

# Open Source et enjeux légaux : Etat de l'art, risques et opportunités



**JANUA**

OPEN SOURCE & OPEN STANDARDS

Telecom Valley - 11 Oct. 2006  
Pascal Flamand - Janua

# Introduction

- La vague Open Source est récente mais prend chaque jour une ampleur plus grande,
- Comme toute activité économique émergente, les acteurs sont de petite taille, le secteur inorganisé et le(s) modèle(s) économique(s) en cours de validation,
- La question du « légal » est au coeur des interrogations et des pratiques,

# Le rôle économique clé du logiciel libre...

- L'uniformisation des logiciels que les grands éditeurs veulent imposer a pour effet de réduire la compétitivité des entreprises utilisatrices.
- L'industrie du logiciel tend à stabiliser des emplois de haut niveau, embauche 30% de chercheurs, et génère des salaires conséquents qui favorisent l'économie locale.
- Depuis des années, l'industrie du logiciel vit au rythme des évolutions technologiques, et représente pour certains pays une source de revenu considérable (24% des exportations des USA, un marché en croissance de +28% en Chine). Avec l'approche des logiciels libres c'est une grande chance d'accroissement de la diversité nécessaire au développement économique du bassin de Sophia.

## Les défis de l'open source sont...

- L'intégration non disruptive de solutions Open-Source à des systèmes d'information traditionnels complexes.
- Le passage d'un business model basé sur licenses, royalties et clients captifs vers un modèle basé sur une offre de services et de déploiement de systèmes ouverts, où le pouvoir de choix et de décision est rendu au client.
- L'intégration, la gestion de solutions, de compétences et de ressources issues du monde entier, et en perpétuelle évolution mais avant tout du service donc de l'emploi "local".



# Les différentes catégories de software

Licence feature vs. Software type	Zero Price Avenue	Redistributable	Unlimited Usage	Source Code Available	Source Code Modifiable	Public « checks-Ins » to core codebase	All derivatives must be free
Commercial							
Trial Software	Non full featured						
Non Commercial Use	Usage dependent						
Shareware	Unenforced licencing						
Royalty-free binaries (« freeware »)							
Royalty-free libraries							
Open Source (BSD-Style)							
Open Source (Apache Style)							
Open Source (Linux/GNU Style)							



# Quels sont les risques perçus de l'utilisation des logiciels libres ?

- Des risques « techniques » :
  - les produits évoluent très vite, quid du MCO
  - quid du support et des engagements ? qui s'engage (SS2L, constructeur..)?
  - quid des forks et du suivi sur le long terme ?
- Des risques de « contamination » :
  - Propagation des termes de la licence à vos développements
  - perte de l'IP de vos développements
  - perte de la protection de vos softs (copyright, brevets...)
  - mise à disposition des sources de vos développements
- La désignation explicite de la loi applicable au contrat de licence

# Un peu de sémantique pour commencer...

- **L'hérité** : le créateur de logiciel désire que les modifications apportées à son logiciel, que ce soit par voie interne ou extension directe du code source, soient diffusées sous les conditions de la licence initiale (GPL, LGPL, QPL ou CeCILL OK)
- **La contamination** : le créateur désire que l'intégration d'un programme à son logiciel (ou que l'intégration de tout ou partie de son logiciel à un programme) soit diffusée sous les conditions de la licence d'origine, il doit adopter une licence dite contaminante GPL, QPL ou CeCILL.
- S'il souhaite que son logiciel **intégré à un autre** puisse être diffusé sous une autre licence que celle qu'il a initialement retenue, son choix doit se porter sur LGPL ou BSD. Cette hypothèse répond à la volonté du créateur de favoriser l'utilisation la plus large possible de son logiciel. Précisons que la LGPL est particulièrement recommandée pour les libraries.
- **Le code source modifié et sa diffusion** : S'il entend autoriser les licenciés à ne pas diffuser le code source du logiciel qu'ils auront eux-mêmes modifié, lequel ne sera alors redistribué que sous sa forme exécutable, la BSD est la seule licence possible.



# Une licence pour chaque cas...

	Le droit d'utiliser, de modifier le code source et le redistribuer	<b>L'hérédité</b> L'obligation de diffuser les modifications du code source avec les conditions de la licence initiale	La contamination** Licence contaminante	L'obligation de diffuser le code source modifié s'il y a diffusion du logiciel modifié
CeCILL	Oui	Oui	Oui	Oui
GPL	Oui	Oui	Oui	Oui
QPL	Oui	Oui	Oui	Oui
LGPL	Oui	Oui	Non	Oui
BSD	Oui	Non	Non	Non

# Les avantages de l'Open Source

- Pour l'utilisateur
  - Liberté de choix, ouverture des formats
  - Pérénnité : vous avez le source...
  - Coût
  - Technologie ouverte, possibilité d'adaptation illimitée
- Pour l'éditeur de logiciels
  - Diffusion rapide, levier marketing
  - Contribution de la communauté au développement
  - Utilisation de briques existantes

# Modèle(s) économique(s)

- Quelques traits généraux
  - La liberté du logiciel conduit à un abaissement des barrières (technologiques) à l'entrée
  - Corollaire : difficulté de vivre de la vente d'un produit
  - Modèle => rémunération sur les services
  - Rééquilibrage des forces entre le client (co-créateur) et l'entreprise
  - Importance des communautés
- Douze modèles économiques, 4 familles :
  - Les stratégies de produit
  - Les stratégies de service
  - La libération des logiciels non stratégiques
  - La vente de produits dérivés

# Les stratégies de produit

- Le « produit vedette » : produit au centre de la stratégie d'entreprise => rémunération par des services dédiés - importance de la marque (barrière commerciale) - exemples : Jboss, Zope, Jabber, Apache
- Intégrer des composants: valeur ajoutée dans intégration, tests, certifications, support.. - exemple : les distributions Linux, RH, Suse, Ubuntu,....
- La double licence : copyleft basé sur le copyright => versions copyleftée et propriétaire - rémunération sur la vente de licences propriétaires - exemples : MySQL, TrollTech (Qt)
- La licence chronodégradable : permettre la perception d'une « rente » pour rentabiliser l'investissement - principe :  $t$ =sortie de la version commerciale propriétaire,  $t+dt$ =publication de son code - exemple : AFPL Ghostscript & GNU - Ghostscript (GPL) ( $dt$  = environ 1 an)

# Stratégie de produit..(suite)

- Le «dynamitage» : Ouverture du code à des fins de diffusion
  - Parallèle avec la politique de la terre brûlée : tuer la concurrence en la privant d'un revenu - exemple : Open Office : perte de 20% de parts de marché de MS Office
  - Déplacer la lutte vers les services (ouverture du code => marché encore concurrentiel)
- La « vitrine » personnelle : un logiciel libre peut être une excellente carte de visite - exemple : Linus Torvalds et Linux, Guido van Rossum et Python, etc

# Les stratégies de service

- Le sur-mesure - exemple : ERP5
- Coopétition et édition mutualiste - exemple : IDX-PKI par IdealX
- Les services dérivés – Intégration, formation, etc....

# La libération des logiciels non stratégiques

- L'ouverture des gadgets : ouverture de tout ce qui n'est pas directement générateur de revenus (pilotes, etc) - exemple : Hewlett-Packard (pilotes d'imprimantes), Apple (base Darwin)
- L'ouverture de l'infrastructure : un site d'information vend de l'information, pas son infrastructure – exemple : LinuxFr basé sur Templeet, Wanadoo (FT) Messenger basé sur Jabber, Sun avec Solaris 10 et Java...

# La vente de produits dérivés

- Les accessoires
  - Merchandising (OpenStuff.Net), livre (O'Reilly)
- Prolongation du cycle de vie (rémunération sur des activités périphériques)
  - Exemple : ID Software (ouverture de Doom quand déclin et vente de scénarios)

# Faut-il passer tous ses softs en libre ?

- Pour le domaine des « utilities » (environnement de développement, serveur d'application, serveur web, messagerie, bureautique, SGBD, SSO) la réponse est clairement oui, la « valeur » sur des produits banalisés est dans le service et l'expertise...
- Pour les applications métiers (ERP, collectivités locales, Ecad-Mcad), la réponse est moins évidente :
  - Si nouvel entrant, oui, permet de « dynamiter » le marché et de lever les barrières à l'entrée
  - Si déjà sur le marché et bien positionné, le modèle apporte peu (pas de contribution, levier marketing inexistant...)

# Les solutions retenues par Janua..

- Le principe de la double licence pour les logiciels métiers (Jaguards et JAC):
  - Le coeur du produit (algo+fichiers de conf en OSS)
  - Les plugins métiers, installeurs et GUI payant
  - La vente avant tout de prestations
- La GPL pour les logiciels d'infra (Kontakt2WCAP)
  - Maintenance
  - Déploiements lourds

**Avez-vous des questions ?**

**Merci de votre attention**



**JANUA**

OPEN SOURCE & OPEN STANDARDS

Telecom Valley 10 Oct. 2006

Pascal Flamand

0 870 260 370

[pflamand@janua.fr](mailto:pflamand@janua.fr)