

## Technologies conseillées sur cursus Atod (2014)

Principaux objectifs :

code modulaire , efficace , intelligible , bien structuré, maintenable , ....

correspondances claires entre les modèles fonctionnels (UML, BPMN, ....) et la mise en œuvre.

### 1.1. Technologies fondamentales

- **JDBC** et **JPA 2** (**Hibernate** ou OpenJPA) pour l'accès aux bases de données relationnelles (ex : **MySQL**)
- **EJB3** ou **SPRING** pour les **services métiers** et la **gestion automatique des transactions**
- **Servlet/JSP** et **JSF2** (+extension **primeFaces**) pour les interfaces graphiques "web"
- **CXF** (ou équivalent) pour la mise en œuvre des **services WEB (SOAP et REST)**
- **GIT** ou **SVN/subversion** ou .... pour le **référentiel de code source** (travail d'équipe)
- **Maven** pour la gestion des dépendances (".jar")
- **ActiveMQ** (ou équivalent) pour la gestion des files d'attentes **JMS**
- un éventuel **serveur "SMTP + pop3"** (ex: James ou ...) pour envoyer et recevoir des mails
- **Jbpm/drools** ou **Activiti** pour la mise en œuvre de processus "**bpmn2**".
- Plugin eclipse "**papyrus-UML**" ou ... pour la modélisation **UML**
- Plugin eclipse "**...Bpmn**" pour la modélisation BPMN2 (choisir entre une version compatible "drools/jbpm" ou une version compatible "activiti").

**NB** : prévoir éventuellement un référentiel (git ou svn) sur GitHub ou GoogleCode et les plugins correspondants sous eclipse.

**NB2**: Technologies qui vont bien ensembles :

Spring + CXF + activiti + "Spring security" + Hibernate/JPA + (Spring JMS) + ...

### 1.2. Technologies annexes (facultatives mais intéressantes)

- **MuleESB/community** ou **ServiceMix** (en mode Osgi avec Camel et sans JBI) ou ....
- **Nexus** comme référentiel d'artifacts maven
- **Jenkins** (ou ....) comme serveur d'intégration continue
- Plugin eclipse "**accéléo M2T**" comme générateur de code "**UML-->Java**" / **MDA** (à paramétrer)
- éventuelle autre technologie "web2/RIA" (ex: **GWT** ou **ext-js** ou **jquery** ou ....) pour ceux qui aiment (ne pas sous estimer le temps à passer pour l'intégration avec le reste)

### 1.3. Technologies déconseillées (sources de problèmes)

- **BPEL** en mode open source (assez compliqué , pas très bien intégré)  
BPEL est sérieusement applicable avec des produits payants d'Oracle ou IBM ou ...
- **richFaces 4** (plus complexe que **primeFaces 3.5** ) , attendre future version 5
- **JB1** (ancienne norme java sur ESB plus ou moins tombée à l'eau)
- **STRUTS2** (pas mieux que **JSF** , chemins relatifs bizarres , peu de composants évolués)
- ...

**NB** : ce document correspond à l'état de l'art "fin 2013 / début 2014" dans un monde plutôt orienté "java". D'autres styles/langages sont possibles .