

Choix d'implémentation d'esup-ecm à Bordeaux 1

Exemple d'intégration des flux RSS Nuxeo dans Esup-Lecture



* www.u-bordeaux1.fr

↳ Le contexte

- **Un besoin de remplacement de l'intranet actuel :**

Le contenu est essentiellement constitué par des documents (word ou pdf) produits par les différentes instances ou composantes.

Quelques services ont cependant quelques pages HTML (informations à modifier souvent, actualités, ...).

Une consultation relativement faible : les documents ne sont pas forcément faciles à trouver (notamment sur des sujets transverses) et il n'y a pas de moyen de savoir lorsqu'un nouveau document est déposé.

Au final, les gens se plaignent régulièrement d'un manque d'informations alors que bien souvent celle-ci est disponible mais mal exposée.

↘ Le contexte

▪ Des gros besoins en terme de travail collaboratif :

Le besoin de partage de document est très important et en l'absence d'une solution réelle, chacun s'est organisé pour pallier à ce manque à sa façon. (utilisation du canal de stockage et de Slide dans l'ENT, utilisation des listes sympa ou de la messagerie à des fins de stockage, partages réseau (non accessibles depuis l'extérieur pour la plupart), demande du mot de passe à son collaborateur pour accéder aux documents sur son ordinateur (sic ...)).

Les usages ne sont pas identiques : il s'agit parfois de simplement partager de l'information à un instant donné entre quelques personnes, d'autres fois des outils de GED avec des notions de versions, de workflow et de référencement sont nécessaires.

↳ La découverte d'Esup-ECM

▪ LA solution ?

Début des tests sur Esup-ECM 1.0 (et ORI-OAI 1.5) en octobre. Fonctionnalités importantes (partage de documents, GED), services proposés et ergonomie vraiment intéressants. Peut répondre à la plupart de nos problématiques.

MAIS ...

Intégration au moins à ce stade pas forcément évidente dans l'ENT (quelques soucis avec l'authentification CAS, plugin de drag&drop non fonctionnel dans un canal iFrame ...).

Et surtout, quels services proposer avec quels points d'entrée et pour quels usages ?...

↳ Les solutions retenues

A partir de Nuxeo 5.3 (corrections de bugs + quelques fonctionnalités supplémentaires) + plugin Esup-ECM 1.1 (merci Raymond !).

Le choix a été fait d'offrir la plupart des fonctionnalités proposées de base dans Nuxeo mais en offrant des points d'entrées différents suivant les services et les usages.

L'idée est que le gens ne se retrouvent pas directement dans l'interface Esup-ECM sans savoir pourquoi ils sont là, en se demandant à quoi tout ça peut bien servir ...

Deux grands axes :

- la consultation des documents (intranet)
- la production et le partage de documents (travail collaboratif).

↳ Les solutions retenues

Pour la partie intranet, on a défini une arborescence pour la publication des documents. Des espaces de travail associés ont également été créés. Pour les services qui ont besoin de publier des informations dans un mode plus dynamique, on utilise WebEngine. Les sites sont accessibles depuis l'ENT et depuis le site institutionnel.

Les notifications permettront aux gens de se tenir informés s'ils le souhaitent des modifications / créations sur les espaces auxquels ils ont accès. (Dans certains cas, ces notifications pourront être activées, pour un groupe donné, d'autorité par l'administrateur de l'espace).

Enfin un certain nombre de flux RSS sur ces espaces de publication ont été intégrés dans esup-lecture pour pouvoir remonter dans le canal actualités de l'ENT.

■ Intégration des flux RSS d'Esup-ECM dans Esup-Lecture

Par défaut les flux RSS de nuxeo sont protégés par une authentification de type BASIC. Il faut passer par une authentification CAS soit en supprimant l'authentification BASIC dans la chaîne des authentifications disponibles (modification du fichier esup-login-contrib.xml), soit en modifiant le fichier authentication-contrib.xml (dans nuxeo-platform-web-common) afin d'utiliser l'authentification CAS pour les flux RSS

Exemple en utilisant le fichier esup-login-contrib.xml :

```
<extension target="org.nuxeo.ecm.platform.ui.web.auth.service.PluggableAuthenticationService"
  point="chain">
  <authenticationChain>
    <plugins>
      <!-- plugin>BASIC_AUTH</plugin -->
      <plugin>CAS2_AUTH</plugin>
      <plugin>ANONYMOUS_AUTH_FOR_CAS2</plugin>
    </plugins>
  </authenticationChain>
</extension>
```

■ Intégration des flux RSS d'Esup-ECM dans Esup-Lecture

Il faut également préciser le nom de la variable dans laquelle est stockée le ticket ainsi que l'URL de validation sur le serveur CAS. (toujours dans le fichier esup-login-contrib.xml)

```
<parameter name="proxyKey">ticket</parameter>
<parameter name="proxyValidateURL">
  https://"@sso-cas-url@/proxyValidate
</parameter>
```

Esup-Lecture doit pouvoir être utilisé en tant que proxy CAS (se reporter à la documentation disponible sur le site d'esup-portail :

<http://www.esup-portail.org/display/PROJESUPLECTURE/CAS+en+mode+portlet>
ainsi que <http://www.esup-portail.org/pages/viewpage.action?pageId=6946888>.)

Déclaration d'un nouveau contexte dans esup-lecture : (esup-lecture.xml) :

```
<context name = "Actualites" id = "nuxeo">
  <description><![CDATA[<h1>Actualités Esup-ECM</h1>]]></description>
  <refCategoryProfile refId="nuxeo1"/>
  <refCategoryProfile refId="nuxeo2"/>
  <refCategoryProfile refId="nuxeo3"/>
  <refCategoryProfile refId="nuxeo4"/>
  <refCategoryProfile refId="nuxeo5"/>
  <refCategoryProfile refId="nuxeo6"/>
</context>
```

■ Intégration des flux RSS d'Esup-ECM dans Esup-Lecture

Définition des catégories : (esup-lecture.xml)

```
<categoryProfile name="Documents institutionnels" id="nuxeo1" urlCategory="file:///home/esup-lecture/nuxeo1.xml"
trustCategory="yes" access="public" ttl="3600" timeout="10000" >
  </categoryProfile>
  <categoryProfile name="Services et composantes" id="nuxeo2" urlCategory="file:///home//esup-
lecture/nuxeo2.xml" trustCategory="yes" access="public" ttl="3600" timeout="10000" >
  </categoryProfile>
  <categoryProfile name="Enseignement et pédagogie" id="nuxeo3" urlCategory="file:///home/esup-
lecture/nuxeo3.xml" trustCategory="yes" access="public" ttl="3600" timeout="10000" >
  </categoryProfile>
<categoryProfile name="Recherche" id="nuxeo4" urlCategory="file:///home/esup-lecture/nuxeo4.xml"
trustCategory="yes" access="public" ttl="3600" timeout="10000" >.
  </categoryProfile>
  <categoryProfile name="Informations pratiques" id="nuxeo5" urlCategory="file:///home/esup-
lecture/nuxeo5.xml" trustCategory="yes" access="public" ttl="3600" timeout="10000" >
  </categoryProfile>
  <categoryProfile name="Usages des services numériques" id="nuxeo6" urlCategory="file:///home/esup-
lecture/nuxeo6.xml" trustCategory="yes" access="public" ttl="3600" timeout="10000" >
  </categoryProfile>
```

■ Intégration des flux RSS d'Esup-ECM dans Esup-Lecture

Puis enfin définition des sources (exemple du fichier nuxeo6.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE category SYSTEM "/home/esup/webapps-2.0.5/esup-lecture/WEB-
INF/classes/properties/category.dtd">
<category edit="all" name="Services num&#233;riques">
<description>Documentations sur les services num&#233;riques de l'Universit&#233;
</description>

<sourceProfiles>

    <sourceProfile id="nuxeo1f" access="cas" name="JeFYCO" specificUserContent="no"
url="http://tramontane.drimm.u-bordeaux1.fr/nuxeo/getSyndicationDocument.faces?docRef=default/3a362504-8aaa-
4229-93f1-47a9065e965b">
        <visibility>
        <allowed>
        <group name="local.0"/>
        </allowed>
        <autoSubscribed>
            <group name="local.0"/>
        </autoSubscribed>
        <obliged />
        </visibility>

    </sourceProfile>
...
</sourceProfiles>

</category>
```

➤ Intégration des flux RSS d'Esup-ECM dans Esup-Lecture

Et ça ne marche pas !... Erreur : « no service name in the URL »
Effectivement, dans la classe `Cas2Authenticator.java` (`addons/nuxeo-platform-login/nuxeo-platform-login-cas2`)
on a :

```
protected String checkProxyCasTicket(String ticket,
HttpServletRequest httpRequest) {
    // Get the service passed by the portlet
    String service = httpRequest.getParameter(serviceKey);
    if (service == null) {
        log.error("checkProxyCasTicket: no service name in the URL");
        return null;
    } ....
```

On a donc modifié la classe comme suit :

```
protected String checkProxyCasTicket(String ticket,
HttpServletRequest httpRequest) {
    String service = httpRequest.getParameter(serviceKey);
    if (service == null) {
        service = getAppURL(httpRequest);
    }
    ...
```

↘ Premières impressions ...

De gros espoirs sur les possibilités de l'outil :

- Encore beaucoup de nombreuses possibilités à exploiter (SharePoint, CMIS, implémentation portlet du moteur de recherche, intégration du tableau de bord...).
- De bonnes réactions sur les première utilisations de quelques « cobayes » : le drag&drop et l'extension LiveEdit sont très appréciés. Certains sont même déjà prêt à abandonner les modes de stockage classique (disque dur monté ou local) pour dès à présent tout mettre dans Esup-ECM !

Mais toujours des doutes et interrogations :

- Comportement de l'application avec la montée en charge : temps de réponse déjà longs sur certains actions sans même être en production ... Dimensionnement à étudier et montée en compétences nécessaire au fur et à mesure des usages.

↘ Premières impressions ...

Besoin de formation et d'accompagnement à l'outil pour les utilisateurs, notamment ceux qui vont être amenés à produire des documents qui vont devoir ensuite être publiés. C'est souvent une façon de travailler à laquelle les gens ne sont pas habitués.

Doutes sur nos choix, en terme d'arborescence (le moins grave), d'autonomie laissé à l'utilisateur (le plus risqué ?)...

Encore des questions non tranchées, notamment sur l'intégration d'ORI-OAI (onglet supplémentaire, nouvelle source de confusions possible) dans cet environnement ou la mise en place d'un nuxeo « spécial ORI »

Déjà des besoins exprimés sur des possibilités de travail collaboratif avec des gens extérieurs à l'Université.

Un outil qui met encore plus en évidence la nécessité d'un véritable gestionnaire de groupe(r)s ...

Petite démo