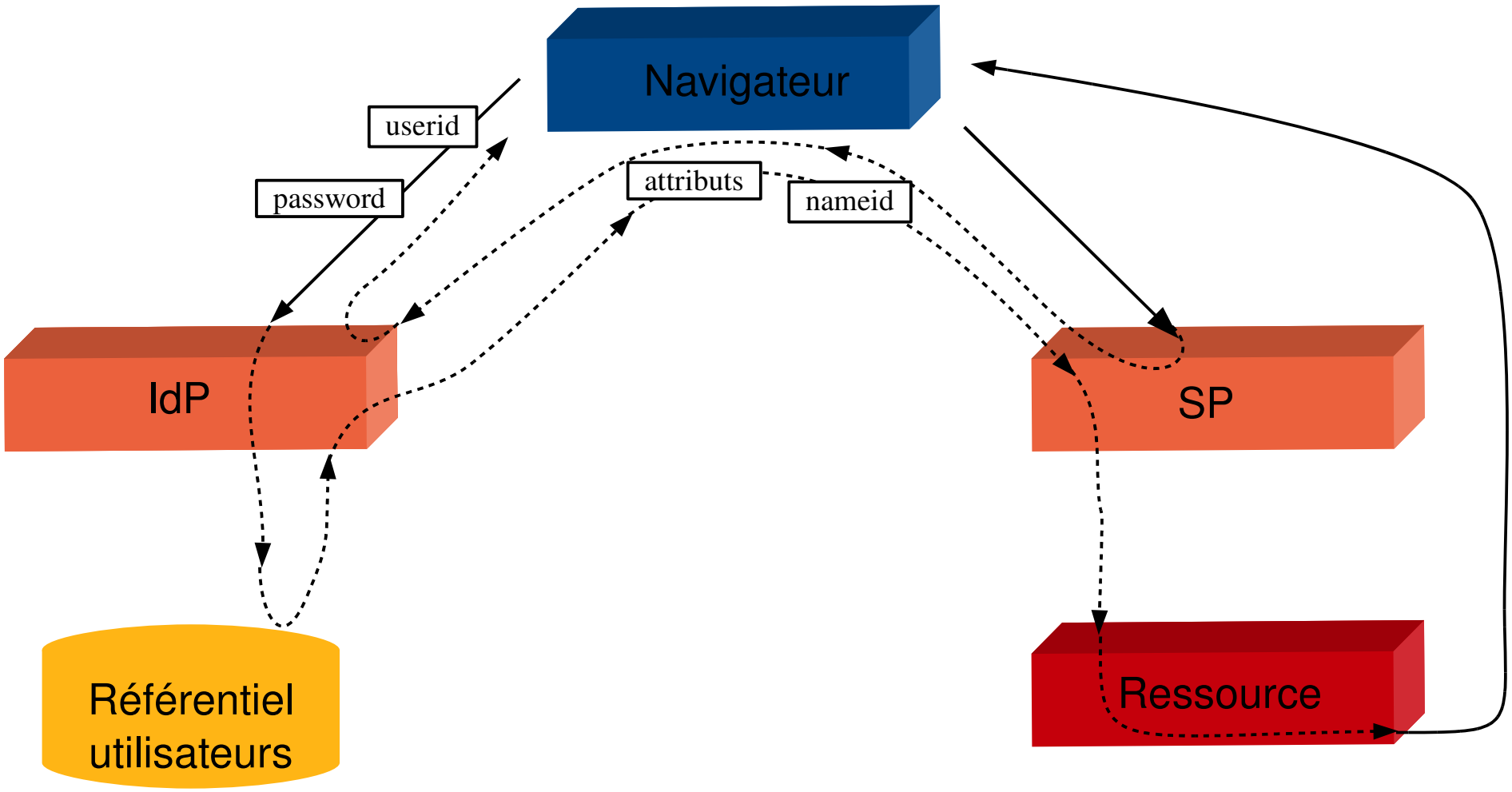
- Pour les installer : Voir les TP de la fédération Éducation recherche :
  - https://services-federation.renater.fr/doc/support-tp-idp.odt
  - https://services-federation.renater.fr/doc/support-tp-sp.odt

Plan de l'exposé

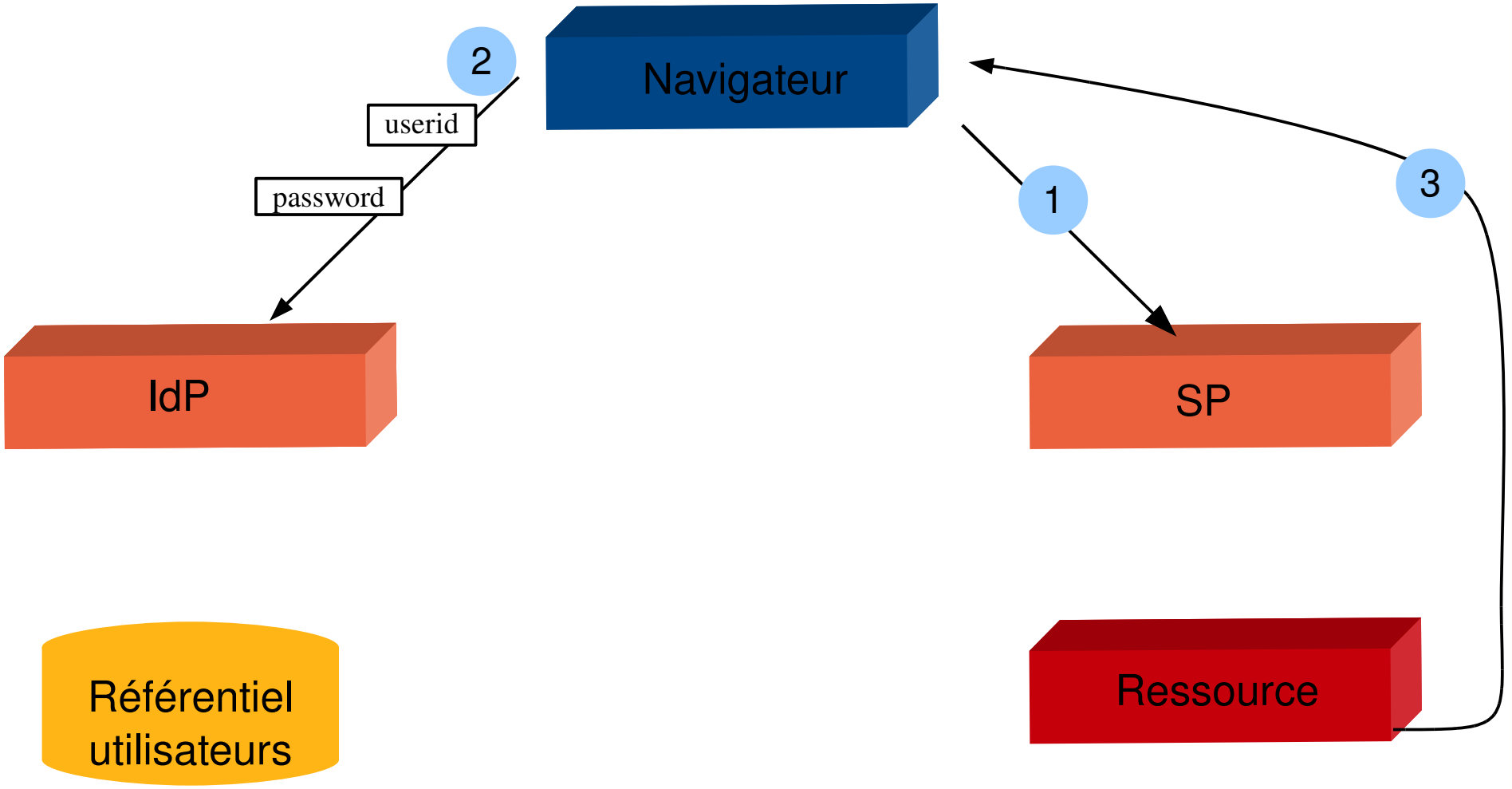
- Position du problème
- L'architecture de Shibboleth
- **Shibboleth en action**
  - ***Lien direct***
  - *Lien direct + SSO*
  - *Utilisation d'un WAYF*
- Do it yourself : shibboliser une application
- Sécurité
- Conclusion


Navigateur
IdP
SP
Référentiel utilisateurs
Ressource

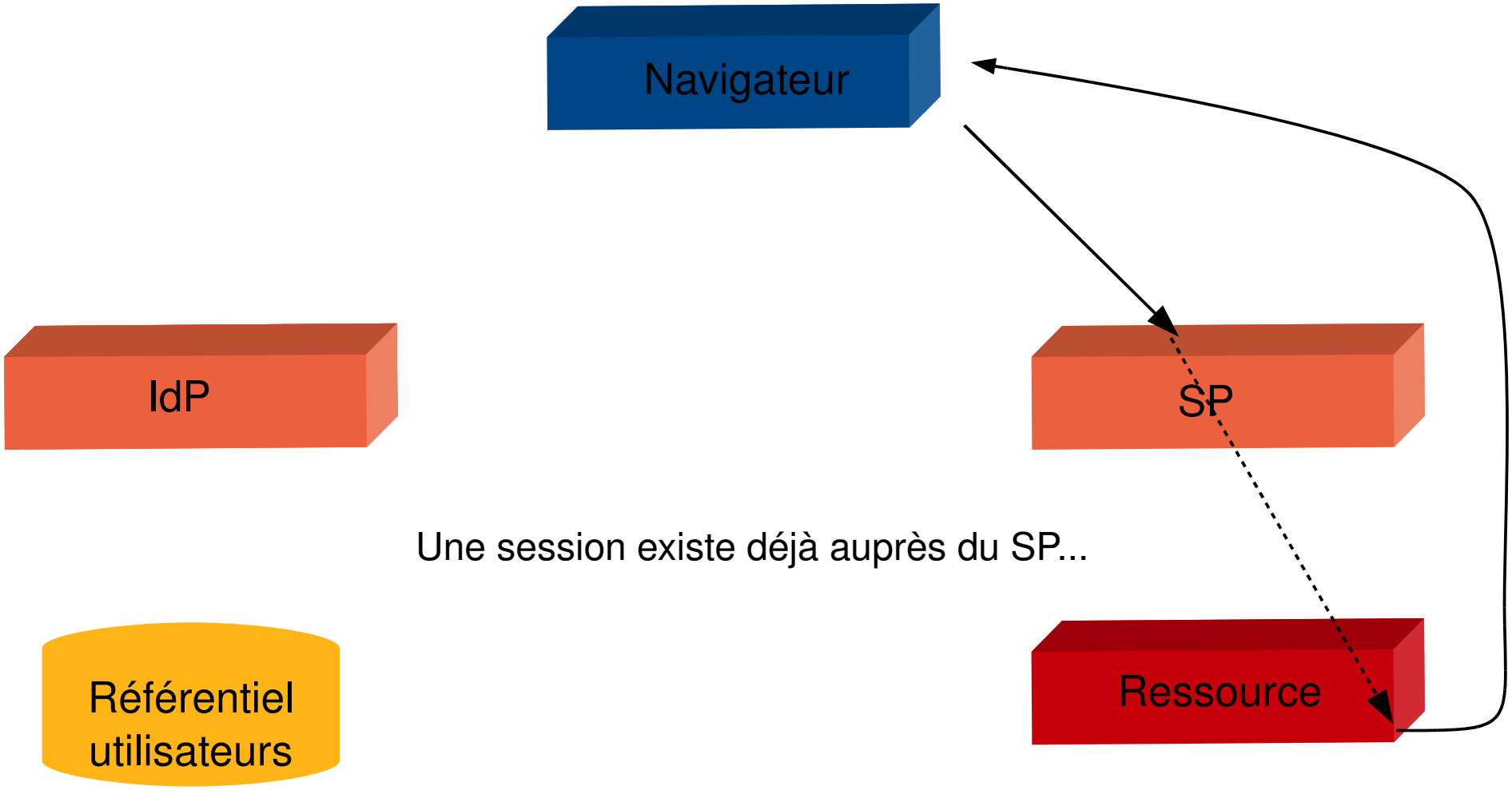

Navigateur
IdP
SP
Référentiel utilisateurs
Ressource


Navigateur
userid
password
attributs
nameid
IdP
SP
Référentiel utilisateurs
Ressource

Du point de vue de l'utilisateur, ça donne :

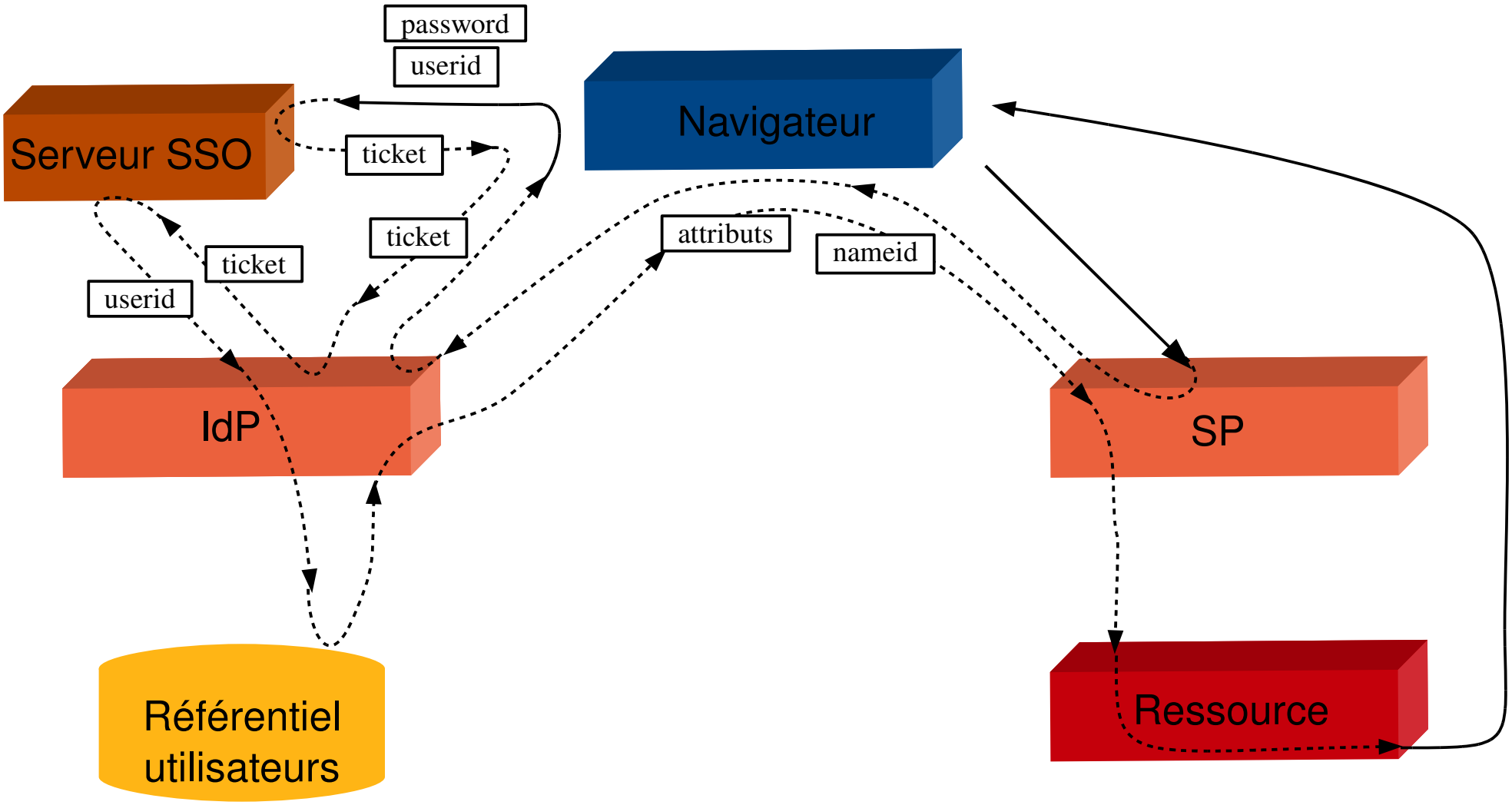


Requêtes suivantes vers le même SP :

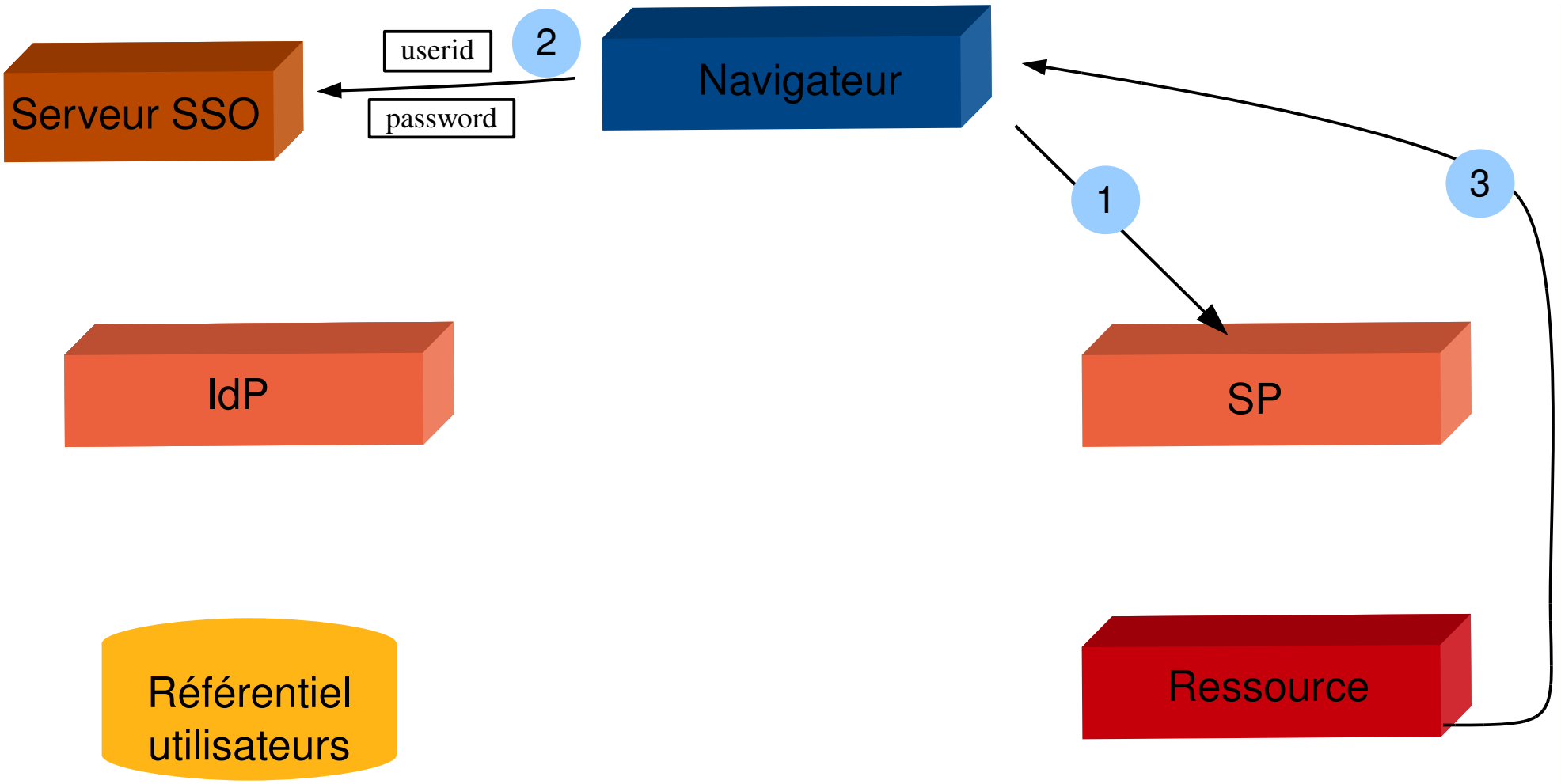


Plan de l'exposé

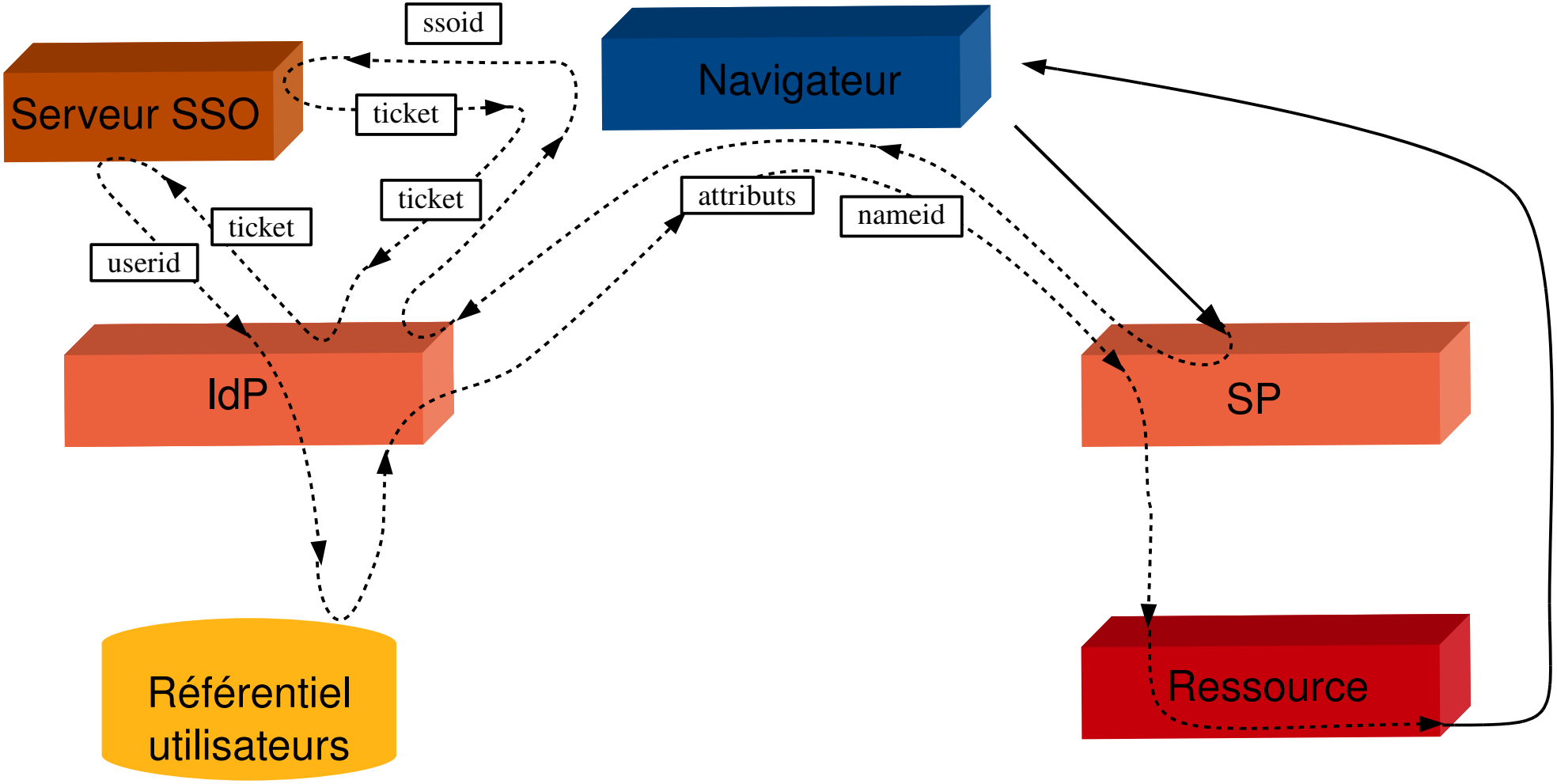
- Position du problème
- L'architecture de Shibboleth
- **Shibboleth en action**
  - *Lien direct*
  - ***Lien direct + SSO***
  - *Utilisation d'un WAYF*
- Do it yourself : shibboliser une application
- Sécurité
- Conclusion


password
userid
Serveur SSO
ticket
Navigateur
ticket
ticket
attributs
nameid
userid
IdP
SP
Référentiel
utilisateurs
Ressource

Du point de vue de l'utilisateur, ça donne maintenant :



Requêtes suivantes vers un autre SP

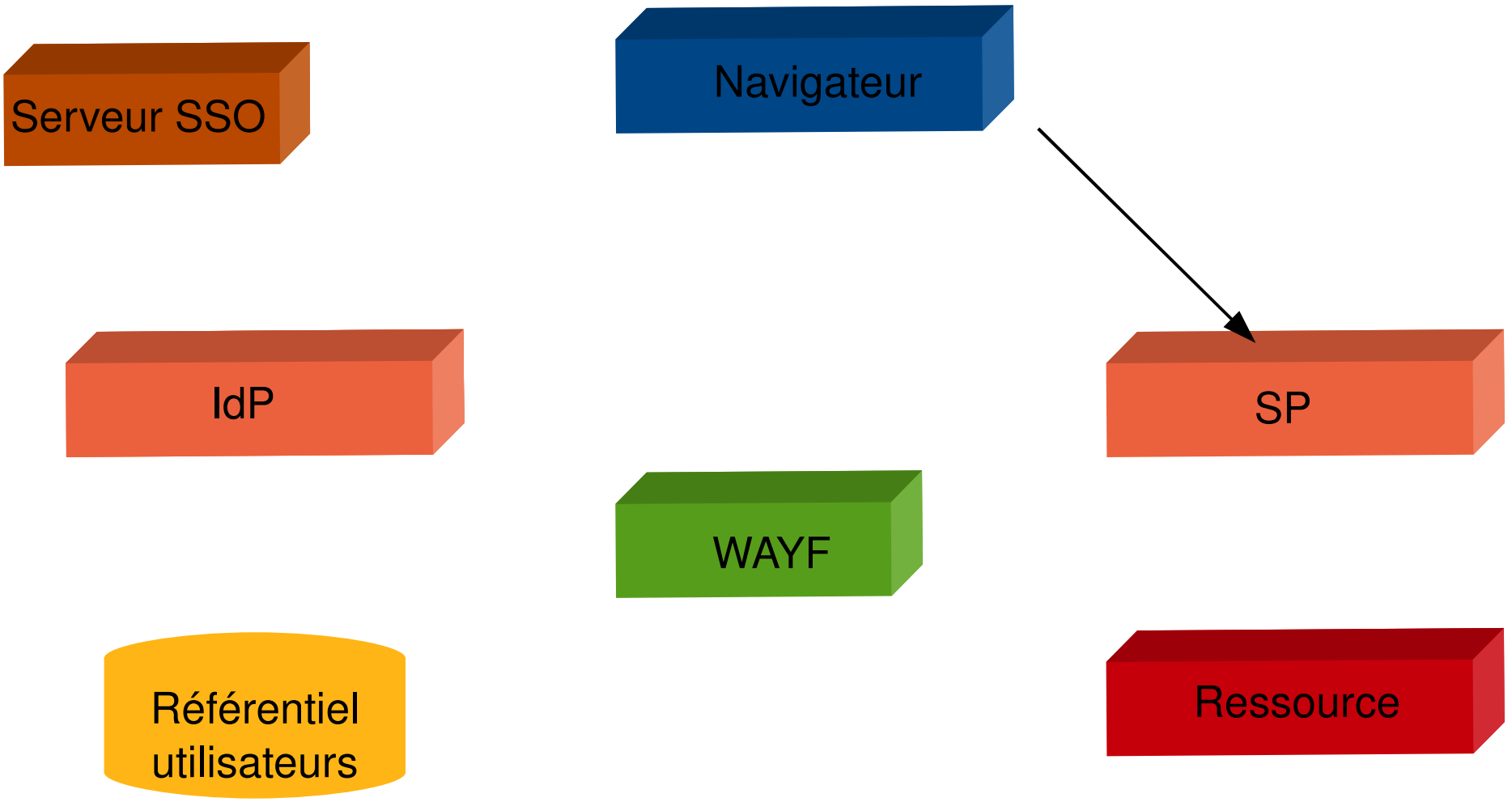


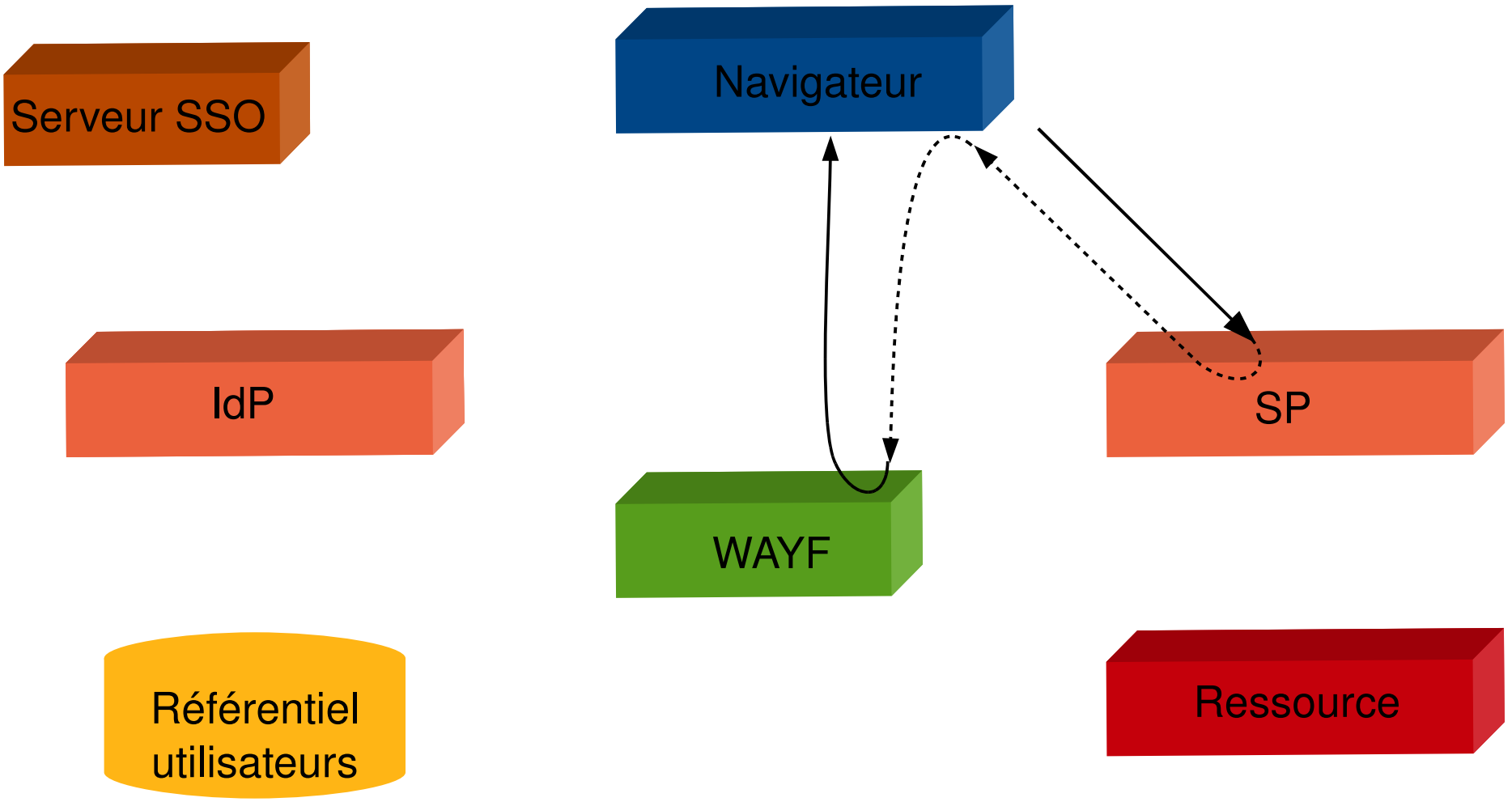
Plan de l'exposé

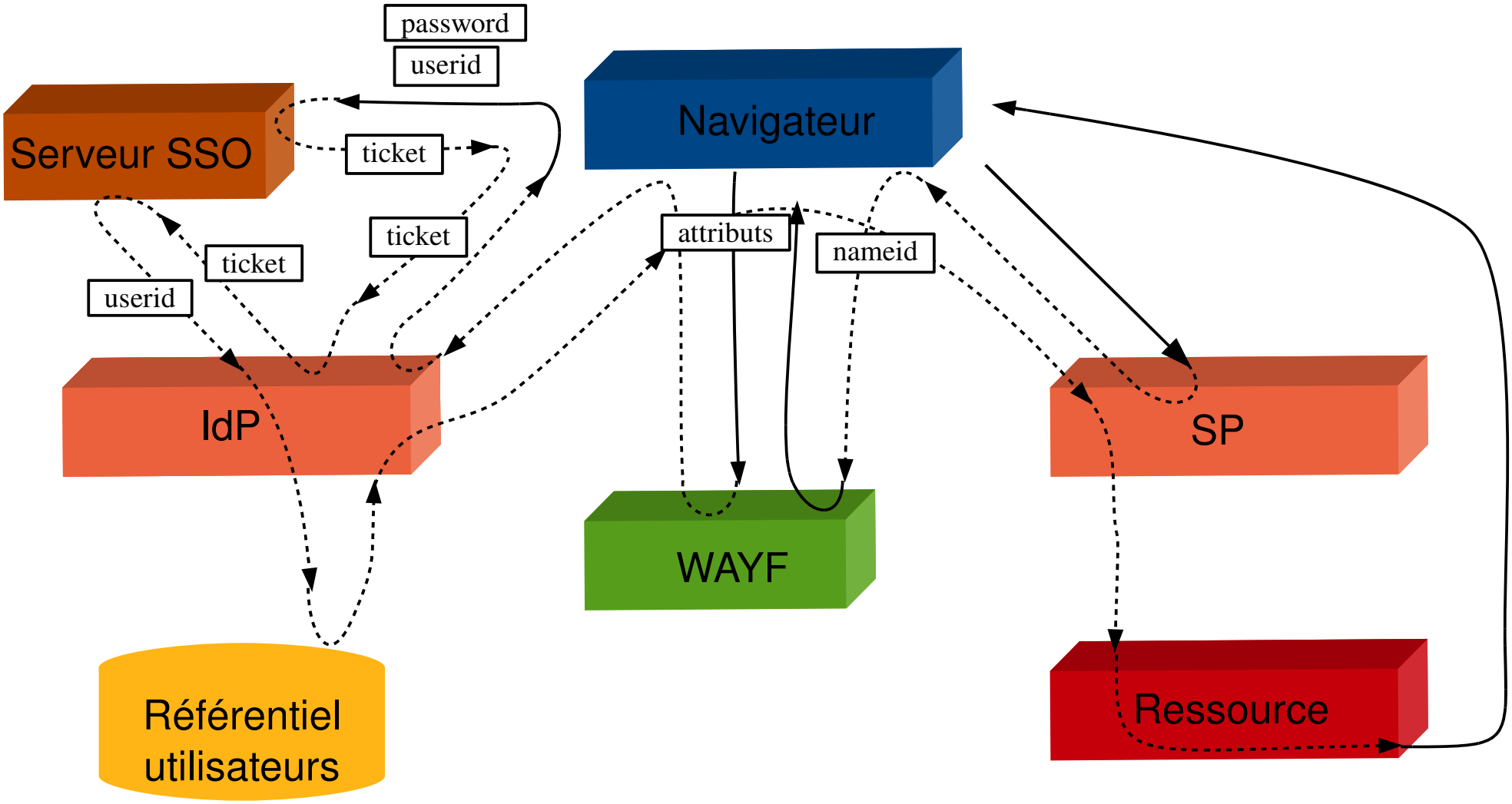
- Position du problème
- L'architecture de Shibboleth
- **Shibboleth en action**
  - *Lien direct*
  - *Lien direct + SSO*
  - ***Utilisation d'un WAYF***
- Do it yourself : shibboliser une application
- Sécurité
- Conclusion

WAYF

- = Where are you from?
- Brique supplémentaire : permet de choisir son IdP
- Optionnel, mais indispensable quand le nombre de membres de la fédération augmente.


Serveur SSO
Navigateur
IdP
SP
WAYF
Référentiel utilisateurs
Ressource


Serveur SSO
Navigateur
IdP
SP
WAYF
Référentiel utilisateurs
Ressource


password
userid
Serveur SSO
ticket
Navigateur
ticket
ticket
attributs
nameid
userid
IdP
SP
WAYF
Référentiel
utilisateurs
Ressource

WAYF et ergonomie

- Une étape de plus
- Rend l'infrastructure de la fédération apparente
- Problème des personnes hors fédération
- Diverses solutions à l'étude :
  - Une « marque fédération » : un nouveau point de repère
  - Intégrer le WAYF à l'application
  - WAYF centralisé : plutôt pour aider les fournisseurs de ressources.

Plan de l'exposé

- Position du problème
- L'architecture de Shibboleth
- Shibboleth en action
- **Do it yourself : shibboliser une application**
- Sécurité
- Conclusion

Do it yourself : shibboliser une application

- Ampleur de la tache variable
- Mettre en place une session applicative
- Intégrer un WAYF dans l'application
- Architecture proxy
- Désactiver l'usage des mots de passe
- Utiliser un mot de passe « statique »
- Nommage Supann des attributs
- Des messages d'erreur précis
- Un plugin Shibboleth paramétrable

Do it yourself... sauf si a déjà été fait !

- Internet2 référence les applications et services shibbolisés : https://spaces.internet2.edu/display/SHIB2/ShibEnabled
- On peut citer notamment :
  - Sympa
  - Dokuwiki, Mediawiki
  - Drupal
  - uPortal
  - In progress au CRU : Foodle, Limesurvey, Big file sharing...

Plan de l'exposé

- Position du problème
- L'architecture de Shibboleth
- Shibboleth en action
- Do it yourself : shibboliser une application
- **Sécurité**
- Conclusion

Sécurité : le contrôle des attributs

- **ARP** (IdP): Attributes release policy
- Un fichier XML : attribute-filter.xml
- Contrôler, au niveau de l'IdP, les attributs qui seront envoyés à chaque SP.
  - Exemple : tous les SP n'ont pas à savoir si un utilisateur est étudiant, professeur ou chercheur...
- **AAP** (SP): Attributes acceptance policy : contrôler les valeurs des attributs renvoyés par les IdP

Sécurité : le contrôle des attributs

```
<?xml version="1.0" encoding="UTF-8"?>
<AttributeReleasePolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                        xmlns="urn:mace:shibboleth:arp:1.0"
                        xsi:schemaLocation="urn:mace:shibboleth:arp:1.0 shibboleth-arp-1.0.xsd">
   <Description>Simplest possible ARP.</Description>
   <Rule>
      <Target>
        <AnyTarget/>
      </Target>
      <Attribute name="urn:mace:dir:attribute-def:eduPersonAffiliation">
         <AnyValue release="permit"/>
      </Attribute>
   </Rule>
</AttributeReleasePolicy>
```

Exemple de fichier attribute-filter.xml : donner « eduPersonAffiliation »
À tous les SP

Sécurité : le contrôle des attributs

```
<AttributeFilterPolicy id="releaseToSpExampleOrg">

    <!-- Policy requirement rule that indicates this policy is only used for requests from
http://sp.example.org -->
    <PolicyRequirementRule xsi:type="basic:AttributeRequesterString"
                           value="http://sp.example.org"/>

    <!-- Attribute rule for the email attribute -->
    <AttributeRule attributeID="email">
        <!-- Permit value rule that releases any value. -->
        <PermitValueRule xsi:type="basic:ANY" />
    </AttributeRule>

</AttributeFilterPolicy>
```

Autre exemple de fichier attribute-filter.xml :
Accepter de donner tout attribut à un seul SP

Sécurité : la protection des échanges

- De préférence des connexions HTTPS sur tout le cheminement.
- Toutes les assertions SAML sont autosignées
  - Les clés publiques de chaque SP et IdP sont diffusées dans les métadonnées.
- Elles peuvent également être chiffrées

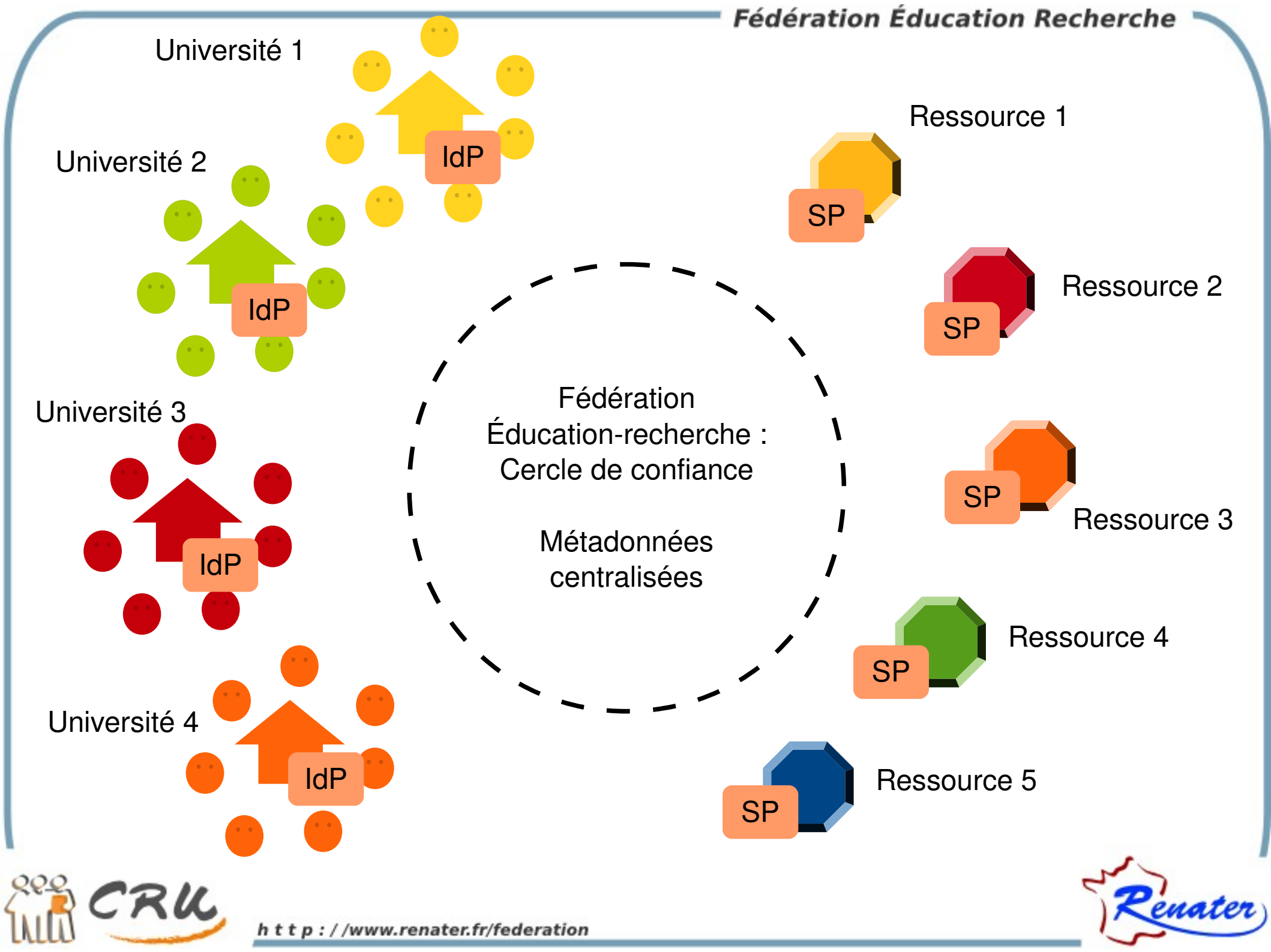
Métadonnées : Un SP

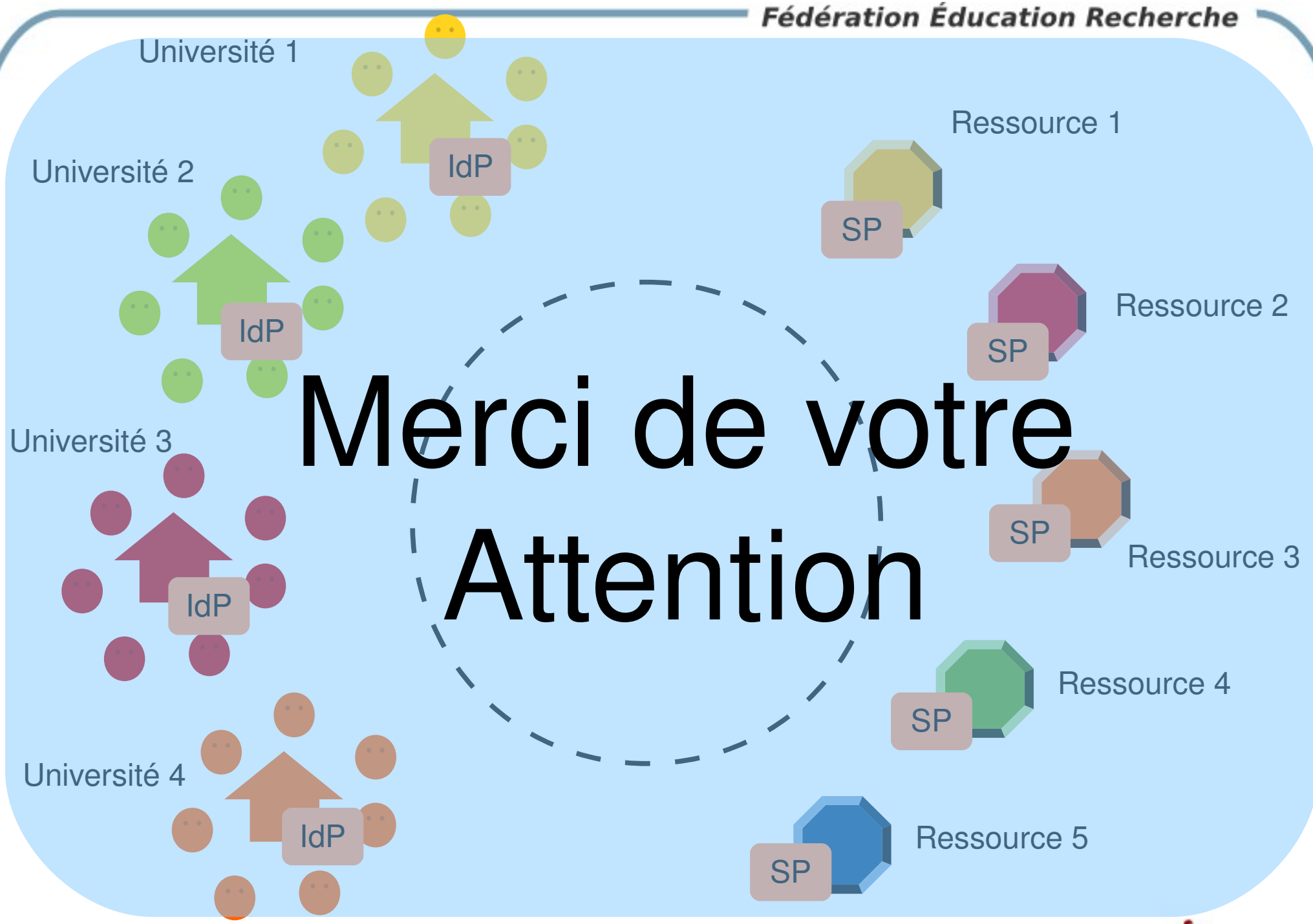
```
<EntityDescriptor entityID="https://virtualhost2.cru.fr">
    <SPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:1.1:protocol
urn:oasis:names:tc:SAML:1.0:protocol urn:oasis:names:tc:SAML:2.0:protocol">
        <KeyDescriptor use="signing">
            <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                <ds:X509Data>
                    <ds:X509Certificate>
                        [Certificat]
                    </ds:X509Certificate>
                </ds:X509Data>
            </ds:KeyInfo>
        </KeyDescriptor>
        <NameIDFormat>urn:mace:shibboleth:1.0:nameIdentifier</NameIDFormat>
            <AssertionConsumerService Binding="urn:oasis:names:tc:SAML:1.0:profiles:browser-post"
        Location="https://virtualhost2.cru.fr/Shibboleth.sso/SAML/POST" index="1"
isDefault="true"></AssertionConsumerService>
    </SPSSODescriptor>
    <ContactPerson contactType="technical">
        <SurName>Olivier Salaün</SurName>
        <EmailAddress>olivier.salaun@cru.fr</EmailAddress>
    </ContactPerson>
</EntityDescriptor>
```

Conclusion... un mot sur la confiance

- Comment ça « autosignées » ?
- Les SP et IdP signent leurs assertions avec leur propre certificat.
- Les clés publiques sont diffusées dans les métadonnées : donc on sait qui est qui.
- Les métadonnées sont distribuées dans un fichier central, lui-même signé.

=> La confiance repose avant tout sur un organisme, garant de la validité des métadonnées.


Université 1
IdP
Université 2
IdP
Université 3
IdP
Université 4
IdP
Fédération
Éducation-recherche :
Cercle de confiance
Métadonnées
centralisées
Ressource 1
SP
Ressource 2
SP
Ressource 3
SP
Ressource 4
SP
Ressource 5
SP


Université 1
IdP
Université 2
IdP
Université 3
IdP
Université 4
IdP
Ressource 1
SP
Ressource 2
SP
Ressource 3
SP
Ressource 4
SP
Ressource 5
SP
Merci de votre Attention