

Le CMS LIFERAY

Claude Duvallet

Université du Havre
UFR Sciences et Techniques
25 rue Philippe Lebon - BP 540
76058 LE HAVRE CEDEX
Claude.Duvallet@gmail.com
<http://litis.univ-lehavre.fr/~duvallet/>

Présentation de Liferay (1/3)

- ▶ Liferay est un portail J2EE open source compatible avec le standard JSR-168.
- ▶ Une de ses particularités est de pouvoir s'exécuter sur la majorité des serveurs d'applications et de bases de données disponibles actuellement.
- ▶ La société Liferay a été fondée en 2000 dans le but de développer un portail abordable, proposant une alternative aux solutions commerciales souvent très chères.
- ▶ Liferay Portal est distribué sous licence MIT, ce qui permet de pouvoir le modifier et le redistribuer librement.
- ▶ Une offre de service est fournie directement par Liferay et par des partenaires.

Le portail Liferay et le moteur de Workflow JBPM

- 1 Les portlets
- 2 Liferay : personnalisation
- 3 Les thèmes de présentation
- 4 La gestion des droits
- 5 Le moteur de workflow jBPM

Présentation de Liferay (2/3)

- ▶ Adhérent strictement aux standards Java et J2EE, Liferay peut s'intégrer dans des environnements multiples :
 - Il peut s'exécuter sur la quasi-totalité des serveurs d'applications Java actuels et se connecter à un très grand nombre de serveurs de base de données.
 - Respectant également le standard JSR-168, il est possible d'y déployer un grand nombre de portlets.
- ▶ Actuellement la dernière version utilisée est la version 6.
 - Liferay est encore aujourd'hui en développement.
 - De nouvelles fonctionnalités sont ajoutées régulièrement.
- ▶ Liferay est un portail d'intégration permettant aux utilisateurs d'accéder à des pages regroupant divers contenus et outils.

Présentation de Liferay (3/3)

- ▶ Liferay respectant le standard JSR-168, cela lui offre la faculté d'intégrer un grand nombre d'extensions.
- ▶ Développé en suivant les standards Java et J2EE, Liferay peut être déployé sur des architectures très variées (serveur d'applications et de base de données) et peut également s'intégrer à un annuaire LDAP.
- ▶ Rapide à installer et possédant un nombre de portlets conséquents, il offre une solution simple et rapide pour déployer un portail d'entreprise tout en permettant de le faire évoluer avec des extensions développées spécifiquement ou disponibles dans le commerce.
- ▶ Il pourra donc intéresser aussi bien ceux qui nécessitent des outils standards prêts à fonctionner que ceux qui ont des besoins plus spécifiques et souhaitent bâtir de nouveaux portlets pour les intégrer dans Liferay.

Principe de base (1/2)

- ▶ Un portail a pour rôles :
 - d'intégrer plusieurs applications distinctes (portlets) par page,
 - d'offrir une gestion des droits d'accès à chacune de ces applications :
 - en vue (standard),
 - et en configuration.
 - de permettre indirectement un découplage entre le template de page, et les applications qu'elle contient.
- ▶ Liferay est aussi un CMS :
 - gestion de contenus WEB,
 - gestion des images et documents attachés à ces contenus,
 - gestion des espaces de publication,
 - gestion des droits liés au workflow standard de publication (contribution/approbation).

Forces et faiblesses de Liferay

- ▶ Les forces :
 - Il possède de nombreux environnements de déploiement.
 - Il est conforme au standard JSR 168.
 - Il permet un agencement des portlets très facile.
 - Il offre de nombreuses possibilités d'extension.
 - Il repose sur le développement de portlets standards.
 - Il possède la vitalité des projets Open Source.
- ▶ Les faiblesses :
 - La documentation de certaines fonctionnalités est manquante.
 - La concurrence d'autres projets Open Source est importante.
 - Il possède une communauté restreinte de développeurs.

Principe de base (2/2)

- ▶ Liferay traite de ces points avec les notions suivantes :
 - Des espaces et une gestion de droits commune entre la gestion de contenus web et le portail
 - Un espace = une communauté (ou une organisation),
 - Contient des pages (correspond à la notion de sous-site), et par conséquent les contenus web et les propriétés des portlets qui les constituent.
 - Une gestion de droits délégués, exploitant la notion de rôle. Chaque rôle peut être appliqué à une communauté, ou bien à l'ensemble du portail.
 - Administrateur,
 - Contributeur,
 - Validateur.
 - Une séparation entre :
 - La présentation et la navigation.
 - Le modèle de disposition des portlets ("colonage").
 - Les "gabarits" de la gestion de contenus.
 - Les portlets.

Le vocabulaire de Liferay

- ▶ Contenu web = Article
- ▶ Type de contenu = Structure
- ▶ "gabarit" de contenu = template
- ▶ Application = portlet
- ▶ Page = Layout
- ▶ Présentation (charte graphique) = Theme
- ▶ "colonage" = modèle de disposition = Layout template
- ▶ Action possible sur un objet (Permissions) :
 - Applicable aux portlets, contenus, sujets de forum...
- ▶ Profil thématique = Rome
 - sert à établir les permissions autorisées pour ce profil
- ▶ Espace de publication (et/ou sous-site) :
 - groupe de travail = Communauté
 - organisationnel = Organisation

Intégration de Liferay (1/2)

- ▶ Liferay supporte tous les serveurs d'application suivant : Borland ES , Geronimo, GlassFish, JBoss, JOnAS, JRun, OracleAS, Orion, Pramati, REXIP, Sun JSAS, WebLogic, WebSphere.
- ▶ Grâce à l'utilisation en interne d'Hibernate pour la persistance des données, tous ses serveurs peuvent être combinés avec les bases de données suivantes : DB2, Hypersonic, Interbase, MySQL, Oracle, PostgreSQL, SQLServer.
- ▶ Cette capacité d'intégration est véritablement un point fort de Liferay.
- ▶ Il offre une très grande liberté de choix et peut permettre de mettre à profit des ressources déjà existantes.
- ▶ L'authentification des utilisateurs peut se faire grâce à une connexion à un annuaire LDAP.
- ▶ En outre, il est possible de déployer les portlets compatibles avec la JSR-168 et des portlets WSRP.

Installation et configuration de Liferay

- ▶ L'installation est très facile.
- ▶ Liferay nous offre la possibilité de télécharger divers bundles incluant des serveurs d'application (Geronimo, JBoss, ou JOnAS) et conteneurs de servlet (Jetty, Resin ou Tomcat).
- ▶ Pour un déploiement sur un autre serveur, il suffit de le télécharger sous forme de fichier WAR ou EAR.
- ▶ La documentation de l'installation et de la configuration est bien fournie.
- ▶ Il est particulièrement facile de configurer la base de données que l'on souhaite utiliser grâce à des scripts de création de bases téléchargeables pour un grand nombre de SGBD.
- ▶ Par défaut, Liferay utilise une base HSQLDB ce qui permet d'avoir un portail fonctionnel sans aucune installation de base de donnée.

Intégration de Liferay (2/2)

- ▶ Des portlets utilisant JSF, Struts, Spring peuvent également être développés.
- ▶ Un portlet RSS est fourni, permettant de lire des flux RSS provenant de sites externes.
- ▶ L'intégration d'autres application du Système d'Information peut se faire de plusieurs manières :
 - Un portlet WebProxy (PortletBridge) est déployable et permet d'accéder à une application distante à l'intérieur d'un portlet.
 - Une authentification basique est possible avec cette portlet et elle permet également d'utiliser une feuille de style XSLT pour modifier la page distante rapatriée.
 - Si l'application expose des WebServices SOAP, un portlet exploitant ces services peut être développé.
- ▶ Par contre, contrairement aux portails commerciaux, aucun connecteur vers des progiciels ou des outils de reporting n'est fourni.

Extensibilité de Liferay (1/2)

- ▶ De nombreux portlets sont déjà intégrés à Liferay et permettent de répondre à certains besoins.
- ▶ Néanmoins, l'intégration ou le développement de nouveaux portlets permet d'utiliser Liferay comme un framework de portail.
- ▶ Il offre ainsi des services de bases très poussés (Gestion des utilisateurs, des permissions, de l'authentification,...).

Le rendu de Liferay

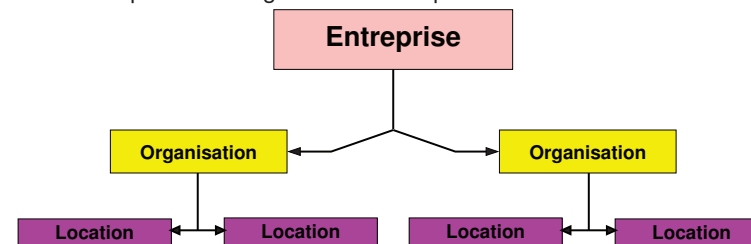
- ▶ Liferay est fourni avec plusieurs thèmes préinstallés et beaucoup d'autres réalisés par la communauté d'utilisateurs sont téléchargeables sur le site.
- ▶ Deux possibilités sont offertes pour créer ses propres thèmes : les développer à l'aide de pages JSP ou bien en utilisant le moteur Velocity.
- ▶ Les thèmes construits avec Velocity sont déployable « à chaud », sans redémarrage du serveur.
- ▶ Dans les 2 cas, il s'agit de fournir un modèle pour les différents éléments de la page (en-tête, pied-de-page, portlet,...), d'ajouter les images qui sont utilisées pour construire le site et la feuille de style qui sera appliquée.
- ▶ Un fichier xml accompagnera le thème permettant de définir le nom sous lequel apparaîtra le thème dans l'interface de choix.

Extensibilité de Liferay (2/2)

- ▶ Les portlets développés seront fournis dans une archive .war comprenant un fichier `portlet.xml` décrivant les propriétés de déploiement (standard JSR-168).
- ▶ Éventuellement l'ajout des fichiers `liferay-display.xml` et `liferay-portlet.xml` permettront de configurer des paramètres par défaut propres à Liferay (exemple : droits par défaut sur le portlet,...).
- ▶ Ces paramètres spécifiques à Liferay ne sont pas un frein à l'interopérabilité car :
 - ils concernent uniquement des fonctionnalités propres à Liferay,
 - ils permettent simplement de faire gagner du temps à l'administrateur lors du déploiement et de l'affectation des droits aux portails.

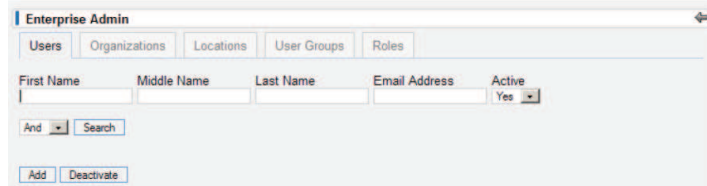
Administration de Liferay (1/2)

- ▶ Administration des utilisateurs
 - Liferay nous propose une administration des utilisateurs basée sur 3 niveaux hiérarchiques : Entreprise, Organisation et Location.
 - Entreprise est le niveau le plus haut dans lequel on peut créer des Organisations dans lesquelles on peut créer des Locations.
 - Ce principe permet de définir des droits d'administration sur ces différents niveaux (on peut avoir des administrateurs s'occupant uniquement d'une Organisation ou d'une Location) et permet de s'adapter à des organisations complexes.



Administration de Liferay (2/2)

- ▶ Parallèlement à cela, des Groupes peuvent être créés permettant de regrouper des utilisateurs provenant d'Organisations ou de Locations différentes.
- ▶ Voici un aperçu du portlet d'administration du niveau Enterprise :

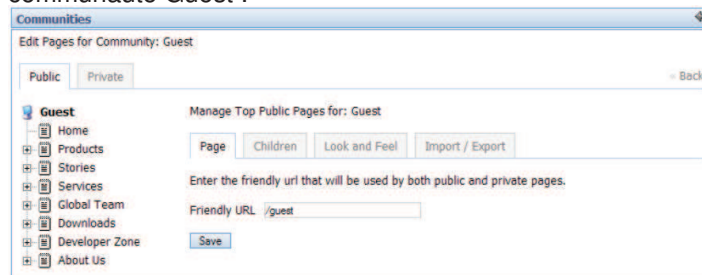


Personnalisation de Liferay

- ▶ La personnalisation des pages du portail s'effectue en 2 étapes :
 - la création des pages,
 - la personnalisation de celles-ci.
- ▶ Des thèmes peuvent ensuite être appliqués aux Communautés ou individuellement aux pages.
- ▶ La personnalisation des pages propres à une communauté ou propre à l'utilisateur est similaire.
- ▶ L'administrateur modifie les pages des communautés de la même manière qu'un utilisateur modifie sa page personnelle (ou les pages sur lesquelles il dispose de droits de modification).

Personnalisation de Liferay : Création des pages

- ▶ Tout d'abord, on crée une hiérarchie de pages qui sont associés soit directement à un utilisateur, soit à une communauté.
- ▶ Voilà par exemple, l'ensemble des pages associées à la communauté Guest :

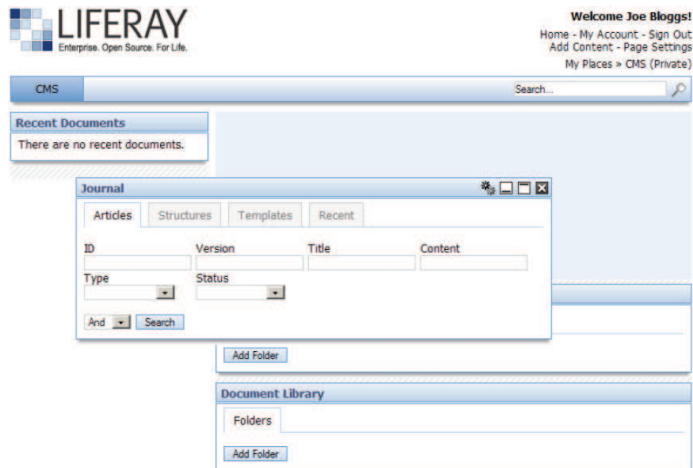


- ▶ Cette hiérarchie pourra être parcourue très facilement grâce à l'ajout sur les pages du portlet de navigation qui permet d'obtenir des liens vers les pages enfants.

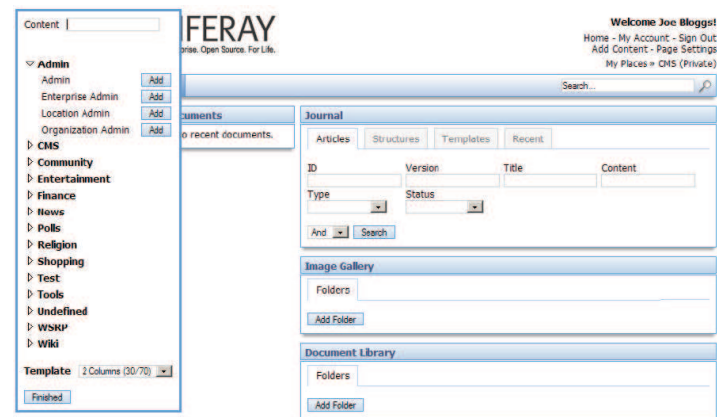
Personnalisation des pages de Liferay

- ▶ La personnalisation des pages s'effectue directement depuis la page en question.
- ▶ En effet, en disposant des droits nécessaires, on peut :
 - modifier l'arrangement de la page en déplaçant les portlets à l'intérieur de celle-ci avec un simple drag & drop,
 - ajouter des portlets en faisant apparaître la liste des portlets disponibles grâce à l'outil « Add Content ».

Exemple : Déplacement du portlet Journal



Exemple : Ajout de nouveaux portlets



Personnalisation de Liferay : Les thèmes

► Les thèmes :

- Le rendu des pages peut être modifié grâce aux thèmes.
- Ces thèmes peuvent être définis pour l'ensemble des pages d'une communauté ou pour des pages individuellement.
- Liferay est fourni avec quelques thèmes préinstallés et beaucoup d'autres disponibles en téléchargement sous forme de fichiers WAR à déployer sur le serveur.
- Cette méthode d'installation est très pratique et permet de déployer un nouveau thème en quelques secondes, sans avoir besoin de redémarrer le serveur.
- L'écriture de thèmes personnalisés semble aisée.
- L'interface de Liferay est construite sur Tiles (Struts), système de templates très répandu.
- Il suffit de créer ou modifier quelques fichiers .JSP ou Velocity et CSS correspondant aux différentes zones du portail, d'ajouter les images utilisées dans ces zones, de packager le tout dans un fichier WAR et le tour est joué.

Le CMS de Liferay : Journal (1/7)

- Un système basique de gestion de contenu (CMS) appelé Journal est présent dans Liferay.
- Loin des vrais systèmes de CMS, il permet simplement de créer des articles et de les publier sur des pages.
- On peut ainsi réaliser simplement un petit site statique.
- Pour gérer les articles, on utilise des Structures.
- Elles permettent de définir quels éléments composeront l'article (par exemple, un texte représentant le titre, une image et un contenu HTML qui représentera le corps de l'article).

Le CMS de Liferay : Journal (2/7)

- ▶ Exemple de structure utilisée pour gérer les articles :

Le CMS de Liferay : Journal (3/7)

- ▶ Ensuite, pour chaque structure créée, on y ajoute un ou plusieurs Templates.
- ▶ Ils servent à définir l'agencement des différents éléments composant l'article.

Le CMS de Liferay : Journal (4/7)

- ▶ Les templates peuvent être définis avec une feuille de style XSL ou en langage VTL (Velocity Template Language).

Editor Type: HTML

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="html" omit-xml-declaration="yes"/>
  <xsl:template match="/">
    <img>
      <xsl:attribute name="src">
        <xsl:value-of disable-output-escaping="yes"
          select="root/dynamic-element[&#92;name='image']/dynamic-content"/>
      </xsl:attribute>
    </img>
    <h2><xsl:value-of disable-output-escaping="yes"
      select="root/dynamic-element[&#92;name='titre']/dynamic-content"/>
    </h2>
    <xsl:value-of disable-output-escaping="yes"
      select="root/dynamic-element[&#92;name='contenu']/dynamic-content"/>
  </xsl:template>
</xsl:stylesheet>
```

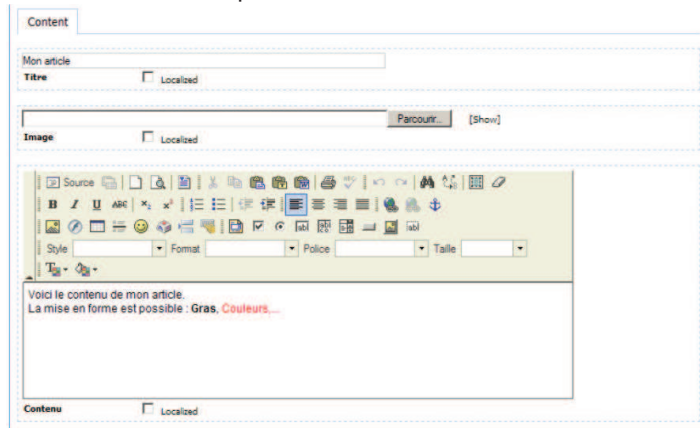
Update Select And Copy Cancel

Le CMS de Liferay : Journal (5/7)

- ▶ On peut ensuite rédiger un ou plusieurs articles s'appuyant sur la structure et le template définis.
- ▶ On définit tout d'abord les propriétés générales de l'article : nom, catégorie, date de publication, d'expiration, ...

Le CMS de Liferay : Journal (6/7)

- ▶ Puis, on peut ajouter le contenu en s'appuyant sur le template et la structure définies précédemment.



Le CMS de Liferay : Journal (7/7)

- ▶ Pour terminer, il suffit d'ajouter un portlet Journal Content sur la page où l'on souhaite afficher l'article pour qu'il apparaisse.
- ▶ Il est à noter que les articles peuvent être localisés et peuvent donc s'adapter à la langue choisie par l'utilisateur.
- ▶ La page d'accueil de Liferay (<http://www.liferay.com>) est un bon exemple de l'utilisation de Journal.

Intégration avec le CMS Alfresco

- ▶ L'outil Alfresco peut être intégré au portail Liferay afin de remplir différentes fonctions dans le cadre de la gestion électronique de document.
- ▶ Un portlet client est disponible dans Liferay permettant de se connecter à un serveur Alfresco.

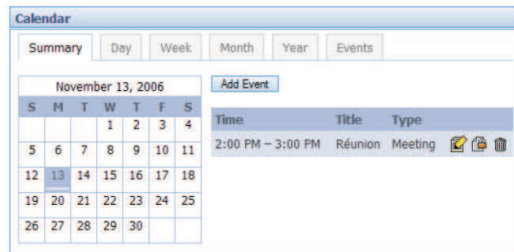
Quelques outils collaboratifs (1/5)

- ▶ Liferay propose une série de petits outils collaboratifs sous forme de Portlets.
- ▶ Les fonctionnalités sont basiques mais les outils sont fonctionnels.
- ▶ On trouve :
 - Favoris : Ce portlet permet de stocker des favoris et de les organiser par dossier.
 - Mail
 - Le portlet Mail permet de se connecter à un serveur IMAP pour lire et envoyer des messages.
 - Toutefois, ce portlet n'autorise que la connexion à un seul serveur pour tous les utilisateurs.

Quelques outils collaboratifs (2/5)

► Calendrier

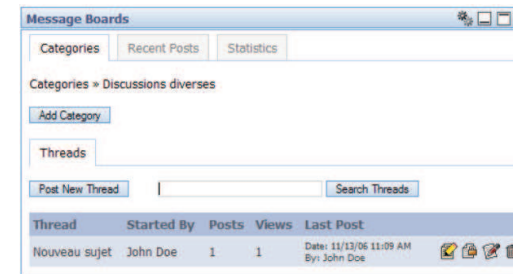
- Le calendrier permet de publier des événements et de les partager éventuellement entre les membres d'une communauté.
- Chaque événement peut faire l'objet de permissions particulière (aucun accès, lecture, lecture/écriture).



Quelques outils collaboratifs (3/5)

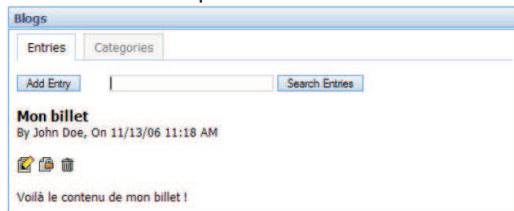
► Forum

- Le forum permet de créer des fils de discussion organisable par catégorie. Il est
- possible de poster également des pièces jointes aux messages.

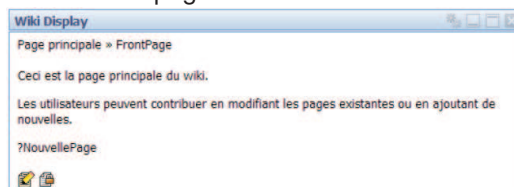


Quelques outils collaboratifs (4/5)

- Blog : Un blog est disponible et permet aux membres d'une communauté de poster des billets.



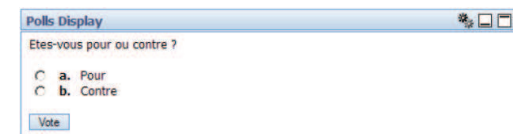
- Wiki : Le portlet wiki permet à tous les utilisateurs de créer ou modifier des pages.



Quelques outils collaboratifs (5/5)

► Sondages

- Un portlet Poll permet de créer des sondages.
- Ceux-ci pourront être visible pour quelques utilisateurs seulement, toute un communauté ou même par les visiteurs du site.



Business intelligence

- ▶ Depuis 2007, Liferay propose un Portlet « Liferay reporting Portlet ».
- ▶ Un système de logging (Log4j) est présent.
- ▶ Il est configurable à l'aide d'un simple fichier xml.
- ▶ Le portlet Admin permet également de modifier les niveaux de déclenchement des logs.

Perspectives (2/2)

- ▶ Avec un portail fonctionnel et utilisable dès l'installation mais possédant une base solide pour l'intégration et le développement de nouveaux portlets, Liferay offre un très bon outil capable de se déployer rapidement et de s'adapter à des besoins complexes.
- ▶ Liferay est sans aucun doute le mieux fini des portails Open Source mais il lui manque encore des connecteurs vers les applications du SI et les outils de reporting pour faire face à ses concurrents propriétaires.
- ▶ Même si depuis la sortie de la version 6, la communauté semble s'accroître, elle n'est pas encore très importante.
- ▶ Elle est malgré tout très active notamment grâce à l'appui de la société Liferay.

Perspectives (1/2)

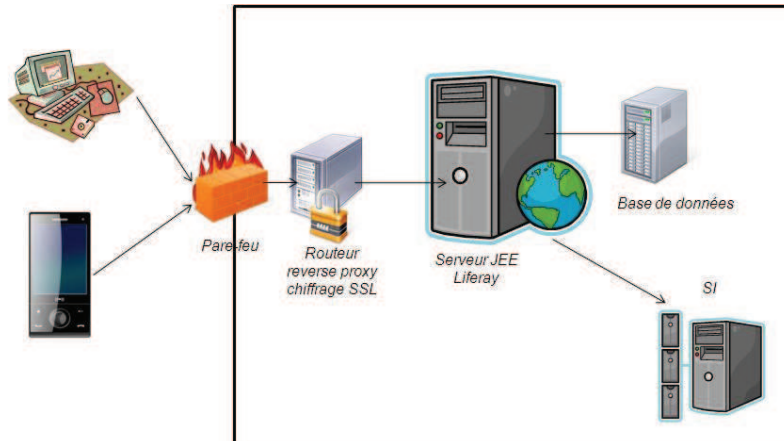
- ▶ Après 10 années d'existence, Liferay nous offre un produit très abouti.
- ▶ Les fonctionnalités de base (gestion des utilisateurs, déploiement de nouveaux portlets,...) sont solides et bien conçues.
- ▶ L'aspect visuel est très réussi et les portlets fournis sont nombreux et fonctionnels.
- ▶ Les pistes de développement actuelles vont vers encore plus de fonctionnalités offertes dans le package de base.
- ▶ Si l'installation et la configuration générale sont bien documentées, en revanche la documentation de certains portlets est légère et peut conduire à des difficultés à configurer certaines fonctionnalités.

Administration en ligne du portail

- ▶ Contrairement à d'autres solutions portail (particulièrement open source), Liferay offre l'accès à une interface graphique d'administration : le panneau de commande ("control panel")
 - Compte personnel :
 - Public type : Utilisateur enregistré
 - Contenus et outils communautaires
 - Public type : contributeurs, gestionnaires
 - Portail : configurations et référentiels
 - Public type : webmasters, responsables métier
 - Serveur : gestion des ressources et applications fournies
 - Public type : exploitant
- ▶ La plupart des paramètres non administrables sont inclus dans un fichier unique.

Présentation de la plateforme logicielle (1/3)

L'architecture générale



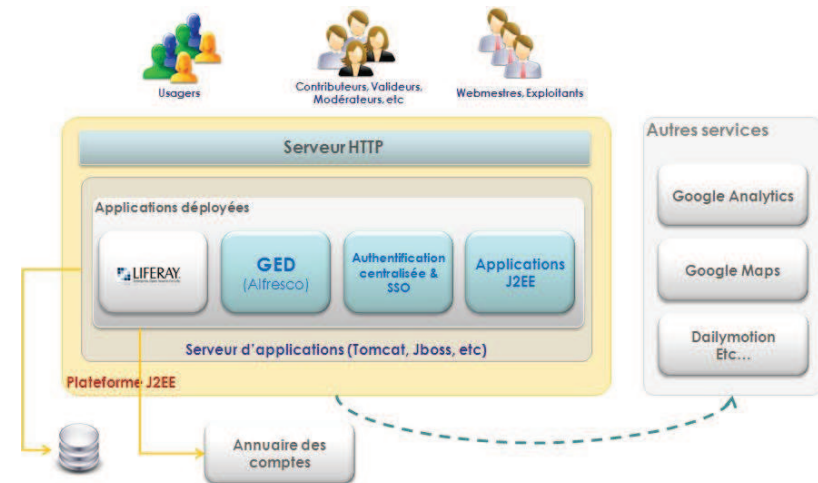
Présentation de la plateforme logicielle (3/3)

► Installation : emplacement des sous-composants.

- Un répertoire de travail : Data/
 - Fichiers de la Bibliothèque de documents (selon paramétrage),
 - Index Lucene (selon paramétrage),
 - hsql (selon paramétrage).
- Lib/ext : librairies nécessaires au fonctionnement du portail.
 - Répertoire de déploiement à chaud des plug-ins : deploy,
 - Répertoire de déploiement des webapps du serveur d'application : le portail (ROOT.war).
- Les plug-ins installés.

Présentation de la plateforme logicielle (2/3)

L'architecture d'intégration de Liferay



Installation du portail (1/2)

- Java
 - Procédure d'installation standard.
 - V1.5 ou 1.6
- Bundle Liferay + Tomcat
 - Il suffit de décompresser !
- La base de données HSQL :
 - Liferay embarque une base HSQL (hypersonic)
 - Les données sont stockées dans /data/hsql/lportal.script
 - Les requêtes de la session en cours sont stockées dans lportal.log
 - Pour un usage de type démo, voire développement uniquement.

Installation du portail (2/2)

- ▶ Faîtes de régulières sauvegardes de votre répertoire "data"
 - Cela revient à sauvegarder un "dump" de la base de données + des documents
- ▶ Pour démarrer :
 - Aller dans le répertoire (tomcat)/bin
 - Exécuter "startup.bat"
 - Une fenêtre doit s'ouvrir, avec la sortie standard de Tomcat.
- ▶ Pour arrêter :
 - Aller dans le répertoire (tomcat)/bin
 - Exécuter "shutdown.bat"

Vue d'ensemble sur les portlets

- ▶ Qu'est-ce que JSR-168 ?
- ▶ Qu'est-ce qu'un Portlet ?
- ▶ Cycle de vie d'un Portlet
- ▶ Avantages des portlets Standard Java
- ▶ Qu'est-ce que JSR-286 ?
- ▶ Modes d'un Portlet
- ▶ États d'écran
- ▶ Préférences
- ▶ Actions et événements
- ▶ Communication Inter-Portlets (IPC)

Après l'installation du portail

- ▶ Le fichier `liferay-display.xml` recense les portlets d'origine de Liferay, qui peuvent être ajoutées à une page du portail :
 - Dans (tomcat)/webapps/ROOT/WEB-INF/.
 - Ensemble des portlets disponibles depuis le menu contextuel, "Ajouter une application".
- ▶ Dans ce fichier :
 - Dans la catégorie "category.cms" (au début), rajouter la ligne suivante `<portlet id="15" />`
- ▶ Cela nous permettra d'ajouter l'administration des contenus web (id=15), dans des pages à accès restreint.

Qu'est-ce que JSR 168 ?

- ▶ JSR-168 est la spécification 1.0 des Portlets ?
- ▶ Créée suite au besoin d'avoir une spécification pour l'affichage multiple de plusieurs applications sur une même page
- ▶ JSR-168 définit le cycle de vie d'un Portlet ainsi que ses caractéristiques

Qu'est-ce qu'un Portlet ?

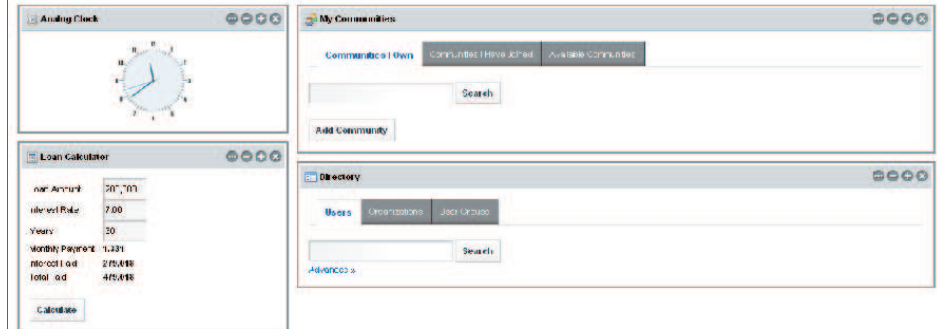
- ▶ Une Portlet est un composant web qui émet des requêtes et génère du contenu dynamique
- ▶ Le contenu généré par un Portlet est aussi appelé "Fragment" (i.e. HTML, XHTML, WML).
- ▶ Il peut être assemblé avec d'autres contenus pour former un document complet

Qu'est-ce qu'une application de type Portlet ?

- ▶ Le serveur d'applications considère une application portlet comme étant une autre application web qui « vit » au sein d'une archive web (.WAR)
- ▶ Outre le fichier descripteur de déploiement standard `web.xml`, les applications Portlet requièrent un descripteur de déploiement additionnel appelé `portlet.xml`
- ▶ Ce descripteur de déploiement informe le conteneur de portlets (Liferay) sur :
 - Les portlets incluses dans l'application
 - Rôles supportés
 - Modes de portlet supportés
 - Etc.

Exemple de portlets

Les portlets sont des fragments de pages WEB :



Phases d'un Portlet (1/2)

- ▶ Les portlets sont différentes des Servlets du moment où elles ont plus qu'un type de requête : Phases.
- ▶ Servlet : la méthode `service()` traite toutes les requêtes.
- ▶ JSR-168 (Portlet 1.0) possède deux phases :
 - Phase de présentation (Render Phase) : exécutée à chaque fois que la portlet a besoin d'être redessinée
 - Phase d'action (Action Phase) : appelée comme résultat d'une ActionURL.
⇒ Permet à la portlet d'effectuer du traitement spécifique pour changer d'état, visible quand la portlet est redessinée.

Phases d'un Portlet (2/2)

- ▶ JSR-286 (Portlet 2.0) ajoute deux nouvelles phases :
 - Phase d'évènement (Event Phase) : appelée quand un évènement est sur le point d'être lancé.
⇒ Les évènements (Events) peuvent être lancés dans la phase d'action (Action Phase) de la portlet et traités durant la phase d'évènement (Event Phase).
 - Resource Serving Phase : appelée par la méthode `serveResource()`.

Avantage des Portlets Standard Java

- ▶ Elles fournissent une architecture de base.
- ▶ L'architecture de base est la même à travers les Portails respectant le Standard Java.
- ▶ Les portlets développées selon ce standard peuvent être exécutées sur tous les conteneur de Portlet respectant ce standard.
- ▶ Documentation (Blue Sunrise).
- ▶ La spécification des Portlets ne va pas changer et vous pouvez de façon fiable vous y référer en cas de besoin d'information.
- ▶ API standard pour les développeurs.
- ▶ Les développeurs peuvent développer un ensemble de portlets standardisées pouvant être déployées sous n'importe quel Portail standardisé.
 - Attention, standardisé signifie : aucun `import com.liferay...` dans les classes et `jsp`.

Cycle de vie d'un Portlet

- ▶ La méthode `init()` : elle initialise la Portlet.
- ▶ La méthode `processAction()` : elle est appelée quand l'utilisateur lance une action.
- ▶ La méthode `render()` : elle dessine le contenu.
- ▶ La méthode `destroy()` : elle libère l'objet Portlet (Garbage collection).

Portlet 2.0 (JSR-286)

- ▶ Démarrage au mois de Novembre 2005 (3 ans après la version 1.0).
- ▶ Menée par IBM et révisée par les principaux vendeurs de solutions Portail (Liferay inclus).
- ▶ Approuvée et publiée au mois le 12 Juin 2008.
- ▶ Conserve la compatibilité avec les portlets 1.0 (JSR-168).
- ▶ Ajoute de nouvelles fonctionnalités en se basant sur l'expérience des différents vendeurs de solutions Portail (dont Liferay).

Les nouveautés

- ▶ Communication Inter-Portlets standardisée :
 - Paramètres publics.
 - Évènements.
- ▶ Resource serving :
 - Support d'AJAX.
 - Support des données binaires.
- ▶ Filtres pour Portlets.
- ▶ Meilleur support des Frameworks Web :
 - Les développeurs ne veulent pas forcément utiliser l'API Portlet directement (Notion de ponts ou Bridges).
- ▶ Des correctifs mineurs et des nouveautés.

Paramètres public ou Public Render (2/3)

- ▶ Les portlets doivent déclarer quel paramètre public elles vont utiliser en utilisant l'élément `supported-public-render-parameter`.
- ▶ Ces paramètres seront disponibles à travers le cycle de vie de la portlet (`processAction`, `processEvent`, `render`, `serveResource`).

```
<portlet>
  <portlet-name>portletA</portlet-name>
  . . .
  <supported-render-public-parameter>
    foo
  </supported-render-public-parameter>
</portlet>
```

- ▶ Un portlet peut lire un paramètre public en utilisant :

```
request.getPublicParameterMap()
```

Paramètres public ou Public Render (1/3)

- ▶ La méthode la plus simple pour une communication Inter-Portlets (IPC).
- ▶ Un développeur peut déclarer plusieurs paramètres publics pour une application Portlet dans le fichier `portlet.xml`.

```
<portlet-app>
  <public-render-parameter>
    <identifiant>foo</identifiant>
    <qname xmlns:x="http://foo.com/p">x:foo2</qname>
  </public-render-parameter>
  . . .
</portlet-app>
```

- ▶ Ne peuvent accéder à ces paramètres que les portlets au sein de l'application.

Paramètres public ou Public Render (3/3)

- ▶ Les paramètres public sont fusionnés avec les paramètres réguliers.
- ▶ Ils peuvent donc être lus en utilisant `getParameter(name)` et `getParameterMap()`.
- ▶ Un portlet peut effacer un paramètre public en invoquant :

```
Response.removePublicRenderParameter(name)
portletURL.removePublicRenderParameter(name)
```

Gestion des événements (1/3)

- ▶ Méthode puissante et fortement découplée pour la communication inter-portlets (IPC)
- ▶ Utilise un Producer-Listener pattern
 - Un portlet génère un événement
 - Zéro ou plusieurs portlets sont en écoute et agissent en fonction de cet événement
- ▶ Permet la communication entre portlet dans différentes applications
- ▶ Optionnellement, le conteneur peut aussi générer ses propres événements
 - Pas d'événements conteneur spécifiques standardisés pour le moment.
- ▶ Les portlets peuvent publier un événement depuis leur propre code de `processAction()` :

```
actionResponse.setevent(event, eventPayload)
```

Gestion des événements (3/3)

- ▶ Par contre, c'est plus propre et plus facile d'utiliser les annotations (nécessite un héritage de `GenericPortlet`) :

```
/**  
 * Cette méthode reçoit l'évènement "MyEvent"  
 * @ProcessEvent (qname={http://foo.com}MyEvent)  
 */  
public void processMyEvent(EventRequest request, EventResponse response)  
    throws PortletException, IOException {  
    ...  
}
```

Gestion des événements (2/3)

- ▶ La publication d'un événement engendre une ou plusieurs invocations de la nouvelle méthode `processEvent` dans la même ou les autres portlets.
 - ▶ De nouveaux événements peuvent aussi être publiés en implémentant `processEvents` :
- ```
actionResponse.setevent(event, eventPayload)
```
- ▶ Il faut noter que l'ordre de production des événements n'est pas garanti.
  - ▶ Exemple de traitement d'évènement :

```
public void processEvent(EventRequest request, EventResponse response)
 throws PortletException, IOException {
 String eventName = request.getEvent().getQName().toString();
 if(eventName.equals(...)) {
 ...
 } else if (eventName.equals(...)) {
 ...
 }
}
```

## Le « Resource Serving »

- ▶ Il permet aux portlets de servir des ressources :
  - Dans la version 1.0 des Portlet, les requêtes d'un Portlet retournaient une page entière.
- ▶ Il est utile pour :
  - la création dynamique d'images et d'autres ressources binaires,
  - le renvoi de fragments XML, JSON, HTML et des appels AJAX
  - etc.



## Les filtres

- ▶ Ce sont des composants Java permettant d'ajouter des traitements en amont et en aval de l'exécution des portlets.
- ▶ Ils sont postérieurs aux filtres de la spécification des Servlets.

## Caractéristiques d'un Portlet

- ▶ Elles offrent des caractéristiques supplémentaires par rapport aux Servlets.
- ▶ Modes de Portlets (Portlet Modes).
- ▶ États d'écran (Window States).
- ▶ Préférences Portlet.

## Les autres aspects

- ▶ Les portlets permettent une meilleure gestion du cache : Etags, resource caching, etc.
- ▶ Ils permettent aussi une meilleure introspection que les Servlets.
- ▶ Ils sont très utiles pour les Frameworks de développement.
- ▶ Ils offrent un support à l'écriture de headers HTTP.
- ▶ Ils utilisent des Annotations pour simplifier la configuration.
- ▶ Ils gèrent aussi les attributs de profile P3P (The Platform for Privacy Preferences Project) :
  - C'est un projet qui vise à standardiser le moyen par lequel un site web peut informer l'internaute de sa politique en matière de protection des données personnelles.

## Les modes de Portlet

- ▶ Chaque portlet a un mode courant indiquant la fonction qu'elle est en train de réaliser.
- ▶ Tous les portails Standard JAVA doivent supporter les différents modes : Edit, View et Help.
- ▶ Les modes d'un Portlet sont définis dans `portlet.xml`.
- ▶ Les développeurs peuvent créer leurs propres modes.

## Les portlets Liferay

- ▶ Liferay permet d'ajouter deux descripteurs de déploiement qui étendent portlet.xml :
  - liferay-portlet.xml.
  - liferay-display.xml.
- ▶ Optionnels mais permettent d'implémenter des fonctionnalités spécifique à Liferay
- ▶ Un fichier liferay-plugin-package.properties pour :
  - Renseigner les versions de Liferay compatibles avec la portlet.
  - Etc.

## Le fichier liferay-portlet.xml

```
<?xml version="1.0"?>
<!DOCTYPE liferay-portlet-app PUBLIC
 "-//Liferay//DTD Portlet Application 5.0.0//EN"
 "http://www.liferay.com/dtd/liferay-portlet-app_5_0_0.dtd">
<liferay-portlet-app>
 <portlet>
 <portlet-name>jsf-portlet</portlet-name>
 <instanceable>true</instanceable>
 <render-weight>1</render-weight>
 <ajaxable>true</ajaxable>
 </portlet>
 <role-mapper>
 <role-name>administrator</role-name>
 <role-link>Administrator</role-link>
 </role-mapper>
 <role-mapper>
 <role-name>guest</role-name>
 <role-link>Guest</role-link>
 </role-mapper>
</liferay-portlet-app>
```

## Le fichier liferay-display.xml

```
<?xml version="1.0"?>
<!DOCTYPE display PUBLIC
 "-//Liferay//DTD Display 5.0.0//EN"
 "http://www.liferay.com/dtd/liferay-display_5_0_0.dtd">
<display>
 <category name="category.sample">
 <portlet id="jsf-portlet"/>
 </category>
</display>
```

## Bilan

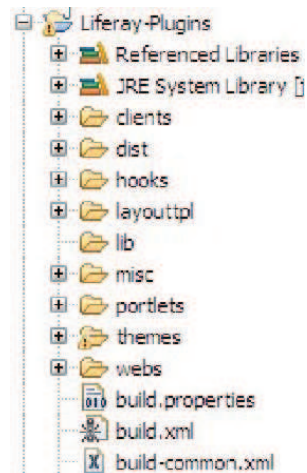
- ▶ Tous les produits, de type "portail", majeurs du marchés adhèrent à la spécification Java pour les Portlets (1.0 ou 2.0).
- ▶ Liferay Portal 4.x → Portlet 1.0 (JSR-168).
- ▶ Liferay Portal 5.x → Portlet 2.0 (JSR-286).
- ▶ Toute connaissance acquise autour de la spécification des Portlets Java peut être appliquée pour n'importe quel autre produit commercial ou libre disponible sur le marché.

## Personnalisation de Liferay (1/2)

- ▶ Il est fréquent, afin de répondre de la manière la plus adaptée qui soit, d'avoir à modifier le comportement d'origine du portail.
  - Par paramétrage.
  - Par modification.
- ▶ Ces modifications, parfois très simples, peuvent devenir une réelle difficulté au moment de procéder à une montée de version du logiciel :
  - si elles ne sont pas effectuées dans un cadre normalisé et conforme à la stratégie voulue par la communauté Liferay.

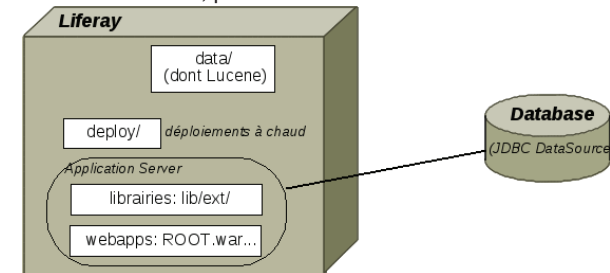
## L'environnement de développement (1/2)

- ▶ Le moyen privilégié est le plugins SDK, permettant de développer des plugins Liferay :
  - Portlet,
  - Layout Template,
  - Thèmes de présentation,
  - Hook.
- ▶ Il fournit un projet de développement et des scripts ANT structurants :
  - Compilation et déploiement unitaires,
  - Travail différentiel,
  - Déploiement à chaud.



## Personnalisation de Liferay (2/2)

- ▶ Le déploiement de sous-composants :
  - Plugin : Portlet, Theme (modèle de disposition), Hook.
  - Extension, patch.



- ▶ Le déploiement de plugins se fait la plupart du temps à chaud dans le répertoire `deploy`.
- ▶ Le déploiement d'extension ou de patch se fait la plupart dans les répertoires "librairie" ou "webapps".

## L'environnement de développement (2/2)

- ▶ Il permet notamment de réaliser des :
  - Thèmes de présentation :
    - Utilisé pour porter la charte graphique,
    - et la navigation.
  - Modèles de disposition :
    - "Colonage complexe ou placement au pixel."
  - Portlets :
    - soit la création de portlets spécifiques,
    - soit le "clonage" de portlets d'origine.
  - Hooks :
    - C'est la méthode d'extension à privilégier.

## Les Hooks

- ▶ Les hooks permettent la réalisation :
  - d'événements sur :
    - le démarrage de l'application.
    - le login (pré et post).
    - les services (pré et post).
  - de fichiers de traductions (language bundle).
  - d'événements sur le modèle (en particulier des entités comme User, Account, Layout...) :
    - create
    - update
    - delete
  - de JSP personnalisées :
    - les JSP d'origine peuvent être surchargées.

## Environnement de développement

### Composants requis sur le poste :

- ▶ JDK 1.5 ou plus.
- ▶ Un IDE Java adapté :
  - Netbeans,
  - Eclipse JEE (conseillé).
- ▶ Liferay (bundle Tomcat, JBoss...).
- ▶ (optionnel) une base locale (généralement MySQL) :
  - Pour consulter la base de données avec un browser sql.
  - Sinon, jetez il faut regarder les fichiers dans data/hsq.
- ▶ (optionnel) Ant :
  - pour le déploiement via ligne de commande

## Les Hooks JSP

- ▶ Les hooks jsp permettent de surcharger une jsp d'origine ;
  - mais en cas de montée de version, le problème reste entier...
- ▶ On peut rajouter des éléments à une jsp d'origine :
  - On privilégie alors l'extension par inclusion.
  - La jsp d'origine ne contient qu'une balise d'inclusion, vers une jsp contenant les nouveaux éléments.
  - Cela facilite la montée de version :
    - seule la jsp d'origine (avec pour seule modification une balise d'inclusion) est potentiellement impactée.

## Les thèmes de présentation (1/2)

- ▶ Le thème est le porteur de la charte graphique du site :
  - C'est une personnalisation d'un produit existant.
    - un forum = le forum de Liferay.
    - Idem pour l'agenda, le wiki, le blog, etc.
    - On ne construit pas un nouveau site de A à Z.
- ▶ Le "maximum de logique de mise en forme" doit être placé dans la CSS principale :
  - Styles CSS (fichier custom.css, voire les autres si besoin).
  - Et dans les templates .vm.

## Les thèmes de présentation (2/2)

- ▶ Pour personnaliser la présentation, on peut modifier et créer sur :
  - Les styles CSS,
  - Les images,
  - Les portlets (applications) spécifiques,
  - Via des templates (extension .vm), on peut modifier le contenu HTML sur :
    - La structure générale de la page (y compris l'entête ou le pied de page),
    - La navigation,
    - Le colonage des portlets,
    - Le contour des portlets.

## Les URL dans Liferay

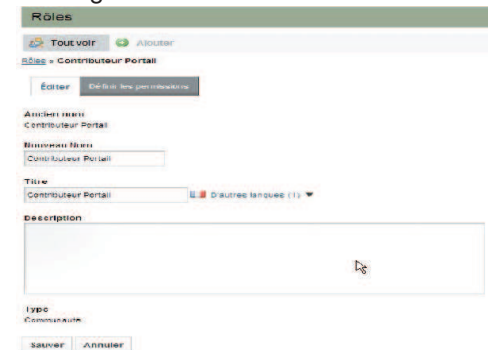
- ▶ Les « friendly » URL dans Liferay :
  - 3 niveaux :
    - /web/guest/home
    - /group/7cogs/home
    - /user/john/social
- ▶ Niveau 1 : désigne la portée de la page
  - web : zone publique
  - user : zone privée d'un espace personnel
  - group : zone privée d'une communauté ou d'une organisation
- ▶ Niveau 2 : désigne l'espace où chercher la page
  - les « friendly URL » de la communauté, de l'organisation, ou « screen name » de l'utilisateur.
- ▶ Niveau 3 : désigne le nom de la page (facultatif).

## Les droits et les espaces dans Liferay

- ▶ Les espaces de publication dans Liferay sont des sites autonomes.
  - La navigation est interne à chacun de ces espaces
- ▶ Deux types d'espaces sont offerts par Liferay
  - Communauté
    - Correspond à un espace de publication transverse : site ou mini-site, groupe de travail.
  - Organisation
    - Encapsule une communauté
    - Dispose d'autres attributs
    - Une organisation peut avoir une organisation parente.

## Configuration des droits et gestion de la sécurité

- ▶ Rôle : peut être régulier, appliqué sur une communauté ou sur une organisation.



- ▶ Pour chaque rôle créé, il faut définir les permissions :
  - Sur l'application
  - Sur le portail

## Configuration des droits et gestion de la sécurité

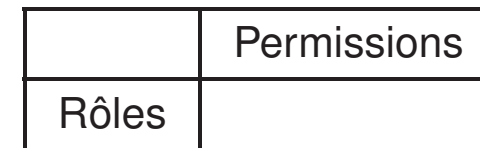
- Permissions associées à un rôle, sur une application donnée :

Application	Ressource	Action	Portée	Communauté
Contenu Web	Contenu web	Mettre à jour	Portail	<a href="#">Effacer</a>
Contenu Web	Contenu web	Vue	Portail	<a href="#">Effacer</a>
Contenu Web	Gestion de contenu web	Ajouter un contenu	Portail	<a href="#">Effacer</a>
Contenu Web	Gestion de contenu web	Ajouter un flux	Portail	<a href="#">Effacer</a>
Contenu Web	Gestion de contenu web	Ajouter un modèle	Portail	<a href="#">Effacer</a>
Contenu Web	Gestion de contenu web	Ajouter une structure	Portail	<a href="#">Effacer</a>
Contenu Web	Gestion de contenu web	Gérer la pré-publication	Portail	<a href="#">Effacer</a>

Affichage de 7 résultat(s).

## Configuration des droits et gestion de la sécurité

- Rôles et permissions : une notion transverse.
  - Dans Liferay, pratiquement tout objet (instance de portlet, sujet de forum, etc) peut être géré via un tableau de cases à cocher, croisant :
    - Permissions accordées sur l'objet (vue, configuration, etc.).
    - Rôle pour lequel la permission est accordée.



## Configuration des droits et gestion de la sécurité

- Exemple : Rôles et permissions associées à un utilisateur.

Rôle	Supprimer	Usurper	Permissions	Mettre à jour	Vue
Guest	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Owner	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Power User	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Publisher	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
User	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Writer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization Administrator	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization Member	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization Owner	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

## Installation de plugins (1/2)

- C'est identique à la configuration des plugins, mais un bouton supplémentaire est présent.

- Pour les thèmes et la présentations des plugins, il est possible d'utiliser des modèles de disposition.

## Installation de plugins (2/2)

### ► Actions possibles

- Upload d'un plugin, pour déploiement à chaud.
- Téléchargement d'un plugin (via une url).
- Parcourir la liste des plugins de la communauté Liferay

Parcourir le dépôt Télécharger un fichier Télécharger le fichier Configuration

Télécharger un fichier WAR pour installer un modèle d'affichage, une application, ou un thème.

Indiquez un contexte facultatif pour le déploiement. Par exemple, `sample-jsp-portlet`.

## Gestion des ressources

### ► Il est possible :

- de mettre à jour des caches (hibernate, cluster, etc.)
- d'utiliser la réindexation du moteur de recherche.

Actions

Lancer le ramasse-mièges pour libérer de la mémoire.

Effacer le contenu du cache de cette VM.

Effacer le contenu du cache à travers le cluster.

Effacer le cache de la base de données.

Réindexer tous les index de recherche.

Réinitialiser le sujet.

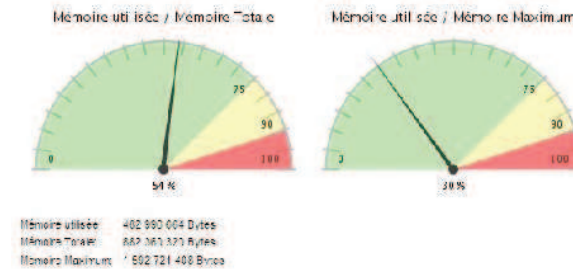
## Autres actions

### ► Administration du serveur :

- Présente un intérêt pour la surveillance et l'analyse, mais aussi pour des actions ponctuelles.

### ► Plusieurs ressources sont disponibles pour :

- la surveillance de la mémoire de la JVM et du garbage collector.



## Présentation de JBPM

- Projet racheté par JBoss.
- Licence apache.
- L'un (sinon le) des projets de workflow le mieux documenté.
- Très léger : 2 jar indépendants, ne nécessite pas l'utilisation d'un serveur d'application (ex : JBoss).
- Le seul à supporter la plupart des langues (XPD, BPEL, etc.).
- Enrichi XPD pour donner JDPL (JBPM Process Definition Language).

## Le moteur de workflow jBPM (1/3)

- ▶ JBPM est un moteur de workflow et de gestion de processus BPEL, adapté aux individus, aux applications et aux services.
- ▶ Le produit est parfait pour être intégré dans un environnement Java lightweight.
- ▶ Le kernel workflow est très léger et peut être extensible pour supporter des langages de définition de processus très variées.
- ▶ Actuellement les langages suivants sont supportés :
  - JPDL : pour le workflow humain,
  - PageFlow : pour la définition des enchaînements des formulaires,
  - BPEL : pour l'orchestration des services web.
- ▶ Les outils d'administration et monitoring workflow sont limités dans la version actuelle.

## Le moteur de workflow jBPM (3/3)

- ▶ JBoss jBPM facilite les modifications des processus métiers complexes, ce qui améliore significativement la souplesse de l'entreprise.
- ▶ Le projet est hébergé par la communauté JBoss et fait partie du stack middleware JEMS proposé par JBoss Group.
- ▶ RedHat est le leader du projet avec une équipe de développement partagé entre la Belgique et les États Unis.
- ▶ En plus de la communauté animée par JBoss Group, le projet est actif avec des utilisateurs et validateurs en Europe et États-Unis.

## Le moteur de workflow jBPM (2/3)

- ▶ JBoss jBPM est un moteur de workflow et de gestion de processus métiers.
- ▶ Il permet le contrôle des processus qui coordonnent les personnes, les applications et les services.
- ▶ Avec son architecture modulaire, JBoss jBPM combine le développement des applications workflow avec un moteur de processus flexible et extensible.
- ▶ L'éditeur de processus JBoss jBPM représente graphiquement les étapes d'un processus métier et facilite le lien entre les analystes et les développeurs.
- ▶ Puisque JBoss jBPM est présenté sous la forme d'une simple bibliothèque Java, il peut être utilisé
  - de façon indépendante dans un programme Java
  - ou avec un serveur d'applications comme JBoss Application Server, pour des applications d'entreprise hautement extensibles.

## Les fonctionnalités de jBPM (1/2)

JBPM est un moteur de processus flexible et extensible :

- ▶ Conçu pour fonctionner seul ou pour fonctionner parfaitement au sein de toute application Java ou J2EE.
- ▶ Modèle de programmation orienté processus qui associe le meilleur de Java aux avantages des techniques de programmation déclaratives.
- ▶ Support des langages de processus JPDL, PageFlow et BPEL.
- ▶ Définition graphique des processus via le Graphical Process Designer (GPD).
- ▶ Le langage JPDL proposé pour la définition des processus workflow humain est très intuitif mais on regrette que des standards workflow humain type XPDL ne soit pas supportés.



## Les fonctionnalités de jBPM (2/2)

- ▶ L'interface graphique de définition des processus sous forme de plugin Eclipse permet de masquer l'utilisation du langage JPDL.
- ▶ L'intégration avec le framework JBoss Seam permet d'interfacer facilement le workflow avec une application web.
- ▶ Les outils d'administration et monitoring workflow sont très limités dans la version actuelle.
- ▶ Néanmoins, elles offrent la possibilité d'exécuter graphiquement un processus workflow.
- ▶ De futures versions mettront en œuvre la technologie innovante de 'Process Virtual Machine' développée en commun par JBoss, OW2 et Bull.

## Caractéristiques de jBPM (2/2)

- ▶ Facilité d'emploi :
  - JBPM est une librairie générique pour la gestion des différents langages des processus.
  - Cette généricité rajoute de la complexité lors de la première prise en main du produit.
  - Le modèle de base est basé sur le concept « Graph Oriented Programming » qui certes apporte de la flexibilité mais qui diffère des approches traditionnelles de workflow.
- ▶ Intégration :
  - JBPM peut être intégré dans des systèmes très légers type Java :
    - application Java stand-alone,
    - ou application web ainsi que dans systèmes plus complexes type J2EE.
  - L'utilisation de JBPM dans des solutions non-java nécessite le développement d'une couche de communication web services.

## Caractéristiques de jBPM (1/2)

- ▶ Versions :
  - La version actuelle est la 3.2.3.
  - La prochaine version majeure devrait inclure le support complet du langage BPEL ainsi que l'éditeur graphique BPEL grâce à l'intégration du plugin Eclipse BPEL.
  - La version 4.x est en cours de développement et vise une nouvelle architecture plus légère et facilement extensible.
- ▶ Robustesse et performance :
  - JBPM est déployé en production dans des organisations appartenant à des domaines d'activités très différents tels que les finances ou la santé.
- ▶ Documentation :
  - La documentation en Anglais est complète.
  - Elle couvre l'installation du produit, les concepts de base, les exemples d'utilisation ainsi que la définition des processus et les concepts liés à la programmation.
  - Elle est disponible sous forme d'un document html.

## Généralisation de la notion de workflow (1/2)

- ▶ Par workflow on entend beaucoup de choses...
  - Cinématique des interactions utilisateurs.
  - La gestion des données.
- ▶ Toutes les applications ont un workflow :
  - Mais il est en général fixé, ou peu modifiable.
- ▶ Quelques exemples :
  - Un wizard,
  - Un formulaire suivi d'une validation.
- ▶ Certaines applications métiers ont une nécessité particulière de devoir s'adapter aux besoins d'une entreprise.
- ▶ Par exemple, tout gestionnaire de contenu (GED, CMS)
  - Rarement le même processus de validation.
  - Un nombre de validation différents, par des acteurs différents (un relecteur, suivi d'une validation par un autre journaliste, puis une autre validation par un directeur de programme).

## Généralisation de la notion de workflow (2/3)

- ▶ Mais aussi toutes les applications liées à la sécurité :
  - PKI (Gestion du cycle de vie des certificats).
  - Provisionning d'identité.
- ▶ Quelques constats :
  - Les développeurs sont rarement spécialisés métier.
  - Les consultants métiers ont rarement des connaissances techniques suffisantes pour manipuler du code.
- ▶ Les idées :
  - Création de moteur de workflow permettant un découpage entre les parties techniques et les parties métiers.

## Les WebFlow

- ▶ Les webflow (ou pageflow) sont un cas particulier des workflow dans le cas d'une utilisation d'un site web.
- ▶ Décris l'enchaînement des pages.
- ▶ Pas de persistance systématique.
- ▶ Une brique métier (au sens workflow) peut être composé d'un webflow lorsqu'il y a interaction avec l'utilisateur via une interface web.

## Généralisation de la notion de workflow (3/3)

- ▶ Certaines personnes ont réfléchi à des moteurs de workflow :
  - WfMC (Workflow Management Coalition).
- ▶ Les idées :
  - Gérer, archiver les instances de workflow.
  - Gérer les données manipulées au sein d'une instance de workflow.
  - Orchestrer les différents modules que composent une instance de workflow (briques métiers).
- ▶ En pratique cela donne :
  - Une vision métier (représentation sous forme de graphique) d'un workflow, généralement réalisé par un consultant métier.
  - Une « glue » technique permettant la mise en oeuvre du workflow métier.

## Exemples d'utilisation (1/2)

- ▶ L'équipe sécurité travaille sur le workflow pour l'application de PKI EJBCA, pourquoi ?
- ▶ Aucun client n'a le même besoin métier.
- ▶ Dans le cas d'une demande de certificat, il peut exister de très nombreux points d'entrée différents, et des cinématiques totalement différentes en fonction du point d'entrée.
- ▶ Les workflow métiers peuvent évoluer au sein d'un client, d'où l'idée de faire une application très modulaire.

## Exemples d'utilisation (2/2)

- ▶ Pour bien comprendre, prenons un exemple :
  - La DRH - à la signature d'un contrat - remplit un formulaire web (outil de gestion des identités interne à la société).
  - Cela provoque la demande de deux certificats pour cet utilisateur (signature et chiffrement).
  - À son arrivée, l'employé génère une carte à puce avec son certificat de signature.
  - Plus tard dans la journée, obtention de son certificat de chiffrement.
- ▶ Et un autre totalement différent :
  - Une entreprise souhaite équiper tous ses routeurs de certificats pour authentification ipsec.
  - Les routeurs génèrent une demande de certificat (via le protocole SCEP) auprès de la PKI.
  - Un administrateur doit valider les demandes.

## Le langage JDPL

### JBPM Process Definition Language

- ▶ Un workflow est composé d'un ensemble de node :
  - Un (unique) startNode : étape à l'origine de la création de l'instance du workflow.
  - TaskNode : étape en attente d'une interaction avec l'utilisateur.
  - State : étape en attente d'une action extérieure (webservices, attente de réponse d'un composant externe à l'application).
  - Un ou plusieurs endNode : archivage de l'instance du workflow et libération des ressources.
- ▶ Les nodes task et state ont un état wait

## Les aspects pratiques de la gestion de Workflow

- ▶ Dans un premier temps, il est question de définir un langage permettant de représenter le workflow, malheureusement il en existe un certains nombres :
  - BPM (Business Process Management)
  - XPD (XML Process Definition Language)
  - BPEL (Business Process Execution Language)
- ▶ La définition de ces langages permettent d'utiliser indépendamment différents moteurs de workflow.
- ▶ Permet également la création d'interface graphique permettant la manipulation des workflow.

## D'autres types de nodes

- ▶ Fork : séparation du workflow en N branches devant se réunir via un node de type join
- ▶ DecisionNode : condition sur une variable de l'instance de workflow (utilisation d'un langage simple ou délégation à une classe java).
- ▶ Node : action interne à l'application mais n'ayant pas d'interaction ni avec l'utilisateur ni avec un élément externe (exemple : notification par courriel).
- ▶ On relie tous ces nodes par des transitions, représentant le changement d'un node à un autre au sein de l'instance.

## Le comportement d'un node

- ▶ Chaque node possède trois événements déclencheurs :
  - onNodeEnter : déclenchée à l'entrée du node.
  - onNodeLeave : déclenchée à la sortie du node (n'a pas conscience du node suivant).
  - onTransition : déclenchée lors de la transition vers le node suivant.
- ▶ Pour information : un événement n'est pas lié qu'à une seule action, mais peut déclencher plusieurs actions.

## La gestion des données dans un workflow

- ▶ Très important : un node n'a pas à connaître l'existence des autres nodes, il manipule un ensemble de données dans un contexte d'exécution.
- ▶ C'est-à-dire :
  - Il n'y a pas de variables directement passées d'un node à l'autre
  - Un node utilise l'instance du workflow pour manipuler ses données
  - Limitation possible de la visibilité des variables en utilisant le mécanisme des swimlanes