

// GUIDE OPEN SOURCE

RÉFLEXIONS SUR LA CONSTRUCTION ET LE PILOTAGE D'UN PROJET OPEN SOURCE



L'intérêt des entreprises pour les logiciels et solutions *Open Source* ne cesse de croître¹. Le Logiciel Libre représente aujourd'hui 3,6 % de la demande en Logiciels et Services en France et devrait atteindre 10% du marché dans 4 ans.

Cette activité, qui participe à l'évolution de l'informatique et de ses acteurs, contribue à renforcer la compétitivité de la France en matière de développement de logiciel². Qualité mais aussi robustesse, sécurité, évolutivité et innovation, le développement *Open Source* ne cesse de répandre ses bienfaits. Et ce n'est d'ailleurs pas une surprise si les acteurs traditionnels du logiciel – éditeur comme intégrateurs – ont aujourd'hui rejoint le mouvement. Hélas, nous sommes encore dans une phase d'acceptation, toutes les craintes ne sont pas encore tombées et le paysage est encore assez flou pour de nombreux clients. Ceci est d'autant plus paradoxal que notre conviction intime est que tout le monde fait aujourd'hui de l'*Open Source* !

Dans notre « évangélisation » en faveur de l'*Open Source* nous nous sommes aperçus que, même parmi ses acteurs, tous les changements engendrés par ces nouvelles pratiques n'étaient pas toujours bien perçus, que notre métier « manquait encore de maturité » sur cet aspect. Comment commercialiser un logiciel sous licence *Open Source* ? Quelle est la valeur qu'une entreprise peut apporter dans la chaîne de distribution d'un Logiciel Libre ? Quelles sont les garanties qu'une société peut offrir sur un projet *Open Source* et jusqu'où peut-elle céder son code ? À la vérité, la liste est encore longue et je ne ferai que répéter ce qui est si bien appréhendé par le document...

On dit que « ce qui se conçoit bien s'énonce clairement » et il était donc nécessaire de réfléchir sur les impacts – dans un premier temps juridiques et contractuels – de l'*Open Source* pour les entreprises. Celles-ci peuvent ainsi améliorer leurs pratiques et établir un discours clair pour leurs clients. C'est donc tout naturellement que la FniLL s'est saisie de ce sujet en se joignant au Syntec Informatique pour répondre au souhait commun de leurs adhérents : présenter dans un document clair et précis les enjeux, les impacts et les réponses à apporter aux licences *Open Source*.

Nous nous félicitons ainsi de la publication de ses travaux, résultat d'un foisonnement intellectuel intense ayant duré plus d'une année, qui représente à la fois une vulgarisation, un partage d'expérience et de préconisations quant à l'usage de l'*Open Source*.

Je n'ai plus qu'une chose à dire : bonne lecture !

Alexandre Zapolsky

Président de la Fédération Nationale de l'Industrie du Logiciel Libre
Président-Directeur Général de LINAGORA

¹ Étude OPIIEC Mars 2009 « Les métiers du logiciel libre en France », www.syntec-informatique.fr – tendance 2008-2009 du logiciel libre, www.ob2l.com

² Rapport de la Commission pour la libération de la croissance française, sous la présidence de Jacques Attali, La documentation française, 2008
Source : <http://lesrapports.ladocumentationfrancaise.fr/BRP/084000041/0000.pdf>

Syntec informatique : Chambre Professionnelle des Sociétés de Conseil et de services informatiques, des Éditeurs de Logiciels et des sociétés de Conseil en Technologies, Syntec informatique représente 1000 groupes et sociétés membres, soit près de 80% du chiffre d'affaires et des effectifs de la profession (300 000 collaborateurs, 42 Milliards d'euros de chiffre d'affaires).

Présidé depuis juin 2010 par Guy Mamou-Mani, Syntec informatique contribue au développement des Technologies de l'Information et de la Communication et de leurs usages, assure la promotion des entreprises des Logiciels & Services et la défense des intérêts collectifs professionnels. Syntec informatique, observateur et analyste privilégié du secteur des Logiciels & Services, informe l'ensemble de l'écosystème des TIC des chiffres et tendances de la profession et représente le secteur auprès de différents organismes et des pouvoirs publics.

www.syntec-informatique.fr



Le Logiciel Libre constitue une approche de plus en plus utilisée, notamment dans les administrations publiques, les industries ainsi que les secteurs technologiques (systèmes mobiles, embarqués, temps réel...). Bien avant cela, nombre de directeurs ou de responsables des Systèmes d'Information ont fait entrer le Logiciel Libre dans les couches basses de leur architecture (infrastructures), mais de plus en plus aussi vers les couches hautes (applications).

Une stratégie mixte et pragmatique consiste classiquement à panacher Logiciel Libre et logiciel propriétaire, quelque soit le composant concerné, en fonction de multiples paramètres techniques et organisationnels. Aussi, certains éditeurs de logiciels proposent une offre *Open Source* en marge de leur version standard, et couramment des éditeurs *Open Source* offrent des logiciels en double licence. Ceci rend la situation d'autant plus complexe.

A ce stade, la croissance de l'adoption des Logiciels Libres constitue l'un des facteurs de mutation du marché Français des logiciels et services. Elle engendre de nouveaux enjeux liés aux aspects vertueux et aux risques juridiques, et qui vont de la structure des modèles économiques jusqu'à la mise en place d'une gouvernance d'entreprise adaptée.

Autant de questionnements auxquels doivent répondre les différents services opérationnels des entreprises, de l'élaboration du contrat à la mise en œuvre technique et commerciale.

Dans notre approche très pragmatique de ce phénomène, nos clients et nous même sommes confrontés, au sein des nos applications internes et de nos efforts d'industrialisation, à la prise en charge de ces modèles de licences très variés et de leur impact sur nos métiers.

Afin de pouvoir refléter ce phénomène, et d'en cerner les différents aspects, Syntec Informatique et la FniLL (Fédération Nationale de l'Industrie du Logiciel Libre) ont décidé de mettre en place un groupe de réflexion commun. Avec ce groupe, nous analysons sans préjugés la symbiose des solutions *Open Source* avec les logiciels propriétaires, explicitant les définitions, le cadre général, les problématiques contractuelles et stratégiques.

Cette publication, nous l'espérons, aidera les acteurs confrontés à ces enjeux, à se poser les bonnes questions afin de mettre en place les nécessaires modes de fonctionnement. Le résultat, nous en sommes convaincus, en sera une prise en charge optimum avec les meilleurs bénéfices pour l'écosystème.

En vous souhaitant une utile et profitable lecture.

Olivier VALLET

Membre du Conseil d'Administration de Syntec informatique
Président du comité *Open Source*

SOMMAIRE

PRÉAMBULE	6
0. INTRODUCTION : PROPOS LIMINAIRES	9
0.1. LE LOGICIEL, L'APPRÉHENSION JURIDIQUE D'UN OUTIL TECHNIQUE	9
0.1.1. LE LOGICIEL, UN OBJET AVANT TOUT TECHNIQUE	9
0.1.2. LE LOGICIEL, UN OBJET JURIDIQUE	9
0.1.2.1. LA RECONNAISSANCE D'UN DROIT DE PROPRIÉTÉ INTELLECTUELLE	10
0.1.3. LES PRÉROGATIVES DU TITULAIRE DES DROITS DE PROPRIÉTÉ INTELLECTUELLE	11
0.2. LE LOGICIEL LIBRE, UNE FIGURE ORIGINALE	12
0.2.1. ORIGINE HISTORIQUE DU LOGICIEL LIBRE ET <i>OPEN SOURCE</i>	13
0.2.2. QUAND PARLE-T-ON DE LIBRE ET D' <i>OPEN SOURCE</i> ?	13
0.2.3. LA PLACE DES LICENCES <i>OPEN SOURCE</i> DANS NOTRE DROIT POSITIF	14
1. IMPACTS DE L'<i>OPEN SOURCE</i> SUR LA CONDUITE DE PROJETS INFORMATIQUES	15
1.1. PRISE EN COMPTE DE LA DIMENSION CONTRACTUELLE : PROCESSUS D'ANALYSE DES LICENCES.....	15
1.1.1. TENUE D'UN INVENTAIRE	16
1.1.1.1. RÔLE DE L'ÉQUIPE TECHNIQUE DE DÉVELOPPEMENT OU DE CONCEPTION	16
1.1.1.2. RÔLE DE L'ÉQUIPE JURIDIQUE	17
1.1.2. CHOIX DE LA LICENCE <i>OPEN SOURCE</i>	17
1.2. PRISE EN COMPTE DE LA DIMENSION TECHNIQUE : RAPPELS SUR LES LIVRABLES D'UN PROJET INFORMATIQUE	17
1.2.1. DESCRIPTION DU LIVRABLE	17
1.2.2. DESCRIPTION DU CODE SOURCE	18
2. IMPACTS DE L'<i>OPEN SOURCE</i> SUR LA GOUVERNANCE DE L'ENTREPRISE	19
2.1. IMPACTS SUR LA POLITIQUE COMMERCIALE ET ÉDITORIALE DE L'ENTREPRISE	19
2.1.1. LE MODÈLE « DISTRIBUTEUR À VALEUR AJOUTÉE »	19
2.1.2. LE MODÈLE DE LA LICENCE DOUBLE ET/OU DÉCALÉE	20
2.1.3. L'OPEN CLOUD COMPUTING	21
2.1.4. LE MODÈLE DE « SERVICES À VALEUR AJOUTÉE »	22
2.1.5. LE MODÈLE D'INTÉGRATEUR HYBRIDE	22
2.2. IMPACTS SUR LA RELATION CLIENT	22
2.2.1. L'EXPRESSION DES DEMANDES	23
2.2.2. COMPRENDRE LES MOTIVATIONS DES DEMANDES	23
2.2.2.1. LA PRÉSERVATION DU SAVOIR-FAIRE	23
2.2.2.2. AUTONOMIE VIS-À-VIS DU FOURNISSEUR	24
2.3. IMPACTS SUR L'ORGANISATION DES SERVICES DE L'ENTREPRISE	25
2.3.1. IMPACTS SUR LES SERVICES EXISTANTS	25
2.3.1.1. SERVICE TECHNIQUE	25
2.3.1.2. SERVICE JURIDIQUE	26
2.3.1.3. SERVICE ACHAT	26
2.3.1.4. SERVICE MARKETING ET COMMERCIAL	27
2.3.1.5. SERVICE RESSOURCES HUMAINES	27
2.3.1.6. DIRECTION GÉNÉRALE	28
2.3.2. ÉVOLUTION ORGANISATIONNELLE RECOMMANDÉE	28
2.3.2.1. RÉFÉRENT <i>OPEN SOURCE</i>	28
2.3.2.2. COMITÉ DE DIRECTION <i>OPEN SOURCE</i>	29

2.3.2.3.	COMITÉ DE REVUE OPEN SOURCE	29
2.3.2.4.	CHARTRE OPEN SOURCE DE L'ENTREPRISE	29
2.3.2.5.	APPROFONDISSEMENTS	30
2.3.3.	RELATIONS AVEC LA COMMUNAUTÉ <i>OPEN SOURCE</i>	30
3.	IMPACTS DE L'<i>OPEN SOURCE</i> SUR LES CONTRATS	31
3.1.	CHOISIR LA LICENCE <i>OPEN SOURCE</i> ADAPTÉE AU BESOIN	31
3.1.1.	TACTIQUE ET TECHNIQUE DANS LE CHOIX DES LICENCES <i>OPEN SOURCE</i>	31
3.1.1.1.	STRATÉGIE DANS LE CHOIX DE LA LICENCE	31
3.1.1.2.	PERSONNALISATION DE LA LICENCE	32
3.1.2.	TYPES DE LICENCES <i>OPEN SOURCE</i>	32
3.1.2.1.	TYPOLOGIE DES LICENCES	32
3.1.2.2.	L'EXEMPLE DE LA GNU GPL	34
3.1.3.	ENJEUX DE LA COMPATIBILITÉ ENTRE LICENCES	36
3.1.3.1.	LICENCE UNIQUE	36
3.1.3.2.	CAS D'INCOMPATIBILITÉ ET SOLUTIONS ALTERNATIVES	37
3.2.	RÉDIGER LE CONTRAT ADAPTÉ À LA LICENCE	37
3.2.1.	LA CLAUSE « DÉFINITIONS »	38
3.2.2.	LA GARANTIE D'ÉVICTION	38
3.2.2.1.	LE PRINCIPE	38
3.2.2.2.	DANS LE CONTRAT DE SERVICES	39
3.2.2.3.	LES CONSÉQUENCES LIÉES AUX SOUS LICENCES (LIBRES)	39
3.2.3.	GARANTIE CONTRACTUELLE ET MAINTENANCE	40
3.2.3.1.	LES GARANTIES TECHNIQUES SUR LE LOGICIEL	40
3.2.3.2.	LA MAINTENANCE	40
ANNEXE 1 :	LEXIQUE TECHNIQUE ET JURIDIQUE	41
ANNEXE 2 :	FICHE SYNTHETIQUE DECRIVANT UNE LICENCE ANNEXEE AU CONTRAT	42
ANNEXE 3 :	ANNEXE CONTRACTUELLE DE DESCRIPTION D'UNE FOURNITURE LOGICIELLE	43
	NOM ET DESCRIPTIF FONCTIONNEL SOMMAIRE.....	43
	LISTE DES COMPOSANTS EXTERNES UTILISÉS	43
	AU NIVEAU DU PROCESSUS DE BUILD.....	43
	AU NIVEAU DU PROCESSUS D'INTÉGRATION/PACKAGING.....	43
	AU NIVEAU DU PROCESSUS DE DÉPLOIEMENT	44
	DESCRIPTION DU CONTENU DE LA LIVRAISON À UN UTILISATEUR/CLIENT	44
	DESCRIPTION DU CONTENU DE LA LIVRAISON DESTINÉE À UN SÉQUESTRE	44
	SOLUTIONS FINANCIÈRES AVEC UN LEASER (SOCIÉTÉ DE LEASING, DE CRÉDIT-BAIL) :	45
	FOURNITURE À UN INTÉGRATEUR.....	45
	SYSTÈME DE PREUVE VIS-À-VIS D'UNE AUTORITÉ DE TUTELLE.....	46
	SYSTÈME DE PREUVE EN MATIÈRE DE DROITS D'AUTEUR	46
ANNEXE 4 :	MODÈLE DE DOCUMENT DE DIFFUSION LINAGORA : LICENCE, DIFFUSION ET CONTRIBUTEURS	47
ANNEXE 5 :	RÉFÉRENT <i>OPEN SOURCE</i>	48
ANNEXE 6 :	TABLEAU DE CLASSIFICATION ET DE CROISEMENT DES LICENCES	49
	TABLEAU 1 : REDISTRIBUTION SOUS UNE AUTRE LICENCE (COMPATIBILITÉ)	49
	TABLEAU 2 : REDISTRIBUTION SOUS UNE AUTRE LICENCE (DOCUMENTATION)	50
ANNEXE 7 :	LISTE DE RESSOURCES UTILES RELATIVES AUX LICENCES OPEN SOURCE	51
ANNEXE 8 :	DÉFINITION DE L'OSI COMMENTAIRES/REMARQUES	52
ANNEXE 9 :	LICENCE ET PROCHAINE VERSION DU DOCUMENT	53
REMERCIEMENTS		59

SIGLES

Art. :	Article
ASP :	Application Service Provider
BSD :	Berkeley Software Distribution
C. civ. :	Code civil
CC :	Creative Commons
CD :	Compact Disk
CeCILL :	CEA, CNRS, INRIA Logiciel Libre
CIL :	Correspondant Informatique et Libertés
CNIL :	Commission nationale de l'informatique et des libertés
CPU :	Central Processing Unit
cf :	se référer à
CPI :	Code de la propriété intellectuelle
CSPLA :	Conseil Supérieur de la Propriété Intellectuelle
CVS :	Concurrent Versions System
DADVSI :	Loi n° 2006-961 du 1er août 2006 relative au droit d'auteur et aux droits voisins dans la société de l'information
DRM :	voir MTP
GNU :	GNU's Not Unix
GNU AGPL :	GNU Affero General Public License
GNU GPL :	GNU General Public License
GNU FDL :	GNU Free Documentation License
GNU LGPL :	GNU Lesser General Public License
GNU SFDL :	GNU Simpler Free Documentation License
EFF :	Electronic Frontier Foundation
EOLE :	European OpenSource & Free Software Law Event
EPL :	Eclipse Public License
EUPL :	European Union Public License
FLOSS :	Free Libre Open Source Software
FniLL :	Fédération Nationale du Logiciel Libre
FSF :	Free Software Foundation
FSFD :	Free Software Definition
FFII :	Foundation for a Free Information Infrastructure
HP :	Hewlett-Packard
INPI :	Institut national de la propriété industrielle
LAL :	Licence Art Libre
MPL :	Mozilla Public License
MTP :	Mesure Technique de Protection
OEB :	Office Européen des Brevets
OHMI :	Office de l'harmonisation dans le marché intérieure (Alicante)
OMPI :	Organisation Mondiale de la propriété intellectuelle
OIN :	Open Invention Network
OSD :	Open Source Definition
OSDL :	Open Source Development Laboratory
OSI :	Open Source Initiative
OSL :	Open Software License
PI :	Propriété Intellectuelle
PME :	Petites et Moyennes Entreprises
R&D :	Recherche et Développement
SaaS :	Software as a Service
SFLC :	Software Freedom Law Center
SLA :	Service Level Agreement
SSII :	Société de Services en Ingénierie Informatique
SSLL :	Société Spécialisée en Logiciels Libres
Th. :	Thèse
URL :	Uniform Resource Locator
URI :	Uniform Resource Identifier
v. :	Version
VVL :	Veni, Vidi, Libri

PRÉAMBULE

Les Logiciels Libres – ou *Open Source* – sont aujourd'hui couramment – si ce n'est systématiquement – utilisés dans l'industrie de l'informatique (édition, intégration, système embarqué, etc.). Il est même possible d'affirmer, sans langue de bois, qu'ils font partie intégrante de ces métiers : quel chef de projet ne s'est jamais posé la question de la réutilisation d'un Logiciel Libre ? Quel client ne s'est pas vu proposer des solutions basées – au moins en partie - sur des composants sous licence *Open Source* ?

Après un laps de temps qui aura été nécessaire à ce déploiement, il semblerait donc qu'il ne soit plus utile de militer en faveur du Logiciel Libre, mais de constater qu'il fait partie du « paysage » et en accompagner le développement. En cela, il faut comprendre comment se comporter vis-à-vis de celui-ci (en tant que simple utilisateur ou éditeur) afin de modifier la gouvernance des projets, et les procédures organisationnelles de développement. Pour tirer un maximum de profit de ce choix, il convient d'appréhender tous les impacts du recours aux Logiciels Libres. Deux attitudes sont possibles : admettre ses particularismes et agir en conséquence ou continuer à se voiler la face, mais pour combien de temps encore...

Au-delà de la prise en compte des spécificités des Logiciels Libres, de nombreuses questions se posent quant à la pérennité du système « propriétaire » et le maintien des garanties les plus larges quant aux logiciels commercialisés. À l'heure du développement voire du succès des logiciels et solutions *Open Source*, les services juridiques et nombre d'opérationnels sont peu – voire pas du tout – formés aux différents types de licences *Open Source*. Ils éprouvent des difficultés à appréhender ce phénomène et à valider les contrats intégrant ces dernières.

Pour répondre à leur interrogation et les aider à comprendre ce phénomène, Syntec informatique et la FniLL (Fédération Nationale de l'Industrie du Logiciel Libre) ont décidé de mettre en place **un groupe de réflexion commun** afin de permettre à leurs membres de partager et de mutualiser leurs expériences et compétences. Composé de compétences hétérogènes et complémentaires (juristes, développeurs, chefs d'entreprise, consultants, architectes, etc.) et assisté de nombreux experts, le groupe s'est réuni régulièrement durant l'année 2009 et livre aujourd'hui le résultat de ses réflexions.

Dépassant largement le cadre juridique initialement visé, ce présent document se veut **une analyse détaillée des principes de l'Open Source** et de ses impacts pour les sociétés qui décident d'y recourir. L'*Open Source*, et plus largement l'*Open Innovation*, constitue en effet une nouvelle figure économique qui s'inscrit dans un mouvement plus vaste de dématérialisation, d'externalisation et de valorisation du patrimoine de l'entreprise.

Par conséquent, la présente étude s'adresse aussi bien aux sociétés développant des logiciels intégralement sous licence libre ou *Open Source*³ qu'à celles imbriquant ces derniers au sein de logiciels plus vastes (éditeurs de logiciels, SSII, intégrateurs *lato sensu*), ou celles (simplement ?) utilisatrices.

Le présent document est un guide de gouvernance qui a pour vocation de présenter les contraintes et les avantages inhérents aux licences *Open Source* et à l'usage des Logiciels Libres. On s'apercevra que l'impératif essentiel, tient dans la mise en place d'une collaboration active entre les différents services (notamment technique et juridique) de l'entreprise. Cette collaboration est, en effet, indispensable pour répondre aux besoins de sécurité, de traçabilité et de pérennité. Ces objectifs devraient conditionner le choix des développeurs, et par suite, le choix des utilisateurs et clients de Logiciels Libres. Autre impératif, renforçant les enjeux organisationnels au sein des entreprises, les principes fondamentaux et originaux du Logiciel Libre doivent être pris en compte, depuis le département R&D jusqu'au service commercial, en passant par les services juridiques et ressources humaines. Cela permet de sécuriser la démarche de développement en amont, et d'optimiser la démarche commerciale en aval. Si la dimension technique reste maîtresse dans l'édition de logiciels, il convient de maîtriser l'ensemble des conséquences juridiques que l'emploi de modules ou d'environnements libres et *Open Source* entraîne pour l'exploitation finale du logiciel.

Ainsi, recourir aux Logiciels Libres accroît la nécessité d'un travail collaboratif interservices et, ce faisant, une plus grande rigueur dans l'établissement des méthodes de travail et l'organisation interne des sociétés. Il est capital de disposer de réflexes et de principes opérationnels encadrant avec précision l'introduction, l'utilisation et la commercialisation de Logiciels Libres.

³ Cf. terminologie *infra*.

Le présent document se présente comme un guide de bonnes conduites recensant les pratiques en matière :

- de conduite de projets informatiques intégrant des composants libres ou entièrement sous licence *Open Source* (compréhension et sécurisation) ;
- d'organisation des services de l'entreprise (gouvernance) ;
- de contractualisation des services fournis (grâce à un recensement et une analyse des pratiques actuelles).

Après le rappel des principales notions juridiques relatives au logiciel en général et au Logiciel Libre, en particulier (**Partie 0**), il est apparu indispensable de mesurer les impacts de l'*Open Source* sur :

- La conduite de projet (**Partie 1**)
- La gouvernance de l'entreprise (**Partie 2**),
- Les contrats (**Partie 3**).

Pour donner à ces travaux la diffusion la plus large possible et permettre la rédaction de nouvelles versions, cette étude est distribuée cumulativement sous les licences GNU FDL 1.3 ⁴, Creative Commons By-SA 3.0 ⁵ et Licence Art Libre 1.3. Soyez donc « libres » de partager ces réflexions, de les copier et diffuser afin de participer à l'expansion de ces principes et bonnes pratiques – dans le respect des mentions de paternité et de licences.

En espérant que la lecture de ce document soit aussi instructive pour vous que sa rédaction le fut pour nous !

Olivia FLIPO

Avocat à la cour
Cabinet Staub & Associés

Benjamin JEAN

LINAGORA
Floss legal expert
Responsable du centre juridique open source

⁴ Voir le texte entier en annexe et sur le site GNU <http://www.gnu.org/copyleft/fdl.html>

⁵ Voir le texte entier en annexe et sur Creative Commons By-SA <http://creativecommons.org/licenses/by-sa/3.0/legalcode>

0. INTRODUCTION : PROPOS LIMINAIRES

0.1. LE LOGICIEL, L'APPRÉHENSION JURIDIQUE D'UN OUTIL TECHNIQUE

La Commission de terminologie et de néologie de l'informatique⁶ définit le logiciel comme un « *ensemble des programmes, procédés et règles, et éventuellement de la documentation, relatifs au fonctionnement d'un ensemble de traitement de données* ».

La notion revêt néanmoins de multiples facettes selon le regard qu'on lui porte. Nous en retiendrons deux qui nous intéressent plus particulièrement : le logiciel comme *objet technique* d'une part et comme *objet juridique* d'autre part⁷.

0.1.1. LE LOGICIEL, UN OBJET AVANT TOUT TECHNIQUE

Techniquement, ce que l'on appelle logiciel correspond à « *un ensemble d'instructions et de règles qui permet à un ordinateur ou à un système informatique d'assurer une tâche ou une fonction en particulier* ». D'autres termes renvoient à cette même définition et celui qui s'en rapproche le plus – tout en étant moins ambigu – est celui de programme.

Sa forme peut être variée mais deux états se distinguent clairement :

- **Le code source**, qui est l'ensemble des lignes de codes qui constituent les instructions et règles de fonctionnement du logiciel. Il peut se définir comme un ensemble d'instructions écrites dans un langage de programmation⁸ informatique⁹ compréhensible par un être humain, permettant d'obtenir un programme¹⁰ pour un ordinateur¹¹.
- **Le code objet** est l'ensemble d'instructions qu'est capable de lire et d'exécuter¹² un microprocesseur¹³ (en code binaire exécutable). Un code objet est obtenu en compilant un code source¹⁴. La conséquence est que le code objet (résultant de la compilation) n'est pas directement intelligible par l'esprit humain.

Il est généralement nécessaire de procéder à la compilation du code source en code objet pour pouvoir faire fonctionner le logiciel, mais certains langages, dits « langages interprétés » (PHP, Ruby, Perl, etc.), sont convertis en instructions exécutables par la machine au moment de leur exécution (et sont donc distribués sous forme de code source). Dans la même veine, d'autres nécessitent la présence d'un interpréteur, on parle alors de langages semi-interprétés (par exemple Java, avec la Machine Virtuelle Java).

0.1.2. LE LOGICIEL, UN OBJET JURIDIQUE

Depuis leur création, les logiciels ont rapidement acquis une valeur patrimoniale forte, tant pour les auteurs (ou société éditrice) que pour les utilisateurs. Par nature immatériels – et donc non rivaux (l'usage des uns ne limite pas l'usage des autres) et non exclusifs (tout le monde peut en jouir) –, contrôler leur diffusion était très difficile.

Ainsi, pour répondre au besoin d'appréhension de ce bien immatériel, les juristes se sont très vite interrogés sur la meilleure protection à conférer au développeur d'un logiciel, entre brevet (auquel on songe en raison de son aspect technique) et droit d'auteur (considérant le langage informatique comme une forme d'expression).

C'est la seconde qualification qui fût retenue en France et plus largement en Europe : les législations reconnaissant par ce mécanisme un droit de propriété intellectuelle aux auteurs de logiciels.

⁶ Par un arrêté du 22 décembre 1981 pour l'« Enrichissement du vocabulaire de l'informatique ».

⁷ Pour être plus complet, il faudrait au moins parler de la vision informationnelle (un ensemble d'instructions écrites en langage humain et traduit en code binaire pour être interprété par une machine électronique) et consumériste (le logiciel étant un produit de consommation, soumis en cette qualité aux dispositions du code de la consommation).

⁸ http://fr.wikipedia.org/wiki/Langage_de_programmation

⁹ <http://fr.wikipedia.org/wiki/Informatique>

¹⁰ http://fr.wikipedia.org/wiki/Programme_%28informatique%29

¹¹ <http://fr.wikipedia.org/wiki/Ordinateur>

¹² <http://www.dico.fr.com/cgi-bin/n.pl/dicofr/definition/20010101001811>

¹³ <http://www.dico.fr.com/cgi-bin/n.pl/dicofr/definition/20010101003501>

¹⁴ <http://www.dico.fr.com/cgi-bin/n.pl/dicofr/definition/20010101001128>, Liste des instructions d'un programme exprimées dans un langage que l'homme est capable de manipuler aisément. Sans le code source il est très difficile de modifier un programme.

0.1.2.1. LA RECONNAISSANCE D'UN DROIT DE PROPRIÉTÉ INTELLECTUELLE

1. Une protection du logiciel en tant qu'Œuvre

Lors des premiers contentieux en matière de logiciel et devant le silence de la loi, les juges se sont naturellement tournés vers l'édifice législatif et jurisprudentiel du droit d'auteur en assimilant l'écriture d'un logiciel à celle d'une œuvre littéraire.

Le législateur a confirmé peu de temps après cette orientation par la loi du 3 juillet 1985¹⁵ qui est venue compléter l'article L. 112-2 du CPI d'un alinéa 13¹⁶ afin de désigner expressément le logiciel comme une « œuvre »¹⁷. Néanmoins, pour prendre en compte la dimension industrielle de cette création, quelques ajustements furent apportés au droit d'auteur « artistique » au détriment de l'auteur, personne physique d'un logiciel : il s'agit de l'attribution automatique des prérogatives patrimoniales du droit d'auteur à l'employeur, et de l'amputation d'une partie des droits moraux, etc.

Corollaire de ce choix, les lois nationales et communautaires ont expressément exclu le logiciel du champ de la brevetabilité¹⁸.

Néanmoins, d'autres pays (parmi lesquels les États-Unis et le Japon) recourent à la notion de brevets logiciels. C'est probablement l'une des raisons pour lesquelles une pratique semblable s'est développée, en dehors de tout cadre juridique, au sein des offices nationaux européens (au premier rang desquels l'Office Européen des Brevets) : le dépôt de brevets est ainsi accepté sur les « inventions mises en œuvres par ordinateur », puis aujourd'hui sur des logiciels en tant que tels. Frein à l'innovation et à l'interopérabilité, cette extension du brevet est une problématique complexe et souvent remise en question¹⁹.

2. Une protection différenciée des composantes du logiciel

Autour du programme en tant que tel, d'autres composantes (qui sont des stades antérieurs du logiciel, ou qui accompagnent ce dernier) forment ce que l'on appelle le « logiciel » :

- **Le matériel de conception préparatoire** qui couvre, selon la directive communautaire, « l'ensemble des travaux de conception aboutissant au développement d'un programme à condition qu'il soit de nature à permettre la réalisation d'un programme d'ordinateur à un stade ultérieur ». Ceci comprend les ébauches et les maquettes, les analyses, les organigrammes, mais pas les spécifications fonctionnelles ni le cahier des charges initial qui sont des préalables au développement du code source et restent au stade des idées non protégeables.

- **L'architecture du programme** : il s'agit de la façon dont sont enchaînés les différents sous-programmes et dont sont déclarées et utilisées les variables. C'est essentiellement l'architecture qui différencie deux programmes aux fonctionnalités identiques.

A contrario, les fonctionnalités d'un logiciel et les **algorithmes** (processus systématiques de résolution d'un problème par le calcul) ne peuvent être protégées (contrairement à leur implémentation) car ils restent assimilés aux idées et sont « de libre parcours ».

Enfin, en périphérie du logiciel, d'autres éléments sont aussi protégés :

- **Le nom du programme** peut être protégé par le droit d'auteur comme le titre d'une œuvre, s'il présente une originalité – protection pouvant éventuellement se cumuler avec les droits issus d'une marque déposée.

- **La documentation** qui permet d'utiliser le logiciel. Selon les dispositions légales applicables au logiciel, la documentation, œuvre de l'esprit écrite, fait partie du logiciel et bénéficie de sa protection juridique. On doit néanmoins la différencier de la documentation « utilisateur » qui, elle, sera protégée de manière indépendante.

¹⁵ Loi n°85-660 du 3 juillet 1985 relative aux droits d'auteur et aux droits des artistes-interprètes, des producteurs de phonogrammes et de vidéogrammes et des entreprises de communication audiovisuelle.

¹⁶ « Sont considérés notamment comme œuvres de l'esprit au sens du présent code : [...] 13° Les logiciels, y compris le matériel de conception préparatoire ; [...] »

¹⁷ Le caractère scientifique des logiciels n'exclut pas pour autant cette qualification d'œuvre de l'esprit : « L'élaboration d'un programme d'ordinateur est une œuvre de l'esprit originale dans sa composition et son expression allant au-delà d'une simple logique automatique et contraignante, il ne s'agit pas d'un mécanisme intellectuel nécessaire, les analystes-programmeurs ont à choisir comme les traducteurs d'ouvrages entre divers modes de présentation et d'expression, et leur choix porte ainsi la marque de leur personnalité » (TGI Paris, 27 mars 1987).

¹⁸ Art. L 611-10 du Code de la propriété intellectuelle et Article 52 de la Convention du Brevet Européen du 5 octobre 1973.

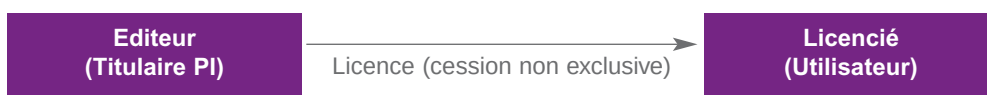
¹⁹ Le Parlement européen ayant rejeté le 6 juillet 2005 la directive « brevet logiciel » (par 648 voix pour, 14 contre et 18 abstentions) qui proposait de légaliser la pratique de l'Office Européen des Brevets.

0.1.3 LES PRÉROGATIVES DU TITULAIRE DES DROITS DE PROPRIÉTÉ INTELLECTUELLE

Dès lors que le logiciel est une création originale, un monopole est établi au profit de son développeur qui est investi des droits moraux sur la création et des droits patrimoniaux d'exploitation.

1. Les droits patrimoniaux dévolus à l'auteur, un droit exclusif d'exploitation

Les droits patrimoniaux sont les droits qui permettent l'exploitation du logiciel. C'est l'utilisation de ces droits que l'on aménage par contrat (souvent appelé « licence ») : les droits d'utilisation ou d'exploitation sont cédés, le plus souvent de façon non exclusive, par le titulaire de droits (éditeur ou SSII par exemple) sur le logiciel au licencié (utilisateurs par exemple).



Les droits patrimoniaux sont de trois catégories :

- a) Le droit de reproduction (notamment le chargement, l'affichage, l'exécution, la transmission ou le stockage du logiciel²⁰) – seul l'usage du logiciel en SaaS (*Software as a Service*) supprime la nécessité pour l'utilisateur de disposer d'un tel droit ;
- b) Le droit de distribution qui permet au titulaire de droits d'autoriser la mise sur le marché, à titre onéreux ou gratuit ;
- c) Le droit de modification²¹ : **droit de correction**, nécessaire pour maintenir le logiciel en état de fonctionnement et la conformité du logiciel à l'usage décrit ; et **droit d'adaptation**, entendu comme le droit de procéder à des modifications et évolutions fonctionnelles du logiciel.

2. Le droit moral de l'auteur, un droit extrapatrimonial

Par ailleurs, l'auteur dispose de multiples autres prérogatives regroupées au sein de la notion de « droit moral ». Dans la tradition du droit d'auteur, le droit moral est un droit « *perpétuel, inaliénable et imprescriptible* »²² de l'auteur sur son œuvre : il ne peut pas le céder et/ou le perdre par le non usage.

Traditionnellement, ce droit se compose :

- Des droits **de divulgation** (qui permet de décider de la première mise à disposition),
- Du **droit de repentir et de retrait**,
- Du **droit au respect de son nom et de sa qualité** (droit de paternité).
- Du **droit au respect de l'œuvre**.

En revanche, l'auteur d'un logiciel ne peut exercer son droit de repentir ou retrait (pour retirer son logiciel du marché), ni s'opposer à la modification du logiciel par son employeur²³.

3. Les notions d'auteur et de titulaire de droits

Par principe, l'auteur d'une œuvre (logicielle ou non) est le seul titulaire de droits, c'est-à-dire celui qui crée l'œuvre (logicielle ou non), mais le CPI prévoit un système dérogatoire au droit commun lorsque l'auteur d'un **logiciel** est un **salarié**. Il prévoit une dévolution automatique des droits patrimoniaux à l'employeur dès lors que le salarié a créé le logiciel « *dans l'exercice de ses fonctions ou d'après les instructions de son employeur* »²⁴ (CPI, art. L. 113-9).

²⁰ Le Code prévoit toutefois une exception, celle de la copie de sauvegarde (lorsque celle-ci est nécessaire pour préserver l'utilisation du logiciel).

²¹ En principe, il n'est jamais concédé aux utilisateurs, mais éventuellement à des prestataires intermédiaires chargés de la maintenance corrective et/ou évolutive du logiciel (Tierce Maintenance Applicative). Le CPI prévoit également un droit de décompilation très fortement limité.

²² Ce droit est inaliénable en droit français, par opposition au copyright anglo-saxon.

²³ CPI, art. L. 121-7.

²⁴ Étant entendu qu'il est tout à fait possible d'y déroger contractuellement au profit du salarié.

L'employeur est alors titulaire des droits patrimoniaux et, partant, il bénéficie de l'ensemble des prérogatives attachées à l'exploitation de l'œuvre.

Ainsi, en vertu de la dévolution automatique, c'est donc, sauf stipulation contraire²⁵, l'entreprise développant un logiciel qui, en tant que titulaire des droits d'auteur, décide de son développement, de sa mise sur le marché et choisit la licence sous laquelle le logiciel sera commercialisé.

4. La licence, un outil juridique d'exploitation du logiciel

La société titulaire des droits d'auteur sur le logiciel peut transférer au licencié, utilisateur, en tout ou partie, à titre exclusif ou non exclusif, ses droits patrimoniaux et plus spécifiquement, le droit de reproduction et les droits de traduction (dans un autre langage informatique), d'adaptation, d'arrangement ou de modification. Seul l'auteur – son employeur en cas de création salariale – peut décider des droits qu'il concède aux tiers (ses clients) par démembrement de son monopole d'exploitation. Dans cette optique, tout droit qui n'est pas expressément concédé demeure strictement réservé au titulaire des droits.

Ne pouvant céder plus de droits qu'elle n'en dispose, la société doit donc s'être assurée :

- que ses salariés ont développé le code source dans l'exercice de leur fonction (*cf. supra*) ;
- à défaut, acquérir auprès de ses salariés la propriété intellectuelle de leurs travaux (voir clauses spécifiques, *infra*) ;
- que le développement du logiciel n'a pas impliqué l'insertion de code source appartenant à un tiers (ou ajouter les stipulations exactes des licences *Open Source* si celles-ci existent) ;
- que la cession des droits par voie de licence est formalisée, en cas de recours à des tiers dans le développement du logiciel.

Les licences *Open Source* fonctionnent fondamentalement comme les licences classiques et, juridiquement, il n'y a qu'une différence de degré entre la cession sous licence *Open Source* d'un logiciel et la simple licence d'utilisation : la liberté de l'utilisateur est plus ou moins grande, en fonction des droits que lui a reconnus son auteur à travers la licence.

0.2 LE LOGICIEL LIBRE, UNE FIGURE ORIGINALE

Depuis quelques années, le monde économique a vu s'affirmer une figure originale : le « *Logiciel Libre* ». Cette figure, dont il convient de retracer un historique rapide, a fait l'objet de nombreuses confusions terminologiques et conceptuelles avant d'apparaître clairement pour ce qu'il est : un mode alternatif de conception et d'exploitation de logiciels.

Un Logiciel Libre est un logiciel dont l'utilisation, l'étude, la modification, la duplication et la diffusion sont universellement autorisées sans contrepartie²⁶. Par opposition, et c'est la raison de l'appellation, un logiciel est dit propriétaire²⁷ lorsqu'il reste la propriété d'une seule personne qui n'autorise que limitativement les usages sur le logiciel. Dans ce cas, même s'il peut être gratuit, l'éditeur garde généralement la maîtrise de son logiciel, et notamment du code source qu'il maintient secret.

D'un point de vue technique, un logiciel est dit libre lorsqu'il est utilisable et modifiable sans limitation et qu'il est fourni avec toutes les informations utiles à cette fin (code source documenté et lisible, scripts d'installation, documentation, etc.).

D'un point de vue juridique, c'est un logiciel pour lequel l'auteur entend partager son monopole, avec ou sans condition de réciprocité, afin de favoriser sa diffusion et sa réutilisation par d'autres. À cette fin, une licence *Open Source* accompagne le logiciel afin d'assurer aux détenteurs d'une copie du logiciel les droits de le copier, l'utiliser, le modifier et le distribuer.

C'est aussi un usage « à rebours »²⁸ des droits de propriété intellectuelle, puisque tout ce qui n'est pas expressément cédé demeure sous la stricte exclusivité du titulaire des droits²⁹. L'écosystème de l'*Open Source* procède à une « automatisation du partage des monopoles » et considère par principe que l'exploitation du logiciel est permise, à condition de respecter les limitations clairement énumérées par les licences (logique « partage sauf exception »).

²⁵ La pratique existant notamment dans les sociétés qui abandonnent ce droit au profit de leurs employés au cas par cas dans le cas du développement d'un logiciel libre. Hewlett Packard en est un exemple.

²⁶ Source Wikipedia http://fr.wikipedia.org/wiki/Logiciel_libre

²⁷ Voir notamment http://fr.wikipedia.org/wiki/Logiciel_propri%C3%A9taire

²⁸ Voir Benjamin JEAN & Sébastien CANEVET, « L'évolution du droit d'auteur à l'ère du numérique », contribution à l'ouvrage "La bataille Hadopi", ed. In Libro Veritas, 2009.

Logiciel Libre et « propriétaire » reposent tous deux sur le droit d'auteur, mais là où toute prérogative n'est concédée, sur un logiciel « propriétaire », qu'en dérogation à un monopole personnel d'exploitation, le logiciel libre inverse la logique et consacre le partage comme principe.

Les Logiciels Libres se distinguent du **domaine public**³⁰ en ce que leur auteur conserve encore un contrôle entier sur l'exploitation de ces derniers. Au surplus, c'est précisément parce que l'auteur (développeur ou éditeur) dispose sur sa création de l'intégralité des droits de propriété intellectuelle qu'il est susceptible de placer ensuite sa création sous une licence *Open Source*, et par conséquent de faire de son logiciel un Logiciel Libre. Par conséquent, le non-respect d'une licence *Open Source* est, très logiquement, sanctionné sur le fondement de la contrefaçon³¹.

Enfin, un logiciel libre n'est pas non plus un **logiciel gratuit**³² : De nombreuses exploitations commerciales de Logiciels Libres existent³³, et c'est justement cette liberté offerte à tous de vendre le logiciel qui tend à réduire le prix de vente à la seule valeur ajoutée. De façon plus pragmatique, un Logiciel Libre n'est gratuit que lorsqu'il a été payé ou financé (par un investissement en recherche de l'éditeur ou par des contrats externes), mais tout ce qui a déjà été payé ne le sera plus ensuite (forte incitation à l'innovation).

0.2.1 ORIGINE HISTORIQUE DU LOGICIEL LIBRE ET OPEN SOURCE

Précurseur en la matière, Richard Matthew STALLMAN est le fondateur du mouvement pour le Logiciel Libre. Il a fondé, en 1984, la *Free Software Foundation* (FSF), organisation américaine à but non lucratif, pour aider au financement du projet GNU (acronyme récuratif pour GNU's Not Unix) et de la communauté du Logiciel Libre. La FSF a notamment en charge, avec l'assistance du *Software Freedom Law Center* (SFLC) dirigé par l'avocat Eben MOGLEN, la rédaction des différentes licences « GNU » : GNU *General Public License* (GPL), GNU *Lesser General Public License* (LGPL), GNU *Affero General Public License* (GNU AGPL) et GNU *Free Documentation License* (GNU FDL).

L'*Open Source Initiative*³⁴ (OSI) est née d'une scission avec la FSF, afin de labelliser les licences réunissant les critères de la « définition *Open Source*³⁵ ». Centrée sur les méthodes de développement de logiciel à code ouvert où le service reprend le pas sur le produit lui-même, l'OSI a rédigé une définition de l'*Open Source* (l'*Open Source* Définition³⁶). Le site regroupe actuellement plus de 66 licences labellisées Open Source selon un processus bien établi.

La FSF promeut une certaine liberté des utilisateurs de logiciels. L'OSI considère comme optimal ce type de développement de logiciels *Open Source* : les outils sont les mêmes, pas les finalités.

0.2.2 QUAND PARLE-T-ON DE LOGICIEL LIBRE ET D'OPEN SOURCE ?

Dans les faits, la définition de la FSF (portant sur les Logiciels Libres) et celle de l'OSI (portant sur les Licences *Open Source*) se rejoignent et les deux notions sont souvent assimilées sous le vocable *Open Source* alors qu'elles correspondent à deux définitions complémentaires :

²⁹ Sorte de logique de « réservation sauf exception ». L'article L.131-3 du Code de la propriété intellectuelle dispose à ce titre que : « La transmission des droits de l'auteur est subordonnée à la condition que chacun des droits cédés fasse l'objet d'une mention distincte dans l'acte de cession et que le domaine d'exploitation des droits cédés soit délimité quant à son étendue et à sa destination, quant au lieu et quant à la durée.

Lorsque des circonstances spéciales l'exigent, le contrat peut être valablement conclu par échange de télégrammes, à condition que le domaine d'exploitation des droits cédés soit délimité conformément aux termes du premier alinéa du présent article.

Les cessions portant sur les droits d'adaptation audiovisuelle doivent faire l'objet d'un contrat écrit sur un document distinct du contrat relatif à l'édition proprement dite de l'œuvre imprimée.

Le bénéficiaire de la cession s'engage par ce contrat à rechercher une exploitation du droit cédé conformément aux usages de la profession et à verser à l'auteur, en cas d'adaptation, une rémunération proportionnelle aux recettes perçues. »

³⁰ Par ailleurs, et contrairement à d'autres droits nationaux, il est impossible en France de renoncer à ses droits pour verser son œuvre au profit du domaine public avant l'écoulement du délai de sa protection.

³¹ Pas moins d'une quinzaine d'assignations ont eu pour origine le non respect de la GNU GPL V.2 sur le logiciel Busybox en 2009.

³² On confond souvent Logiciel Libre et logiciel gratuit, aussi appelé freeware ou gratuiciel.

³³ Par exemple le système d'exploitation Red Hat.

³⁴ Fondée par Todd ANDERSON, Chris PETERSON, John MADDOG HALL, Larry AUGUSTIN, Sam HOCKMAN et Eric S. RAYMOND, Site Internet : <http://opensource.org/>.

³⁵ <http://opensource.org/docs/definition.php>.

³⁶ Disponible sur le site de l'OSI : <http://www.opensource.org/docs/osd>

Selon la **FSF**, le Logiciel Libre se fonde sur quatre critères (ou « libertés fondamentales ») :

1. La liberté d'utiliser/d'exécuter un programme pour tous les usages (liberté 0),
2. La liberté d'étudier le fonctionnement interne d'un programme et de l'adapter à ses besoins (liberté 1),
3. La liberté de diffuser des copies du programme (liberté 2),
4. La liberté de modifier le programme afin de l'améliorer et de redistribuer ensuite la version modifiée (liberté 3).

Cela ne signifie pas que le logiciel soit forcément « **gratuit** », mais que l'utilisateur pourra bénéficier de ces libertés et que son code source sera librement distribué, sans restriction d'ordre technique ou juridique.

Selon l'**Open Source Definition**³⁷, est *Open Source* une licence qui vérifie les critères suivants :

1. La libre redistribution du logiciel (sans coût de licence donc) ;
2. Le code source doit être fourni ou être accessible ;
3. Les dérivés des œuvres doivent être permis ;
4. L'intégrité du code doit être préservée (un tiers ne peut pas s'approprier le travail d'un autre et les contributions de chacun sont clairement attribuées) ;
5. Pas de discrimination entre les personnes ou les groupes ;
6. Pas de discrimination entre les domaines d'application (la licence ne doit pas être un instrument politique) ;
7. La licence s'applique sans dépendre d'autres contrats (elle ne peut, par exemple, être liée à un accord de non-divulgaration) ;
8. La licence ne doit pas être propre à un produit (elle est attachée au code, non au logiciel en tant que produit ou projet) ;
9. La licence d'un logiciel ne doit pas s'étendre à un autre³⁸ ;
10. La licence doit être neutre technologiquement.

Par souci de précision, nous nous référons dans ce document à la définition de l'OSI lorsque nous parlerons des licences (« *Open Source* ») et à la définition de la FSF lorsque nous évoquerons les logiciels (« *Libres* »).

0.2.3 LA PLACE DES LICENCES OPEN SOURCE DANS NOTRE DROIT POSITIF³⁹

Les licences *Open Source* sont – la qualification juridique est unanimement partagée – des contrats gracieux de cession non exclusive de droits d'auteur⁴⁰. Le caractère gratuit – non pas dans la mise à disposition, mais dans la cession de droits – et l'étendue de la cession constituent les principales distinctions entre Logiciel Libre et propriétaire.

Sur le fond, la différence est plus profonde. La logique « propriétaire » repose sur un éditeur définissant de façon restrictive les droits concédés à un licencié, en fonction de ses besoins, pour une durée parfois déterminée et une utilisation restreinte. A l'opposé, la logique *Open Source* repose sur le partage du monopole. Le code source est ouvert et disponible, les droits sont systématiquement concédés de façon très large, la diffusion du logiciel est non discriminante, pour la durée des droits qui s'y applique et le monde entier. Néanmoins, le caractère « *libre* » des licences ne signifie par pour autant qu'elles ne comportent pas de contraintes. On verra en particulier que la famille la plus connue des licences *Open Source* comporte au contraire une obligation forte en terme de réciprocité : la clause de *copyleft*.

³⁷ Voir : <http://www.opensource.org/docs/definition.php>

³⁸ L'étendue large de la GNU GPL est conforme à ce critère en ce qu'elle force l'extension au logiciel « comme un tout ».

³⁹ Voir notamment Benjamin JEAN, « Option Libre, Utiliser les licences Open Source en connaissance de cause », , ed. In Libro Veritas, Collection Framabook, à paraître (second trimestre 2010)

⁴⁰ À ce sujet, voir les travaux de Mélanie CLEMENT-FONTAINE et notamment « Les licences Open Source sont des licences par lesquelles l'auteur autorise la copie, la modification et la diffusion de l'œuvre modifiée ou non, de façon concurrente, sans transférer les droits d'auteur qui y sont attachés et sans que l'utilisateur ne puisse réduire ces libertés tant à l'égard de l'œuvre originale que de ses dérivés » M. Clément-Fontaine, « Les œuvres libres », Thèse Montpellier, 2006, n° 134.

Concernant le droit applicable et la juridiction compétente, il suffit, comme dans tout contrat international, de se référer au Droit International privé en l'absence de clause expresse dans les licences elles-mêmes⁴¹ – quelques licences apportant en effet ces précisions⁴² ou renvoyant expressément aux règles de droit international privé⁴³.

Enfin, la majorité des licences *Open Source* contiennent aujourd'hui des dispositifs qui viennent limiter le risque d'éviction lié au brevet logiciel⁴⁴ (en ajoutant à la cession de droits d'auteur, une **renonciation** aux brevets du concédant – ou une cession de droits sur ledit brevet). Parallèlement, les principaux acteurs américains de l'*Open Source* opèrent des regroupements de brevets à titre défensif : initiatives individuelles⁴⁵ ou collectives⁴⁶, elles font directement face aux *patents trolls*⁴⁷.

1. IMPACTS DE L'OPEN SOURCE SUR LA CONDUITE DE PROJETS INFORMATIQUES

L'*Open Source* peut permettre un développement accéléré et efficace donnant aux entreprises une base de codes éprouvée et maintenue, sans commune mesure avec leurs propres ressources. Néanmoins, l'inclusion de code tiers requiert une vigilance accrue : d'une part, quant au respect d'un certain formalisme (1.1) et, d'autre part, quant à la conservation d'un certain nombre d'informations (1.2). Ainsi, le corollaire d'une plus grande liberté est celui d'une plus grande responsabilité.

De nombreux critères permettront de faire le choix d'un Logiciel Libre, parmi lesquels : son adéquation avec les enjeux industriels, les besoins fonctionnels et techniques, la maturité et pérennité du code, la qualité du code, sa prise en main, sa feuille de route, le respect des standards et son interopérabilité, et enfin le dynamisme de sa communauté et sa licence.

Cette partie a pour objectif de montrer, service par service, l'influence et les changements induits par le recours aux licences *Open Source*.

1.1 PRISE EN COMPTE DE LA DIMENSION CONTRACTUELLE : PROCESSUS D'ANALYSE DES LICENCES

Afin de maîtriser les enjeux liés au respect des licences, c'est-à-dire veiller à ce que la redistribution des résultats se fasse conformément aux termes des licences attachées à chacun des composants distribués, il est nécessaire de mettre en place une procédure adaptée dès la conception du logiciel. Celle-ci doit favoriser une interaction forte entre les équipes juridiques et techniques.

En effet, si la licence attachée aux développements spécifiques conçus dans le cadre d'un projet peut être sélectionnée de façon discrétionnaire, cette liberté est fortement limitée lorsque ces développements sont distribués en combinaison avec une multitude d'autres composants logiciels, tous ayant leur usage conditionné au respect de leur licence.

Ainsi, chacune des licences attachées aux composants du livrable doit être minutieusement analysée afin d'appréhender très justement sa portée et ses obligations : ce qui amène notamment à distinguer les licences permissives - seules les obligations doivent être transmises par les licences postérieures - de celles dites *copyleft* pour lesquelles tant les obligations que les droits doivent s'y retrouver⁴⁸.

⁴¹ De nombreuses licences prévoient de telles clauses : comme la Mozilla Public licence ou l'Eclipse Public License

⁴² À l'instar de la Mozilla Public licence qui privilégie la juridiction et la Loi Californienne.

⁴³ Telle l'OSL ou l'EUPL.

⁴⁴ Le groupe thématique Logiciel Libre mis en place par le pôle de compétitivité SystemaTic a rédigé une rapide Charte d'engagement de ses membres vis-à-vis des brevets logiciels consultable sur le site <http://www.systematic-paris-region.org/fr/index.html>

⁴⁵ Plusieurs milliers de brevets sont ainsi regroupés en garantie par des sociétés comme IBM, Nokia, Sun, etc. – notamment l'« IBM Statement of Non-Assertion of Named Patents Against OSS », le « Sun patent program » ou encore le « Novell Statement on Patents and Open Source Software ».

⁴⁶ Non limitativement : l'Open Source Development Laboratory (OSDL), « no software patents », l'Electronic Frontier Foundation (EFF – avec l'initiative Patent busting project), la Foundation for a Free Information Infrastructure (FFII) ou l'Open Invention Network (OIN) auquel ont souscrit des sociétés comme Sony, IBM, NEC, Red Hat, Philips et Novell.

⁴⁷ L'accent peut notamment être mis sur Linux Defenders. Lancée le jeudi 11 décembre 2008 par l'OIN, l'initiative permettra de protéger les communautés de logiciels Open Source contre les menaces récurrentes en matière de brevets. Elle collecte ainsi au travers le monde et auprès des développeurs indépendants de logiciels libre / Open Source un maximum d'innovations susceptibles d'être brevetées afin d'en réaliser une publication défensive et de les verser à l'état de l'art.

⁴⁸ Il convient dans les faits de procéder à une analyse fine des différents termes des licences afin d'apprécier in fine la compatibilité de la licence utilisée lors de la redistribution (sur ce point, cf. infra Partie 3).

Ainsi, pour rester en cohérence avec certaines de ses obligations de licenciée, la société sera amenée à proposer une licence qui prendra la forme soit d'une Licence *Open Source* unique, soit d'une combinaison de licences *Open Source*.

Deux postulats gouvernent le choix de la ou des licences régissant le statut du produit final⁴⁹ :

- être en mesure de céder effectivement tous les droits que la licence finale confère aux licenciés subséquents ;
- ne pas obliger moins ces derniers que ce à quoi l'on est soi-même obligé.

1.1.1 TENUE D'UN INVENTAIRE

Tenir un état des lieux : exercice qui ne se réussit que dans la constance et la rigueur. La tenue d'un inventaire clair et précis des composants logiciels utilisés est une nécessité et nulle société ne peut en faire l'économie – ne serait-ce que parce qu'il s'agit bien souvent de l'objet même du contrat.

Simple formalisme lorsqu'il est intégré dans un processus de réutilisation de code tiers – ou véritable accouchement lorsqu'il est réalisé peu de temps avant la distribution, il convient de mettre en place un processus qui soit fluide et permette aux équipes juridiques et techniques d'interagir rapidement. Idéalement, il sera fluidifié par la mise en place d'un portail dédié qui permettra :

- d'encadrer et de guider la reprise de composants *Open Source* (et donc d'éviter toute reprise non contrôlée),
- d'unifier les versions des logiciels utilisés,
- et de faciliter leur prise en main.

La maîtrise de l'usage de l'*Open Source* (par exemple en fonction du type ou du domaine d'application) pourra aussi être envisagée au sein de la politique *Open Source* de la société (qu'elle soit éditrice, intégratrice ou simple utilisatrice de logiciel).

Préalable indispensable, c'est sur cette liste de composants et de licences que l'équipe juridique pourra ensuite se prononcer (la plupart du temps moyennant un certain nombre d'aller-retour pour obtenir les précisions nécessaires).

1.1.1.1 RÔLE DE L'ÉQUIPE TECHNIQUE DE DÉVELOPPEMENT OU DE CONCEPTION

Les processus varient et dépendent fréquemment de ceux déjà en place au sein de chaque structure, mais deux documents techniques (au moins) sont nécessaires :

• Le premier pour réunir toutes les informations sur les composants inclus dans la solution – qu'il s'agisse de développements spécifiques ou non - :

- informations relatives aux composants (nom, URL d'origine),
- à leur licence (nom, version, URL, éventuellement les termes additionnels qui conditionnent son usage),
- et à leurs auteurs (nom, informations permettant de les contacter et éventuellement société).

• Le second décrivant précisément toutes les interactions entre ces composants (il prendra généralement la forme d'un ou plusieurs schémas et/ou d'une présentation écrite/orale). Les différents types de relations seront bien renseignés :

- relation explicite (l'appel du composant B est codé dans le composant A),
- relation implicite (l'appel du composant B n'est pas codé en tant que tel dans le composant A, mais le composant A appelle d'autres composants par paramétrage et le projet B fait partie de ces composants),
- relation possible (l'utilisateur de l'application fournie par le projet peut paramétrer lui même des relations entre les composants et un appel de B par A fait partie de cette éventualité).

⁴⁹ In fine la compatibilité de la licence utilisée lors de la redistribution (sur ce point, cf. infra Partie 3).

Cette liste doit être mise à jour au fur et à mesure et il peut être nécessaire de conserver un historique de ce fichier. Il doit être à la disposition du service juridique et sera *in fine* annexé (de préférence en l'état) au contrat.

1.1.1.2 RÔLE DE L'ÉQUIPE JURIDIQUE

Se basant sur les documents élaborés par l'équipe technique, l'équipe juridique va réaliser un audit de l'ensemble des composants listés et décrits afin de déterminer :

- si ceux-ci sont propriété (ou copropriété) de la société ou de ses partenaires,
- si les licences utilisées sur les composants répondent à la politique de la société et/ou du projet.

En cas d'inadéquation, le problème sera relevé et transmis à l'équipe technique.

Les risques juridiques liés à une contestation de l'utilisation faite de composants par rapport à leur(s) licence(s) sont, par nature, à apprécier par les responsables de l'entreprise susceptibles d'être mis en cause. En conséquence, il est nécessaire que l'entreprise ait défini une politique en matière de licences, avant même qu'un premier projet ne soit réalisé. À défaut, il est prudent pour les équipes de développement de proposer et de faire valider a priori par les services juridiques une telle politique (qui peut limiter l'usage de certains composants logiciels, de certaines licences *Open Source*, ou mettre en place un processus particulier).

Par ailleurs, chaque licence présente dans le code fera l'objet d'une fiche synthétique permettant de faciliter la comparaison et d'apprécier leur portée respective⁵⁰.

1.1.2 CHOIX DE LA LICENCE OPEN SOURCE

Dans la mesure du possible, les équipes veillent à ce que les licences utilisées sur les composants développés spécifiquement permettent une reprise et une réutilisation de ces dernières. De plus, elles doivent être adaptées au type du logiciel, aux autres logiciels de la société, à l'écosystème dans ce domaine, etc.

Afin de faciliter le choix, une politique claire en matière de licence doit être décidée, rédigée et respectée. Le but ici est la livraison d'un logiciel « sécurisé », c'est-à-dire respectueux des licences applicables aux composants logiciels qui le composent – étant entendu que, bien souvent, la rédaction de cette politique traduira l'acceptation d'un niveau de risque juridique.

Par ailleurs, dans un objectif de simplicité, les logiciels seront distribués en limitant au strict minimum le nombre de licences différentes utilisées (Cf. *infra* Partie 3 décrivant les exceptions et interprétations qui permettent de réutiliser des licences tout en les adaptant aux besoins spécifiques).

1.2 PRISE EN COMPTE DE LA DIMENSION TECHNIQUE : RAPPELS SUR LES LIVRABLES D'UN PROJET INFORMATIQUE

1.2.1 DESCRIPTION DU LIVRABLE

Les livrables d'un projet s'entendent de ce qui est défini comme tel dans le contrat. Rien ne doit être implicite. Le livrable peut très utilement être décrit dans un document technique dit de « fourniture logicielle ».

Que l'on soit dans un schéma de distribution de Logiciels Libres ou non, un certain nombre de bons comportements, de « Bonnes Pratiques » doit être retenu⁵¹. Il est ainsi notamment nécessaire de :

- **Définir** le contenu et le périmètre technique : ce qui correspond généralement à l'objet principal du contrat de fourniture de logiciel (nom et descriptif fonctionnel sommaire de la livraison, pré-requis de déploiement, liste détaillée de la fourniture logicielle et des documentations qui l'accompagnent, procédures de test et de validation, procédures de recette, etc.).
- **Justifier** clairement des licences et des noms des titulaires de droits sur les composants qui le constituent ou qui ont participé à sa constitution (Cf. *supra* concernant l'établissement de ladite liste).

⁵⁰ Un exemple de fiche générique figure en annexe 2

⁵¹ Voir en annexe 2, exemple d'« annexe contractuelle de description d'une fourniture logicielle ». La liste d'informations donnée n'est pas exhaustive du point de vue de la livraison d'un logiciel puisqu'elle ne prend pas en compte les contextes projets spécifiques (ex : liste des exigences couvertes, liste des anomalies corrigées...). Pour autant, les informations décrites permettent de délimiter précisément de quoi le logiciel est constitué et quels sont les droits qui lui sont attachés.

- **Définir** clairement d'éventuelles limites de responsabilité sur des composants externes non conçus par le prestataire et livrés par le prestataire et/ou requis pour le déploiement. Il faut notamment décrire le niveau de version et/ou l'état technique sur lequel le prestataire aura validé la compatibilité ou le bon fonctionnement de ces composants externes choisis ou imposés par le client/l'intégrateur.

- **Servir**, le cas échéant, de document de référence en cas d'arbitrage ou de litige, que celui-ci porte sur des aspects de responsabilité civile ou contractuelle, de propriété, de contrôle par une autorité de tutelle, etc.⁵².

Ce document pourra être une annexe du contrat (de licence ou de fourniture) ou du procès-verbal de recette définitive⁵³. Enfin, ce document pourra aussi servir de description de référence dans le cadre d'une réponse à appel d'offres, affichant clairement la liste des composants externes et les conditions de leur réutilisation.

1.2.2 DESCRIPTION DU CODE SOURCE

Les avantages techniques et les aspects juridiques du Logiciel Libre doivent être appréhendés dès la phase de conception. Deux recommandations peuvent ainsi être formulées concernant la « rédaction » de code :

- **Veille quant à l'état de l'art.** Avant de développer un composant, il est nécessaire d'établir un état de l'art et rechercher s'il n'existe pas un logiciel ou une suite de logiciels libres appropriés. Cette pratique permet un gain en temps et en effort de développement. Il est en général moins coûteux de chercher et d'adapter un composant existant et éprouvé que de le développer en partant de rien. Les Logiciels Libres sont souvent les solutions les plus adaptées (absence de frais de licence, possibilité de les adapter, support des communautés, etc.), à condition de réaliser le travail de renseignement, de validation des licences, et de songer au caractère du projet (communautaire, attaché à un éditeur, etc.) pour prévoir les pistes d'évolutions possibles.

- **Détection de codes.** Parallèlement à un l'état de l'art, il faut que les chefs d'équipe s'assurent de l'absence de copie de code et donc de contrefaçon (codes sans indication des sources, absence de licences, etc.). Ainsi, les entreprises peuvent recourir à des outils de détection afin de faciliter, automatiser et fluidifier le travail d'audit de code. Il existe des outils qui permettent d'assister tout ou partie de ces tâches : par exemple des logiciels de détection de code source ou de licences. Ces produits (et services) aident les entreprises à comprendre la composition de leurs logiciels et à connaître l'origine de chacun des composants. Ces logiciels peuvent rapidement avoir une place prédominante dans la mise en place d'une gouvernance⁵⁴, et ils permettent notamment de gérer la réutilisation des composants tout au long du cycle de développement logiciel en toute sûreté tout en respectant les obligations des licences et en minimisant les risques.

Les principaux logiciels permettant d'auditer un code sont :

- **BlackDuck** : Solution logicielle de gestion de la conformité et de la sécurité permettant la maîtrise des risques généralement associés à l'*Open Source* en automatisant les processus liés à la gestion du code source tout au long du cycle de vie du développement des applications – recherche, sélection, approbation, validation et surveillance⁵⁵.
- **FOSSology** (FOSS pour Free and Open Source Software) : Outil d'analyse de code logiciel construit autour d'une architecture ouverte et modulaire. Les modules inclus permettent l'analyse des licences, l'extraction de méta données, l'identification du type MIME (Multipurpose Internet Mail Extensions). Il dispose d'une large communauté et provient d'un développement interne rendu libre sous licence GPLv2 par la société HP et continue de bénéficier de son soutien pour le développement de futures versions⁵⁶.
- **Palamida** : Solution industrielle de sécurité dédiée au logiciel sous licence *Open Source*, elle permet d'identifier rapidement et de suivre du code non documenté, les vulnérabilités associées, ainsi que les problèmes de propriété intellectuelle ou de conformité⁵⁷.
- **OSLC** (Open Source License Checker) : outil *Open Source* (GNU GPL) d'inspection et d'analyse des licences de packages *Open Source*. OSLC analyse les informations de licences par l'extraction des informations de licence de tous les codes sources, puis il effectue une comparaison avec le texte de référence archivé dans la base et finalement résume toutes les informations de licence du paquetage⁵⁸.

⁵² Il pourra en effet être utilisé par des arbitres, des experts ou toute personne ayant à en connaître ou compétente pour en traiter.

⁵³ Pour la livraison initiale et pour toute livraison de version majeure.

⁵⁴ Ce n'est d'ailleurs pas une surprise si la société Hewlett-Packard a développé son propre outil de détection pour maîtriser le risque d'utilisation involontaire d'un composant open source, allant même jusqu'à en faire un logiciel libre : Fossology.

⁵⁵ Voir : <http://www.blackducksoftware.com/>

⁵⁶ Voir : <http://fossology.org/>

⁵⁷ Voir : <http://www.palamida.com/>

⁵⁸ Voir : <http://forge.ow2.org/projects/oslc3/>

- **Protecode** : Outil permettant de gérer le développement de logiciel en temps réel grâce à la mise en place d'une gouvernance efficace et la rationalisation de l'actif de code source, afin d'augmenter la valeur de l'entreprise et réduire les coûts⁵⁹.

2. IMPACTS DE L'OPEN SOURCE SUR LA GOUVERNANCE DE L'ENTREPRISE

Ce chapitre décrit de manière globale les *business model Open Source* et leur impact sur l'écosystème de l'entreprise.

Après avoir défini son *business model* (2.1.), l'entreprise affinera son discours pour répondre aux principales préoccupations du client (2.2.).

Les bonnes pratiques organisationnelles internes aux entreprises, liées à la réalisation et/ou l'utilisation des solutions informatiques comprenant des composants libres, feront l'objet d'un développement (2.3.).

2.1 IMPACTS SUR LA POLITIQUE COMMERCIALE ET ÉDITORIALE DE L'ENTREPRISE

Le logiciel libre bouleverse les conceptions classiques du modèle industriel des éditeurs. Ainsi la gestion de la propriété intellectuelle ne doit plus être pensée uniquement en termes d'interdiction, mais aussi en termes d'autorisation – la gageure étant pour les sociétés d'ouvrir tout en conservant le contrôle.

Dès lors que l'éditeur ouvre et diffuse le code source de ses produits gratuitement, il ne peut plus compter sur les redevances de licences pour rentabiliser les travaux de Recherche & Développement effectués en amont. En outre, et à l'évidence, l'éditeur perd l'avantage concurrentiel que constituent la confidentialité et le secret d'affaires.

Il doit affiner l'usage qu'il entend faire des droits de propriété intellectuelle et prendre l'avantage sur d'autres axes de compétition, ce qui revient à modifier son « *business model* ». Cette évolution se traduit par la mise en place d'une politique en matière de propriété intellectuelle (permettant de dissocier, mais aussi de rendre cohérente, la gestion des différents droits de propriété intellectuelle : droit des marques, brevets, droits d'auteur, etc.).

L'*Open Source* n'est donc pas un *business model* en soi, mais une combinaison de différents modèles existant.

2.1.1 LE MODÈLE « DISTRIBUTEUR À VALEUR AJOUTÉE »

Les Logiciels Libres sont généralement des applications fortement paramétrables, auxquelles sont associés des modules librement réutilisables, modifiables, personnalisables, en fonction des besoins de chaque utilisateur. Ce caractère « protéiforme » des applications proposées et des licences liées engendre une valorisation inéluctable des services associés, tendance encore renforcée par l'amélioration des outils (gestion de projets, méthodes de collaboration innovantes entre le client et le prestataire, supervision et pilotage, etc.). Cela conduit à concilier innovation, standardisation, distribution et personnalisation spécifique. D'où la spécialisation progressive des sociétés du secteur en faveur de modèles hybrides d'édition de logiciel, de distribution et d'intégration spécifique, très orientés vers une économie de services.

Lors de l'achat de logiciels propriétaires, le client paie généralement un premier prix pour utiliser le logiciel, un second prix pour les services de maintenance (pour une durée déterminée et renouvelable) ainsi qu'un troisième prix pour le paramétrage du logiciel.

Parallèlement, lors de l'acquisition de Logiciels Libres auprès d'un distributeur, l'utilisateur a souvent le choix entre télécharger son produit ou acheter un support physique (incluant le média et la documentation). C'est alors un abonnement à une prestation récurrente de maintenance et d'assistance qui permet d'accéder au système automatique de mises à jour, ainsi qu'aux services variés tels que l'assistance technique ou la formation. L'utilisateur est donc libre de consommer les services proposés, selon ses besoins, auprès du prestataire le plus compétitif. Les sociétés génèrent donc leurs revenus à partir de la vente des supports physiques du logiciel, et/ou surtout de la commercialisation de services à valeur ajoutée tels que formation, support technique et assistance⁶⁰.

⁵⁹ Voir : <http://www.protecode.com/>

⁶⁰ Différentes formes de supports existent, notamment des services équivalents à ceux offerts par un éditeur classique. Voir 3.2.4 Garantie contractuelle et Maintenance.

2.1.2 LE MODÈLE DE LA LICENCE DOUBLE ET/OU DÉCALÉE

Cette pratique, de plus en plus répandue, consiste à exploiter un même logiciel sous deux types de licences (il est donc possible d'en jouir selon l'une ou l'autre des licences) et de proposer diverses options et services connexes. Ce modèle est adopté à la fois par des éditeurs classiques qui révolutionnent leur modèle économique (Ingres, Sun) et des éditeurs 100% *Open Source* (MySQL, Linagora, TrollTech, etc.).

De plus en plus d'éditeurs adoptent une politique de doubles licences (*ou dual licensing*). Le but est double :

- exploiter pleinement un nouveau statut d'éditeur logiciel *Open Source* et,
- retirer des ressources de l'exploitation commerciale des produits tout en remplaçant activités de maintenance et de développement.

Les éditeurs proposent de publier pour la Communauté une version standardisée et stabilisée de leur offre, afin de prendre position sur leur marché, et continuent d'investir sur de nouvelles versions vendues sous forme d'offres additionnelles ou de maintenance autour de leur noyau de base. En parallèle, ces éditeurs proposent une offre supérieure ou équivalente⁶¹ en fonctionnalités aux clients demandeurs, sur la base d'une tarification traditionnelle, selon une autre licence (mixte ou propriétaire). Le principal effet vertueux de ce modèle consiste à imposer peu à peu sur le marché un standard technique par ouverture systématique de son code source, ce qui augmente l'accessibilité du client et le prototypage potentiel. La rémunération de l'éditeur est acquise via les services connexes et celui-ci assure le pilotage du projet *Open Source*.

Les éditeurs et prestataires les plus connus sur le marché des logiciels libres ont d'ores et déjà adopté une politique de *dual licensing* :

- Talend⁶², éditeur de solutions d'intégration de données (dual License) ;
- Oracle, via ses rachats successifs ;
- Berkeley DB⁶³, base de données sous licence de type permissif ;
- MySQL⁶⁴, éditeur de gestionnaires de bases de données (dual License) ;
- Sun Microsystems⁶⁵, précurseur dans le développement du logiciel libre (le langage informatique Java est sous licence *Open Source*, ainsi que son système d'exploitation OpenSolaris), est également à l'origine du projet OpenOffice.org, qui équipe aujourd'hui les principales administrations françaises. Sun a racheté MySQL et s'est récemment fait racheter par Oracle ;
- Linagora⁶⁶, qui édite de multiples logiciels *Open Source* : OBM, LinPKI, LinShare et LinID ;
- Red Hat⁶⁷, précurseur en matière de services informatiques basés sur l'*Open Source*, avec par exemple sa solution JBoss JTS disponible sous licence *Open Source* et sous licence propriétaire.
- Nokia, au travers de son rachat de la société Trolltech, éditeur de la bibliothèque Qt, diffuse sa solution sous triple licence GNU GPL, GNU LGPL et propriétaire (cette dernière solution permettant de ne pas fournir le code source sur les modifications, de créer des applications propriétaires, d'avoir une mise à jour et un support préférentiels, etc.)⁶⁸.

Ces éditeurs distinguent une version « *communautaire* », soumise à une licence *Open Source* et une version « *entreprise* », soumise à une licence propriétaire classique – les deux versions étant généralement identiques. **Mais attention seul un éditeur qui détient l'intégralité des droits de propriété intellectuelle sur son produit est en mesure d'adopter ainsi des licences distributives sur les produits qu'il développe.**

Une variante (le modèle à licence *décalée*) consiste pour l'éditeur, sur un principe équivalent, à ne facturer que les versions récentes de ses logiciels (6 à 12 mois) et ensuite à les publier sous licence *Open Source* dès commerciali-

⁶¹ On retrouve notamment cet exemple avec MySQL: la même version est disponible sous les deux licences, mais le choix de la licence commerciale assure une plus grande souplesse pour celui qui souhaite intégrer intimement MySQL dans son produit avec une licence propriétaire.

⁶² Voir : <http://fr.talend.com/products-data-integration/matrix.php>

⁶³ Voir : <http://www.oracle.com/technology/software/products/berkeley-db/htdocs/oslicense.html>

⁶⁴ Voir : <http://dev.mysql.com/downloads/>

⁶⁵ Voir : <http://fr.sun.com/practice/software/solaris/get.jsp>

⁶⁶ Voir : <http://obm.org/doku.php>

⁶⁷ Voir : <http://www.redhat.com/software/rhelorfedora/>

⁶⁸ Voir : <http://qt.nokia.com/products/licensing>

sation d'une nouvelle version majeure. Le client standard de l'éditeur ne paie ainsi que pour l'innovation fonctionnelle et technique, et éventuellement pour le support, tandis que l'éditeur oriente son modèle économique vers davantage de services (maintenance logicielle, support revendeur, assistance utilisateur) et de vente de modules additionnels⁶⁹.

2.1.3 L'OPEN CLOUD COMPUTING⁷⁰

Le modèle SaaS (*Software as a Service*) existe depuis plusieurs années maintenant et le contexte économique actuel⁷¹ est particulièrement favorable à son développement – voire à son extension. Derrière les différents termes couvrant le *Cloud Computing* (*Infrastructure as a Service*, *Platform as a Service*, *Software as a service*, *Pay per use*, *On demand*, etc.), il y a une idée commune, un concept fédérateur : celui de ressources informatiques disponibles à la demande et facturées uniquement en fonction de leur utilisation. Ces ressources peuvent être des services, des applications, des plates-formes applicatives permettant d'héberger les applications des entreprises, du stockage, de la CPU ou de l'infrastructure réseau, voire un mélange de l'ensemble. Même si le recours aux logiciels sous licence *Open Source*, sans frais de licence et fournis en mode *Cloud*, tend à se développer⁷², les services prennent bien souvent le pas sur ceux-ci (les logiciels étant accessoires et substituables)⁷³.

Dans ce contexte, aux problèmes classiques posés par le *Cloud* (dépendance vis-à-vis du fournisseur, sensibilité des données hébergées, aucune maîtrise sur la pérennité et stabilité des services⁷⁴), s'ajoutent ceux liés à l'impact du *Cloud Computing* sur les Logiciels Libres. Ainsi, en l'absence de distribution des Logiciels Libres⁷⁵, la majeure partie des licences (même *copyleft*) n'imposent aucune communication des codes sources des logiciels lors de leur utilisation sous forme de service⁷⁶. En effet, la mise à disposition d'un service sur le web ne nécessite pas l'installation d'un logiciel par l'utilisateur final. Dès lors qu'il n'y a pas distribution de logiciel, les droits et devoirs des licences ne s'imposent pas. Il s'agit ainsi de problématiques similaires à celles rencontrées avec les logiciels propriétaires, puisqu'il y a une possibilité de rendre « indisponible » le code source du logiciel délivrant le service.

D'où la question : « le Logiciel Libre est-il soluble dans le *Cloud* ? »⁷⁷ car de nombreuses sociétés proposent des services sur Internet qui reposent sur des Logiciels Libres en adoptant une position plus ou moins ambiguë vis-à-vis de l'*Open Source*⁷⁸. Devant la lacune des licences traditionnelles, dénommée « ASP Loophole »⁷⁹, une série de nouvelles licences⁸⁰ – la licence GNU Affero GPL en tête – fait en sorte que le licencié qui utilise un logiciel pour fournir un service à des utilisateurs via le réseau ne puisse le faire sans redistribuer les modifications qu'il aura apportées à ce logiciel⁸¹.

Deux raisons semblent pouvoir motiver le choix de ces licences :

- La première : un choix de « forcer » la collaboration et les échanges entre des acteurs industriels concurrents (avec ou sans contrat de consortium), dans la droite lignée de ce que l'on appelle coopération⁸², ou
- La seconde : une contrainte liée à l'intégration d'un des nombreux composants sous une licence du type GNU Affero GPL⁸³.

⁶⁹ Par exemple Aladdin SW.

⁷⁰ Voir notamment Benjamin JEAN, « Cloud Computing et Open Source », intervention réalisée lors de la journée JuriTIC sur « le point sur le droit des logiciels... jusqu'au Cloud Computing », 6 mars 2009 à Bruxelles. Plus Livre Blanc du Cloud Computing, Syntec Informatique 2010

⁷¹ Notamment la crise économique, les préoccupations environnementales, la progression du Logiciel Libre (qui font abandonner la simple valorisation par la rente) et l'influence du « Cloud computing ».

⁷² Ainsi en est-il de la majeure partie des logiciels utilisés pour faire fonctionner le web (Apache, PHP, MySQL, GNU/Linux, BSD, etc.), l'Internet (Postfix, Bind, etc.), mais aussi les logiciels plus spécifiques au Cloud tels Hadoop, Eucalyptus, Xen, KVM, etc. Voir notamment l'édition 2009 du Baromètre « Atouts & Bénéfices du modèle SaaS/on demand », réalisée par le Cabinet Markess International, janvier 2009. Voir : www.markess.fr.

Voir également : <http://developers.facebook.com/opensource.php> et <http://blog.twitter.com/2008/01/twitters-starling-released-as-open.html>

⁷³ Il convient toutefois de distinguer deux types d'utilisateurs de Cloud Computing : les professionnels qui utilisent ces services pour y ajouter leur propre valeur, et les consommateurs qui en sont les destinataires terminaux.

⁷⁴ Et ceci, quelles que soient les clauses de responsabilité prévues.

⁷⁵ Voir à ce sujet la partie 3, sur la description des licences.

⁷⁶ Ces dernières assimilant l'usage à ceux permis sans limites « dans la sphère privée » de l'utilisateur. L'explication est simple : 1) L'utilisateur accède via son navigateur au site du licencié ; 2) à sa demande, une requête est envoyée au licencié ; 3) Celui-ci traitera cette requête en interne sur ses propres PC hébergeant le logiciel (et donc le service). Ainsi, le licencié enverra la réponse sans jamais avoir distribué le logiciel.

Il faut néanmoins reconnaître que certains logiciels (tel le code javascript récupéré dans le navigateur, AJAX, etc.) interagissent directement avec l'utilisateur.

⁷⁷ Voir : <http://www.2020flossroadmap.org/roadmap/head-in-the-clouds/#head>

⁷⁸ Alors que certains projets reçoivent leur soutien, les modifications ou améliorations apportées au logiciel par ces dernières ne sont pas forcément reversées.

⁷⁹ Littéralement « Faille ASP ». Voir à ce sujet « GNU Affero GPL version 3 and the "ASP loophole" », <http://www.opensource.org/node/152>

⁸⁰ D'autres licences avaient déjà intégré en leur sein ce mécanisme : l'Honest Public licence, l'Open Software License 3.0, la Reciprocal Public License 1.5, la Common Public Attribution License Version 1.0 (CPAL). L'European Union Public Licence le fait aussi depuis sa version 1.1.

⁸¹ Sur ce sujet, voir Anne PERNY, « SAAS : quelles licences open source adopter ? », mémoire de stage sous la direction de Benjamin JEAN, septembre 2009.

⁸² Néologisme qui renvoie aux notions « coopération » et « compétition ».

⁸³ Il y a aujourd'hui plus de 421 projets sur Sourceforge sous GNU AGPL.

Parallèlement et réaffirmant la liberté de l'utilisateur, un certain nombre de sociétés essaient de casser le modèle actuel grâce à l'*Open Source*⁸⁴ : en utilisant la version communautaire de leur Logiciel Libre⁸⁵ pour leur service et en donnant aux utilisateurs la possibilité d'exporter le code, le thème et les données sur une autre instanciation (mutualisée ou dédiée) de ce même logiciel (c'est notamment le choix réalisé par la société Acquia, à l'origine du CMS Drupal, avec son offre « *Drupal Garden* »)⁸⁶. Il n'y a donc aucun blocage au détriment de l'utilisateur, ce qui s'avère intéressant pour le projet *Open Source*, les utilisateurs et les sociétés qui se greffent sur le modèle : les Logiciels Libres offrent la faculté de déployer pour soi et chez soi une nouvelle instanciation d'un logiciel auparavant utilisé comme service, afin de maintenir une indépendance vis-à-vis d'un fournisseur, un contrôle sur ses services et une confidentialité quant à ses données.

Les enjeux dépassent les simples droits de propriété intellectuelle : la vie privée et les données personnelles sont impactées ainsi que les données libres, l'interopérabilité et la standardisation. C'est ce constat qui a motivé la création d'initiatives, telles que l'*Open Cloud Consortium*⁸⁷, l'*Open Cloud Manifesto*⁸⁸ ou *TIOLive*⁸⁹. Et bien qu'aucune de ces initiatives n'arrive à fédérer les efforts, voire à faire émerger des standards, elles convergent vers un même but retrouver dans le monde du *Cloud* les mêmes qualités que celles que nous trouvons dans le monde *Open Source* (l'absence de barrière d'entrée et de sortie, l'absence de discrimination, l'interopérabilité, la neutralité technologique et la transparence).

2.1.4 LE MODÈLE DE « SERVICES À VALEUR AJOUTÉE »

Il s'agit ici des équivalents *Open Source* des SSII : leurs métiers consistent principalement à l'intégration, la maintenance, le paramétrage, etc., de Logiciels Libres et à l'accompagnement des clients dans cette intégration.

On parle généralement de SSLL (« sociétés spécialisées en logiciels libres ») pour les sociétés spécialisées dans l'intégration et les services autour des logiciels libres, mais les éditeurs de Logiciels Libres ont eux-mêmes fréquemment une offre de ce type (Red Hat, Novell, etc.) et les grands intégrateurs commencent à développer en interne des compétences similaires (même s'ils dépendent fréquemment des compétences fournies par les SSLL).

2.1.5 LE MODÈLE D'INTÉGRATEUR HYBRIDE

On entend par « hybride » le fait de fédérer des offres « produits » et des offres « services » au sein d'un prestataire unique positionné soit sur un axe métier (par exemple la sécurité), soit sur l'axe du système d'information global de l'entreprise (par exemple le système d'information d'une collectivité locale, d'une filiale ou division d'une entreprise du CAC 40) en garantissant au client la cohérence et la pérennité de l'ensemble. Il s'agit là du développement de structures concurrentes des intégrateurs traditionnels (Capgemini, Atos Origin, Steria...).

Les modèles doubles ou hybrides connaissent une croissance forte depuis quelques années et cette croissance se confirme pour les années à venir. Le marché connaît une grande prolifération d'acteurs de petite taille, éditeurs pratiquant le *dual licensing* et SSLL avec des outils libres notamment qui suivent des stratégies de niche. Mais surtout, le Logiciel Libre passe **d'un marché émergent à un marché d'expansion (tous les aspects d'un système d'information, toutes les activités impliquant des ressources logicielles) et de maturation (avec la consolidation des modèles économiques en question).**

2.2 IMPACTS SUR LA RELATION CLIENT

Les *business model* construits autour de Logiciels Libres, dans la mesure où ils comprennent l'accès au code source et à l'exploitation du logiciel, modifient la relation client⁹⁰.

Le Logiciel Libre permet de répondre à certaines demandes du client, demandes relatives à la pérennité de son système d'information et à l'indépendance vis-à-vis d'un fournisseur. Néanmoins, un véritable travail d'explication et de sensibilisation est nécessaire pour tempérer certaines demandes inadaptées (tirées de l'expérience passée de certains clients), et rechercher la solution optimale pour tous.

⁸⁴ Cherchant par ce biais à calquer le modèle *Open Source* sur les services (se servant de l'architecture, de la mutualisation, du partage et de la structure offerte par l'*Open Source*).

⁸⁵ Le choix d'une licence du type GNU Affero GPL étant selon nous un gage de sécurité supplémentaire.

⁸⁶ Voir : <http://www.drupalgardens.com/>

⁸⁷ Voir : <http://www.opencloudconsortium.org/>

⁸⁸ Voir : <http://www.opencloudmanifesto.org/>

⁸⁹ Voir : <http://tio.ffii.org/>

⁹⁰ À l'exception du *SaaS*, ainsi que précisé supra.

L'enjeu est ici de permettre au client d'évaluer l'impact potentiel d'une utilisation de l'*Open Source* en lieu et place d'un composant traditionnel. Même si l'impact peut s'avérer moindre que ce à quoi le client s'attend, il n'est pas dénué d'effet).

2.2.1 L'EXPRESSION DES DEMANDES

Les clients demandent souvent à disposer des « sources » (comprendre le code source) ou bien encore à se voir attribuer l'ensemble des droits patrimoniaux attachés au logiciel.

Dans la logique propriétaire, l'éditeur titulaire des droits d'auteur ne donnera que très rarement accès au code source et le compromis réside souvent dans la mise sous séquestre de ce code source. La question est résolue d'office en matière de Logiciel libre puisque chaque utilisateur se voit transférer un logiciel, avec son code source et la liberté (notamment) de le modifier.

Pour autant, il est impératif de connaître les motivations des clients afin de leur fournir la meilleure réponse et la meilleure solution technique. Ainsi, il ne suffira pas d'avoir accès au code source ou d'être cessionnaire des droits pour assurer la pérennité du système si tel est l'objectif. Encore faudra-t-il être en mesure d'assurer la continuité.

2.2.2 COMPRENDRE LES MOTIVATIONS DES DEMANDES

Pourquoi le client demande-t-il à être propriétaire des « sources » ou au moins à en disposer ?

De façon classique, les demandes relatives au code source ou à la propriété intellectuelle sont souvent animées par un souci d'ordre sécuritaire, d'assurer la pérennité des systèmes. Ces demandes seront d'autant plus insistantes que l'enjeu est lourd pour le client et que le risque de défaillance du projet est susceptible d'avoir des répercussions importantes sur le fonctionnement de l'entreprise. On ne peut blâmer un client préoccupé de savoir comment fonctionne (et fonctionnera) son logiciel. Le logiciel est un outil clé de l'entreprise, il fait tourner les ordinateurs, les téléphones et de nombreux appareils. Pour autant, disposer du code source ou des droits de propriété intellectuelle permet-il d'atteindre l'objectif fixé ?

Il est nécessaire de rappeler au client que le transfert de propriété implique également transfert des droits et devoirs attachés à cette propriété. Ainsi, le client n'aura plus nécessairement intérêt à rester propriétaire de l'application : en adoptant un logiciel propriétaire, il bénéficiera d'une maintenance évolutive mutualisée par la SSII pour tous les utilisateurs concernés ; en optant pour un Logiciel Libre, il aura accès, en plus de la maintenance évolutive mutualisée par une SSLL⁹¹, aux évolutions et améliorations apportées par la communauté (directement ou par le biais de son prestataire).

2.2.2.1 LA PRÉSERVATION DU SAVOIR-FAIRE

Autre explication : dans son cahier des charges, le client transfère beaucoup de son savoir-faire métier à la société de services. Son système d'information contribuera à le différencier de ses concurrents et à en tirer avantage. Par exemple, il peut s'agir de systèmes de fidélisation des clients, de méthodes originales de production, etc. Très logiquement, il ne souhaite pas que son savoir faire soit communiqué à ses concurrents – ce qui pourrait être le cas si le prestataire réutilisait des développements spécifiques pour compléter sa propre offre. Nombre de progiciels sont effectivement nés de la collaboration entre un éditeur et un client.

Notons que l'impact des licences *Open Source*, même copyleft, sera inexistant en l'espèce puisque :

- Dans le cas d'un développement spécifique, seul le client bénéficie de la cession « contrainte » des droits sur le logiciel : à sa charge ensuite de décider d'en limiter la diffusion ou de l'exploiter (les problèmes peuvent néanmoins ressurgir en cas de division de ses activités, acquisition partielle, etc.);
- Dans le cas d'une maintenance, d'un support ou d'une assistance réalisés sur les machines du client et sous son

⁹¹ À l'opposé, le choix de l'*Open Source* peut aussi permettre de maîtriser le cycle de vie du logiciel, sans support dégradé ou montée de version forcée

contrôle, alors le logiciel n'est pas distribué et le client n'est pas contraint de donner accès au code source⁹².

D'où la nécessité, d'identifier en amont d'une part les objectifs du client et d'autre part les composants logiciels. Plusieurs réponses sont alors possibles, dont une relative aux licences *Open Source* :

- Commencer par identifier clairement les composants et/ou développements correspondants aux demandes spécifiques du client. Ils seront alors distingués et un éventuel accord de transfert pourra les viser. Le fournisseur pourra ainsi écarter de la discussion tout ce qui concerne les Logiciels Libres, modifiés ou non.
- Ensuite, proposer des solutions alternatives au transfert de propriété comme une clause d'exclusivité sur le marché considéré, pour une durée et une étendue géographique à déterminer avec le client, par exemple, assurant au client qu'il ne retrouvera pas son savoir-faire chez un concurrent. C'est ici le fournisseur/l'éditeur qui s'oblige le plus, et donne plus de droits à son client, ce qui est tout à fait conforme au texte des licences.
- Transférer le code source sous licence *Open Source* pour garantir au client non seulement la disponibilité des sources correspondantes, mais encore la possibilité de les utiliser, modifier, développer à sa convenance et sans contrainte⁹³. Cette solution permet aussi d'assurer la continuité en cas de disparition du prestataire.
- Transférer les composants spécifiques en pleine propriété (c'est-à-dire céder la titularité des droits). En pareil cas, il convient d'identifier avec le client quels sont les droits patrimoniaux à transférer pour répondre à son attente. Par exemple, droits de commercialisation sur certains marchés ou zones géographiques, droits d'utilisation dans certains contextes, etc. Cela permettra également au client de bien identifier son « vrai » besoin et, au fournisseur, d'y apporter des réponses appropriées en appauvrissant aussi peu que possible son propre potentiel professionnel.

2.2.2.2 AUTONOMIE VIS-À-VIS DU FOURNISSEUR

Il s'agit pour l'utilisateur d'assurer la pérennité globale de son système et donc de la problématique de dépendance envers ses fournisseurs, de leurs disponibilités pour la maintenance et/ou les développements ultérieurs, de la cohérence du système contractuel et technique (avec notamment des enjeux de cohérence entre les divers choix en matière de licences *Open Source*).

Par exemple, il arrive fréquemment qu'un Grand Compte détecte une solution logicielle qui lui convient parfaitement, mais dont le porteur lui semble être une société fragile financièrement. Il aura tendance à se tourner vers un Intégrateur Système qui prendra en charge la cohérence de l'ensemble et qui en sera responsable en maintenance et en suivi.

Le client veut pouvoir s'affranchir de la dépendance d'un fournisseur en cas de défaillance de celui-ci et éviter tout litige commercial, désaccord sur le prix des prestations, etc. Il souhaite donc disposer de l'ensemble des sources et ressources lui permettant d'assumer, par lui-même ou avec l'aide d'un sous-traitant, la continuité de service et les développements. Comme toute médaille à son revers, il perdra alors beaucoup de poids vis-à-vis de son fournisseur qui se sentira libre de s'affranchir, au moins partiellement ou temporairement d'obligations de maintenance et de suivi, au motif qu'il n'est plus indispensable à son client, celui-ci disposant de toute la « matière première » nécessaire. Le fournisseur peut être tenté de mettre ses moyens en priorité vers les clients qui dépendent totalement de lui. Pourtant, si le client externalise ses développements, leur maintenance et même leur hébergement ; c'est bien pour ne pas avoir à le faire par lui-même. Il ne dispose généralement pas des ressources ou des compétences et connaissances nécessaires. Il restera donc, au moins à court terme, dépendant de son prestataire.

Il est possible au client de demander un transfert de l'ensemble des composants (composants matériels, composants plates-formes et composants spécifiques), dans le cadre d'une licence *Open Source*. Il disposera ainsi de l'ensemble des sources – y compris des plates-formes de développement et de déploiement *Open Source* – nécessaires pour assurer par lui-même, ou avec l'aide d'un autre prestataire, la pérennité de son système. Les prestataires qui fournissent ce genre de solutions facturent essentiellement leur expertise en matière d'architecture, de choix de composants, d'assemblage et de validation desdits composants, leur déploiement, leur paramétrage⁹⁴. Le fait pour le client de disposer de l'ensemble des sources représente non seulement un confort « psychologique », mais une réelle possibilité d'indépendance vis-à-vis de la fourniture de services : il peut ainsi changer de prestataire sans coût disproportionné. L'usage de Logiciels Libres est ainsi parfaitement adapté à ce type de demande.

La mise sous séquestre du patrimoine, accessible au client sous condition, est une alternative disponible tant à l'égard de logiciels libres que propriétaires, elle concernera les composants plates-formes, Framework métier et composants

⁹² Certaines licences viennent d'ailleurs préciser ce point et l'étendre : voir notamment l'article 2 de la GNU GPL v3 « You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you. »

⁹³ Au risque de la création d'un fork si le projet est ensuite maintenu par une autre société et dans un autre sens.

⁹⁴ S'ajoute le cas échéant la maintenance évolutive.

spécifiques client. Le véritable enjeu pour le client n'est pas patrimonial, mais bien opérationnel : il souhaite pérenniser son système d'Information ou de production. Naturellement, le contenu de ce séquestre doit être soigneusement décrit (cf. *infra* l'annexe technique de description d'une livraison, en cas de séquestre) et surtout testé. Il doit également faire l'objet d'un suivi pour limiter de façon raisonnable la différence entre la dernière version séquestrée et la version en production⁹⁶.

En matière d'*Outsourcing* et de mutualisation des plates-formes de production (ASP, SaaS), les enjeux concernent principalement les services (en matière de Contrat de Niveau de Service⁹⁷ et de Plan de Reprise d'Activité⁹⁸). Les problématiques sont généralement les mêmes en matière de logiciel propriétaire et de Logiciel Libre – à l'exception des acteurs qui étendent les libertés des utilisateurs à ces usages⁹⁹.

Dans bien des cas, les demandes clients peuvent être traitées par une clause de cession sous licence *Open Source* (si le prestataire assure le pilotage du projet *Open Source*), de transfert des sources (s'il peut assurer, ou faire assurer la maintenance), selon le type de composant concerné. Toutefois et dans la mesure où l'utilisateur n'a pas précisément de compétence en matière de développement et de maintenance de ladite application, disposer des « sources » peut poser plus de problèmes que cela n'apporte de solutions crédibles à ses questions sur la pérennité de son investissement, et donc de tout ou partie de son système d'information¹⁰⁰. Le client doit mesurer l'intérêt de disposer des « sources » à l'aune de ses objectifs.

2.3 IMPACTS SUR L'ORGANISATION DES SERVICES DE L'ENTREPRISE

Une première étude de l'impact sur les services existants amène à recommander une évolution organisationnelle au sein de structures éditrices ou utilisatrices de Logiciel Libre.

2.3.1 IMPACTS SUR LES SERVICES EXISTANTS

2.3.1.1 SERVICE TECHNIQUE

Une gestion plus stricte des projets informatiques s'impose (cf. *supra*) et les considérations développées dans ce document sur les licences des logiciels s'appliquent donc également :

- À l'ensemble de l'infrastructure nécessaire au développement du logiciel, notamment dans le cas de vente d'une solution globale,
- À toute acquisition et fabrication de matériel incluant des Logiciels Libres,
- y compris aux services connexes de support, de maintenance et de formation.

Au stade du développement, de la réalisation, de l'intégration, **le développeur doit garder des traces de son travail au sein de l'inventaire : il doit être toujours en mesure de lister les composants du projet**, mais aussi les éléments techniques permettant de réaliser le projet. L'ensemble de ces informations est destiné à devenir l'annexe technique du contrat. Le développement technique ne peut ignorer les contraintes juridiques (et doit donc être formé à cet égard).

⁹⁶ Option complémentaire : Il est possible de proposer au client que la version initiale comme toute version majeure qui lui sera livrée soit entièrement produite à partir des seules sources contenues dans le séquestre. Il aura alors l'assurance de la parfaite symétrie entre les « sources séquestrées » et la version déployée. Et ceci est valable aussi bien pour du Logiciel Libre que pour du code propriétaire dans la mesure où il n'y a pas de transfert de propriété. En ce qui concerne le Logiciel Libre, cela permet également d'identifier clairement l'état technique des Logiciels Libres que le fournisseur aura intégrés dans sa livraison et qu'il aura testés et validés en conséquence.

Cette solution permet également d'identifier clairement l'état du patrimoine concerné en cas de litige et notamment en cas d'intervention simultanée sur le fonctionnement du logiciel de la part du client comme du fournisseur. Par exemple, état des structures des bases de données, paramétrages...

Ne disposant pas des sources, l'utilisateur ne peut être mis en cause dans ce type de cas. De même, le prestataire sera assuré qu'aucune intervention intempestive ne sera effectuée par l'utilisateur, mettant en cause le bon fonctionnement de l'application.

Le séquestre pourra également recevoir un certain nombre de documents, procédures, composants qui ne sont généralement pas fournis au client, mais dont la disponibilité est importante en matière de reprise de maîtrise de l'application : règles internes de développement et de nommage, copie des bases de source, dossiers d'architectures, procédures automatisées de mise à jour et de maintenance, etc.

⁹⁷ Aussi appelé SLA pour Service Level Agreement.

⁹⁸ DRP en anglais, pour Disaster Recovery Plan.

⁹⁹ Voir *supra* : Open Source et Cloud Computing.

¹⁰⁰ Et pour être efficace, le client devra former en permanence ses équipes, tant pour la livraison initiale que pour les mises à jour et développements additionnels, ce qui est très coûteux. Il perdrait ainsi les avantages financiers d'une externalisation et d'une mutualisation des coûts, son fournisseur pouvant répartir ces coûts sur tous ses clients, à l'exception des composants spécifiques qui pourront faire l'objet d'un traitement contractuel particulier.

Il est recommandé :

- De renseigner au sein de l'inventaire, au fur et à mesure de l'avancée des travaux de développement, une liste qui distingue et détaille les modules et composants logiciels utilisés (une procédure interne complémentaire de mise en place d'un référentiel permet par ailleurs d'assurer un suivi des développements, cf. *supra*).
- De collaborer étroitement avec le service juridique de façon à assurer une compréhension optimale du processus de développement et à guider les choix techniques lorsque ceux-ci reposent sur des contraintes juridiques.
- D'intégrer au service à la fois la politique de commercialisation décidée par l'entreprise éditrice et/ou le prestataire de services informatiques, et les impératifs juridiques définis par l'utilisation de modules libres. Les contraintes du « libre » doivent donc être maîtrisées en amont aux niveaux techniques et juridiques pour recevoir une traduction claire en aval, au niveau commercial. Il s'agit de dire au client ce qui est possible, tant techniquement que juridiquement.

Le raisonnement adopté lors de l'élaboration du logiciel s'applique aussi à la documentation technique. Celle-ci peut-être soumise à une licence différente de celle régissant l'utilisation du logiciel¹⁰¹. Une réflexion à l'égard de cette documentation est d'autant plus importante qu'il s'agit d'un domaine où les entreprises sont créatrices de valeur et qu'une mutualisation sous Licence *Open Source* conformément à une politique clairement définie s'avère bien moins sujette à contrainte qu'une cession *a posteriori* ainsi que l'exigerait le droit d'auteur.

Dans sa recherche de solutions *Open Source* optimum, et dans sa confrontation aux jugements de centaines ou de milliers de développeurs en cas de libération de code, chacun des membres du service technique, acquiert un rôle dont la charge compensera (en partie) le temps gagné par la reprise de code tiers. Parallèlement, la responsabilité de chacun de ses membres sera d'autant plus forte que les choix techniques peuvent porter sur des logiciels aux licences complexes. Un processus simple de validation doit donc être mis en place auprès du service juridique.

2.3.1.2 SERVICE JURIDIQUE

Le service technique et le juridique doivent travailler main dans la main, avant même le lancement du projet et jusqu'à sa délivrance.

L'intervention de l'équipe juridique est nécessaire en interne (validation des logiciels *Open Source* utilisés), pour la fabrication de produits matériels incluant du logiciel, pour l'édition ou l'intégration (dès l'étude de la rédaction de l'appel d'offres, du cahier des charges ou de la réponse). Ses compétences en terme de propriété intellectuelle doivent lui permettre lors de la conception du projet informatique :

- De participer au choix de la(les) licence(s) *Open Source* (en proposant si possible plusieurs options) et à la conception de la politique de la société en matière de propriété intellectuelle (droit d'auteur, mais aussi droit des marques, brevets et autres droits exclusifs),
- de répondre de la faisabilité de la solution vendue,
- de sécuriser le cadre légal de l'usage de la solution par l'entreprise et de sa diffusion au sein de son organisation, à ses partenaires et/ou à ses clients.

Et, ce faisant, de sécuriser l'usage des droits d'auteur, des licences *Open Source*, du droit des marques et de tout autre droit exclusif. Par la suite, son intervention permettra aussi de sécuriser le contrat de service annexe.

Enfin, il doit sensibiliser les acteurs de la société aux enjeux de l'*Open Source*, publier une veille sur le sujet, s'assurer du respect des licences *Open Source* et de la stratégie globale en matière de propriété intellectuelle. Dans le cadre de la gouvernance (à la conception de laquelle il doit être associé), il peut être opportun de lui reconnaître un droit de veto en cas de problème détecté en matière de licences.

2.3.1.3 SERVICE ACHAT

Le service achat doit s'assurer auprès des fournisseurs de l'entreprise que le statut des licences, des solutions et produits acquis est conforme aux attentes de l'entreprise, notamment dans la perspective d'une réutilisation de ces acquisitions au sein d'un produit qui sera lui-même vendu par l'entreprise.

Il est ainsi recommandé de prévoir un volet dans la politique d'achat concernant les règles en vigueur sur les licences *Open Source* (notamment les licences proscrites).

¹⁰¹ Voir en Annexe le modèle de diffusion de document développé au sein de la société LINAGORA.

2.3.1.4 SERVICE MARKETING ET COMMERCIAL

En définissant l'offre, il est nécessaire d'élaborer un argumentaire commercial sur les contributions de l'entreprise dans le domaine, sur les avantages, mais aussi sur l'impact du Logiciel Libre et des licences *Open Source*. Le porteur de l'offre – le commercial – devra pouvoir expliquer au client que les avantages du Logiciel Libre impliquent obligatoirement de renoncer à certaines idées préconçues (plus souvent à tort qu'à raison), comme la nécessité de disposer d'une propriété absolue sur la solution ou de bénéficier sur un logiciel libre des garanties commerciales habituelles des logiciels propriétaires.

Le personnel commercial devra adopter un discours objectif permettant d'écarter les principales demandes irréalisables des clients, notamment sur la propriété du logiciel. Ce point n'est d'ailleurs pas spécifique au logiciel libre et concerne tout autant le logiciel propriétaire (cf. *supra*). **Plus la collaboration entre le domaine technique et juridique sera efficace, plus le discours commercial sera simplifié (transparence et sécurité).**

2.3.1.5 SERVICE RESSOURCES HUMAINES

Respectant les choix stratégiques de la société en faveur de l'*Open Source*, le service des ressources humaines est dans la nécessité d'adapter ses pratiques en matière de recrutement de gestion des compétences et mettre en place un plan de formation afin d'alerter, de sensibiliser et de responsabiliser les techniciens et les commerciaux. Ce choix peut notamment permettre de stimuler une équipe, rendre attractive l'embauche et remotiver les seniors. Il doit aussi permettre d'intégrer, en cours de projet, des compétences *Open Source*.

Par principe, un règlement intérieur (ou une charte) doit définir les règles à suivre lors du recours aux Logiciels Libres (notamment en fonction des licences *Open Source*, de l'objectif poursuivi...) et des moyens permettant de s'assurer de son application doivent être prévus.

Les critères de sélection et de rémunération des collaborateurs doivent être conçus de manière souple et attractive compte tenu de la forte proportion d'autodidactes (pionniers dans une filière en croissance) du domaine. Les grilles de salaires qui sont fonction des diplômes ne sont plus adaptées et il est nécessaire d'appliquer de nouveaux critères, comme l'implication dans les communautés, les projets, l'évaluation par ses homologues, etc. Pour stimuler les équipes, les compétences des salariés devront jouer un rôle majeur, tant au moment de l'embauche que pour l'évolution au sein de la société (mission, formation).

Enfin, les contrats de travail (et les documents annexes) doivent être repensés afin d'inclure les particularismes liés au développement libre et au *business model* des sociétés.

- **Responsabiliser les salariés** : la réutilisation de composants tiers (sous licence libre ou propriétaire) doit être clairement encadrée par une charte annexée au contrat de travail.
- **Clause de non-concurrence**. La clause de non-concurrence doit être proportionnée aux intérêts tant de la société que des collaborateurs, sachant que de nombreux salariés apportent avec eux des compétences et des contributions ensuite valorisées par la société.
- **Publication de logiciels**. S'il est communément admis que la première publication d'un logiciel passe nécessairement par une validation hiérarchique (généralement de la direction), la question est moins simple lorsqu'il s'agit d'un logiciel libre, par définition constamment amélioré, et mis à disposition (parfois dans l'heure). Il est donc nécessaire d'encadrer précisément le rôle et le pouvoir des salariés en charge de cette mise à disposition, généralement les développeurs du projet en question (au sein d'une même charte annexée au contrat de travail – un compromis consiste par exemple à distinguer par type d'améliorations et de versions).
- **Gérer les contributions des salariés**. Rappelés à l'ordre par les dernières jurisprudences, les responsables des ressources humaines veillent dorénavant à compléter utilement les contrats qui ne sont pas exactement des contrats de travail (stagiaire, doctorant, dirigeant, prestataire externe, etc.) par une cession expresse de droit de propriété intellectuelle rejoignant le régime des salariés (cf. *supra*)¹⁰². Néanmoins, il y a une autre difficulté : dans le domaine du logiciel plus qu'ailleurs, les salariés développeurs sont souvent amenés à contribuer dans des cadres juridiques multiples (au sein de la société qui les emploie sur des projets, chez eux dans le cadre de projets personnels ou pour des besoins professionnels, etc.). Sachant que la jurisprudence interprète toute ambiguïté en faveur des salariés, il y a lieu de s'alarmer lorsque le business model de la société repose sur la titularité des droits.

¹⁰² Même si l'hypothèse ne peut être développée dans ce document, les entreprises doivent veiller à récupérer, par des mécanismes a posteriori, les créations salariales qui, n'étant concernées par l'exception au profit des seuls logiciels, restent la propriété des auteurs salariés.

Ainsi, pour assurer la sécurité juridique de la société, ce point doit faire l'objet d'un dispositif normatif (par exemple une charte) entre la société et ses salariés. La société sera amenée à dissocier par période et par projet :

- Pour toute contribution fournie durant le temps de travail, il est possible d'imaginer que la société soit titulaire de droits sur toutes les contributions relatives à ses propres projets, ainsi qu'à ceux dans lesquels elle a un intérêt direct (projet du catalogue de la société par exemple), alors qu'elle n'acquerra pas les droits sur les contributions portant sur des projets dans lesquels elle n'a pas d'intérêt direct. Les projets de la société et ceux dans lesquels elle aurait un intérêt seront clairement listés, afin d'éviter que la contribution d'un salarié dans un projet non connu mette en péril le patrimoine immatériel de la société (notamment en matière de brevet si la licence inclut une cession de brevets)
- Pour toute contribution fournie en dehors du temps de travail, il faudrait distinguer celles qui entrent dans le cadre d'une mission précise (soumises au même régime que précédemment), celles qui portent sur des projets pour lesquels la société est directement intéressée (assimilées aux contributions de mission), et les contributions qui n'intéressent pas la société, sur lesquelles le salarié conserve ses droits.

2.3.1.6 DIRECTION GÉNÉRALE

Les entreprises doivent prendre conscience des enjeux liés à la bonne appréhension du « Libre ». Il n'existe plus, ou à de rares exceptions, de logiciel 100 % propriétaire. 80 % des logiciels contiennent des composants sous Licence *Open Source* et de plus en plus nombreux sont les ingénieurs qui, à chaque étape du développement, réutilisent des composants sous licence *Open Source*.

L'utilisation mal maîtrisée de code *Open Source* peut être à l'origine de risques pour l'entreprise qu'elle soit fournisseur et/ou cliente. L'utilisation « sauvage » des ressources sous licences *Open Source* peut aisément compromettre les droits de propriété intellectuelle, notamment en ajoutant des frais de licence imprévus à la charge de l'utilisateur. Par ailleurs, le non-respect des licences pouvant être sanctionné par une action en contrefaçon, action pénale et/ou civile, il est indispensable d'encadrer ces utilisations pour limiter la responsabilité des décideurs (la simple évaluation des risques ne pouvant suffire).

Sous l'impulsion de la direction, l'entreprise doit :

- Respecter les licences *Open Source* tout au long de son processus du développement à la vente,
- Maîtriser son code,
- Mettre en place une gouvernance *ad hoc*.

2.3.2 ÉVOLUTION ORGANISATIONNELLE RECOMMANDÉE

Pour toute interaction considérée avec le logiciel libre et l'entreprise, il est capital de mettre en place une structure et des règles qui vont permettre à l'organisation de tirer le maximum de son utilisation du logiciel libre, tout en contrôlant les risques potentiels associés.

Pour ce faire, il est indispensable de mettre en place au sein de l'organisation une structure, rattachée à la direction générale, et ayant son soutien, en charge de toutes les interactions, de la politique globale de l'organisation vis-à-vis du Logiciel Libre et du contrôle de son application. Suivant la taille de l'entreprise, la variété de son utilisation des logiciels libres, une structure adaptée devra être mise en place. Au minimum, il est recommandé d'avoir un référent *Open Source*.

À l'instar de l'aspect modulaire de nombreux logiciels libres, il est possible de penser cette organisation en « différentes missions », afin que celles-ci puissent ensuite assurer la charge d'une ou plusieurs structures.

2.3.2.1 RÉFÉRENT *OPEN SOURCE*¹⁰⁴

Son rôle est d'être le « *Monsieur Open Source* » de l'organisation. Il aura en charge de piloter le développement de la politique globale *Open Source* de l'organisation (sous forme d'un manuel), ainsi que des procédures à mettre en œuvre dans les différents services pour prendre en compte les impacts de l'introduction du logiciel libre mentionnés plus haut dans ce document.

Il doit aider à développer un modèle de gouvernance *Open Source*, intégré dans la gouvernance informatique de

¹⁰⁴ Pour un exemple, voir l'annexe 2, la fiche de poste *Open Source* chez Orange.

l'organisation, dont les missions seront notamment de créer le Comité de Direction *Open Source*, de statuer sur l'opportunité de créer un comité de revue *Open Source* dans l'organisation et la formation du service juridique.

Il a aussi comme rôle la veille technologique externe (relais d'information sur le monde du logiciel libre vers l'intérieur de l'organisation, interface avec des communautés) et interne (inventaire des logiciels libres utilisés, relation avec le département juridique, propriété intellectuelle, achat, établissement d'une communauté interne, formation...)

Dans le cas d'organisations plus importantes, ce référent peut devenir une équipe complète, appelée alors Comité de Pilotage *Open Source*, au sein de laquelle les différentes fonctions précédentes seront réparties.

Le développement de cette fonction possède beaucoup de point commun avec le Correspondant Informatique et Libertés (CIL)¹⁰⁵, notamment celui d'être massivement déployé au sein du secteur privé en quelques années seulement.

2.3.2.2 COMITÉ DE DIRECTION OPEN SOURCE

Initié par le référent *Open Source*, ce comité doit être composé au minimum du référent *Open Source*, d'un représentant de la direction générale, d'un représentant du service juridique, et d'un représentant des services techniques. Il est le garant du modèle de gouvernance mis en place et prend toutes les décisions dans l'organisation impliquant des logiciels libres et *Open Source*. Il se fait assister de toutes les compétences (internes ou externes) nécessaires à sa mission.

Son rôle est de valider la gouvernance globale *Open Source*, le manuel de politique *Open Source* la décrivant, ainsi que de fournir les moyens de son application. Il établira le plan de communication interne de sa politique *Open Source* et veillera à la formation du personnel. Il statue sur l'utilisation des logiciels libres interne au sein de l'organisation que vis-à-vis de clients.

Il décide de la politique de participation des employés aux projets de Logiciels Libres, et pourra éventuellement proposer des avenants aux contrats de travail pour prendre en compte les spécificités apportées par l'*Open Source* quant à la propriété intellectuelle.

En étant le garant des risques vis-à-vis de la direction générale, il facilite l'utilisation de l'*Open Source* dans l'organisation pour que celle-ci en tire les meilleurs bénéfices possibles.

2.3.2.3 COMITÉ DE REVUE OPEN SOURCE

Dans le cas où l'organisation serait amenée à redistribuer du logiciel libre (dans du matériel, avec du logiciel, propriétaire ou non, au travers de services), il est important de mettre en place une structure qui statue sur l'utilisation licite, vue de l'organisation, du logiciel libre dans le cadre de cette distribution, au cas par cas, en appliquant les directives promulguées par le Comité de Direction *Open Source*. Ce comité de revue doit comprendre le référent *Open Source* mentionné précédemment, mais aussi un juriste, formé aux problématiques des licences *Open Source*, un responsable technique, un responsable des ventes et éventuellement, suivant la taille de l'entreprise, un responsable propriété intellectuelle, un responsable des ressources humaines... Ces personnes ont un rôle opérationnel et peuvent, suivant la taille de l'organisation, être les mêmes que celles du Comité de Direction.

Ce comité reçoit les propositions des équipes techniques demandant à utiliser des composants logiciels libres dans leur projet, statue sur le caractère licite de la proposition et autorise ou interdit le développement du projet considéré. Ce comité statue aussi sur les requêtes d'employés quant à leur participation à des projets libres, sur leur temps libre ou leur temps de travail. Il peut adopter les outils présentés précédemment, pour faciliter le support de ses activités (gestion de flux, analyse de licences, analyse de code...). Son but peut être de créer un référentiel de Logiciels Libres précédemment inventoriés, analysés et classés en liste blanche, grise ou noire et en permettre la consultation par tout employé pour améliorer la qualité des requêtes lui étant soumises.

2.3.2.4 CHARTE OPEN SOURCE DE L'ENTREPRISE

La mise en place d'une charte *Open Source* au sein de l'entreprise constitue une bonne pratique déjà observée dans plusieurs sociétés du secteur. Elle permet aux développeurs et aux juristes d'avoir la vision la plus exhaustive possible des licences *Open Source* utilisables pour chaque projet ou encore des compatibilités, voire des licences proscrites par le comité de direction. Elle permettra de mieux gérer la réutilisation de code en établissant clairement les règles de réutilisation au sein de l'entreprise et contribuera à l'amélioration de la collaboration entre direction, ingénieurs et juristes. Elle précise aussi les sanctions encourues en cas de non-respect volontaire.

¹⁰⁵ Mis en place par la CNIL, avec le succès qu'on lui connaît.

La charte peut aussi couvrir des sujets connexes comme la politique de l'organisation en matière de contribution et d'utilisation de Logiciel Libre, les contributions personnelles aux communautés Logiciel Libre, des modèles d'entêtes à utiliser dans le code source produit,...

Là encore, on mesure les conséquences organisationnelles. Un modèle de Charte *Open source* est proposé en annexe¹⁰⁶.

2.3.2.5 APPROFONDISSEMENTS

Pour plus de détails sur les fondamentaux de la gouvernance *Open Source*, on se reportera avec profit à la traduction du document « *Open Source Governance fundamentals* » disponible sur le site FOSSBazaar.org

2.3.3 RELATIONS AVEC LA COMMUNAUTÉ OPEN SOURCE

Les communautés *Open Source* sont des groupements de personne (physique ou morale) qui partagent un intérêt et un sens d'appartenance communs à un projet *Open Source*, et qui concourent à sa réalisation.

Lorsqu'une organisation souhaite se constituer en communauté autour d'un projet qu'elle édite ou se rapprocher de la communauté d'un projet *Open Source* qu'elle déploie, une position claire vis-à-vis de ces communautés doit être décidée et suivie.

Ainsi, il est important pour une organisation qui souhaite utiliser des logiciels libres de se poser la question de ses relations avec les communautés qui développent ces logiciels. Ceci sous plusieurs angles :

1. *Quelle visibilité extérieure* est donnée quant à l'utilisation d'un logiciel libre donné¹⁰⁷ ? Autorise-t-on les utilisateurs internes de la société à discuter ouvertement des problèmes rencontrés ? Il est important de ne pas abuser des ressources en temps et en moyens mis en place par les projets et de se conformer à leurs us et coutumes respectifs, qui peuvent fortement varier d'un projet à l'autre.

2. Quelles adaptations doivent être faites sur le logiciel et pour quels bénéfices par rapport à la communauté existante ? Des modifications génériques et non différenciatrices ont tout intérêt à être reversées à la communauté, pour répartir la charge de maintenance et faire un apport en retour de l'utilisation faite du logiciel. La question devra être étudiée plus sérieusement quand il s'agit de modifications soit plus importantes, soit non génériques et qui pourraient être rejetées par la communauté ou porter préjudice la à société¹⁰⁸. Le conseil, en général, est de maximiser les reversements pour entretenir une bonne image d'une part, et répartir la charge et réduire les coûts de maintenance d'autre part.

3. *Quelle politique de contribution* adopte l'entreprise ? Quel retour l'entreprise compte-t-elle faire à la communauté ? Est-ce pris en compte dans son plan et budget marketing ? Même si aucune modification de code n'est effectuée, une contribution sous forme de documentation ou de « bonnes pratiques » est généralement fortement appréciée des projets. De même un emballage, une traduction, des jeux de tests, toutes contributions externes, qui donnent de plus une excellente image de l'organisation, sont recommandés. Si des modifications de code ont lieu, il peut être opportun d'opter pour une politique globale de reversement au sein de l'entreprise et de réfléchir à la propriété intellectuelle associée en particulier dans le cas de licences telles que la GNU GPL v3.

4. Quel engagement à long terme par rapport au projet ? Si le projet est capital par rapport à la stratégie commerciale de la société, il faut considérer la possibilité d'allouer des ressources de l'organisation (temps humain, moyens matériels) au bénéfice du projet. Cette implication permettra une meilleure prise en compte des aspirations de chaque contributeur quant à l'évolution du projet et donc une meilleure adéquation dans le temps de celui-ci avec son utilisation.

D'une manière plus générale, deux écueils sont à considérer dans ce type d'interaction avec une communauté :

- la commercialisation du projet en tournant le dos à la communauté,
- la création d'une version spécifique (*fork*) du projet d'origine.

Dans le premier cas, la communauté risque d'ignorer le projet si son aide est finalement nécessaire ; dans le second, les principaux avantages (qui sont la maintenance du projet par la communauté, l'évolution, etc.) sont supprimés et cela revient à se positionner comme éditeur classique sur un logiciel *Open Source* mal connu (avec tous les investissements en terme de maintenance que cela peut générer).

La création d'une communauté *Open Source* est un projet ambitieux et nécessite un investissement non négligeable aux retours souvent aléatoires : cela revient à partager les outils qui permettront éventuellement à un

¹⁰⁶ Voir la traduction réalisée par Bruno Cornec : "French translation of the HP document FOSS Governance Fundamentals", <https://fossbazaar.org/content/french-translation-hp-document-foss-governance-fundamentals>

¹⁰⁷ Ne serait-ce que par la participation aux listes de discussion avec une adresse professionnelle ou privée.

¹⁰⁸ Cet aspect dépend notamment de la licence et du fait qu'il y ait ou non redistribution.

concurrent de distribuer ce logiciel, mais cela permet aussi d'étendre la diffusion du logiciel (notamment grâce aux travaux de traduction, de documentation et de support que la communauté peut facilement prendre en charge) et éventuellement ses fonctionnalités (le choix de la licence est ici fondamental). Enfin, l'éditeur qui souhaite développer sa communauté doit adopter un certain nombre de principes qu'il est ici difficile de détailler¹⁰⁹ : ne réaliser aucune discrimination entre les contributeurs salariés et bénévoles, adopter une transparence maximale sur la feuille de route du projet et de la prise de décisions.

3. IMPACTS DE L'OPEN SOURCE SUR LES CONTRATS

Ce chapitre décrit les bonnes pratiques en matière de la contractualisation qui entoure l'usage des licences *Open Source* – que ce soit lors du choix des licences ou lors de la rédaction du contrat « global ou final ».

Que l'objectif soit de privilégier la titularité des droits pour s'émanciper des différentes licences *Open Source* utilisées, ou de combiner des Logiciels Libres entre eux afin de créer de nouveaux logiciels, cela requiert une analyse approfondie des compatibilités de licences afin de s'assurer de la faisabilité du projet et de s'engager auprès du client en maîtrisant les risques.

Ainsi, pour contractualiser la réalisation d'un projet qui intègre une ou des licences *Open Source*, il faut distinguer deux situations :

- celle de l'auteur, éditeur qui doit choisir la licence *Open Source* adaptée à son projet (3.1.),
- celle de l'entreprise qui doit consolider la relation contractuelle incluant des licences *Open Source* (3.2.).

3.1 CHOISIR LA LICENCE OPEN SOURCE ADAPTÉE AU BESOIN

3.1.1 TACTIQUE ET TECHNIQUE DANS LE CHOIX DES LICENCES OPEN SOURCE

3.1.1.1 STRATÉGIE DANS LE CHOIX DE LA LICENCE

Le choix de la licence traduit des intérêts juridiques, techniques et commerciaux. Le choix d'une licence *Open Source* suit un processus particulièrement simple à décrire :

La société doit adopter la licence la plus adaptée à son projet. L'existence de centaine de licences laisse présager un nombre tout aussi grand de situations particulières les ayant justifiées. Leur multiplicité préjudicie à leur compréhension, alors qu'il n'y a en réalité que quelques obligations qui différencient les licences les unes des autres¹¹⁰ : les clauses *copyleft*, les clauses relatives aux usages par la communauté, les clauses relatives aux brevets, DRM, Tivoization¹¹¹, la langue utilisée, etc.

La société dresse donc, dans un premier temps, un **état des lieux** qui permet de constater que certaines licences ne peuvent pas être réutilisées en l'état en dehors de projets particuliers, que d'autres ne sont manifestement pas adaptées à ses intentions et, pour finir, que seules quelques-unes répondent en totalité ou partie à ses attentes.

La société peut encore faire le choix de la multilicence (décrite *supra*). On parle de multilicence lorsqu'une seule et même création est soumise à différentes licences : qu'elles soient toutes *Open Source* ou qu'au moins l'une d'elle le soit. Multiplier les licences *Open Source* peut parfois offrir des situations véritablement avantageuses : assurer une compatibilité au bénéfice de plusieurs licences, bénéficier de la renommée d'une licence, optimiser la cession de droits, etc.

Exemple : Le navigateur web *Firefox*, sous licence MPL/LGPL/GPL¹¹², est un très bon exemple de la souplesse

¹⁰⁹ Voir notamment : Karl FOGEL, « *Producing Open Source Software, How to Run a Successful Free Software Project* » – une traduction quasiment terminée est disponible sur le site de l'équipe Framalang.

¹¹⁰ Voir notamment Patrice-Emmanuel SCHMITZ, "Licence compatibility lists: could they compensate proliferation by developing circles of trust?", à l'occasion d'EOLE 2010, au Parlement Européen, <http://www.eolevent.eu/>

¹¹¹ Néologisme définissant la création d'un système qui inclut des logiciels libres, mais utilise le matériel électronique pour interdire aux utilisateurs d'y exécuter des versions modifiées. Source Wikipedia : <http://fr.wikipedia.org/wiki/Tivoisation>

¹¹² Après avoir été quelque temps uniquement sous MPL, il a été estimé que cette seule licence n'était pas suffisante à motiver les contributions.

qu'offrent les licences. En effet, le logiciel est construit de façon modulaire¹¹³ et gagne à conserver une licence qui intègre ce particularisme¹¹⁴. Cependant, afin de rassurer les développeurs et les communautés sur leur faculté de disposer du code pour d'autres projets, les licences LGPL et GPL s'ajoutent à la MPL. De cette façon, le logiciel libre Firefox dispose d'une licence qui lui permet d'être compatible avec les licences les plus populaires tout en bénéficiant des développements sous la licence la plus adaptée aux projets de *Mozilla* – notamment une portée réduite qui amenuise l'aspect contraignant de son *copyleft*¹¹⁵.

Dans un second temps, une fois ce premier état des lieux réalisé, la société doit se poser un certain nombre de **questions indirectement liées au projet** : la licence doit-elle être connue et reconnue par les communautés ? Doit-on s'inspirer des choix opérés pour des projets concurrents analogues ? Certaines licences doivent-elles être proscrites ? Toutes ces questions ne trouvent de réponses qu'au cas par cas, en fonction de l'examen et de l'approche choisie par la société. Incontestablement, une licence disposant d'une large notoriété participe avantageusement à la communication qui suivra la libération du logiciel. En revanche, le choix d'une licence similaire à un projet concurrent est un pari dont il faut bien mesurer les aléas : les contributions pouvant librement circuler d'un projet à l'autre, celui qui réunit la meilleure communauté risque de l'emporter et de détourner à son profit les contributeurs de son concurrent. A l'inverse, bénéficier de l'expérience et de l'expertise déployée par une société concurrente peut justifier un tel choix.

3.1.1.2 PERSONNALISATION DE LA LICENCE

Dans l'hypothèse où un logiciel est distribué sous une licence libre déjà connue il peut être opportun d'ajouter une **interprétation** relativement aux clauses des licences *Open Source*. Lorsque l'un des termes de la licence est source d'interprétation, celui qui choisit la licence peut lever l'ambiguïté qu'elle pourrait engendrer en donnant une portée bien définie à cette clause.

Ainsi, il y a création d'un faisceau d'indices qui pourrait donner des indications aux juges lors d'un litige dont la résolution passe par l'interprétation de la licence. Ainsi le juge pourrait valider l'interprétation définie par l'entreprise qui recourt à la licence. De nombreuses clauses imprécises ou équivoques peuvent être interprétées de la sorte lors du choix de la licence.

L'usage des **exceptions** est une autre technique plus radicale qui consiste à modifier la licence de base en ajoutant, dans une clause jointe à la licence ou insérée dans les en-têtes, une spécificité qui déroge aux termes initiaux avec pour effet de rendre la licence finale plus ou moins contraignante. Il est à noter qu'en l'absence de stipulation contraire, une clause additionnelle permissive pourra être supprimée par tout licencié au moment de la redistribution d'une copie du logiciel¹¹⁶.

La maîtrise de plus en plus fine de la pratique contractuelle liée aux licences *Open Source* conduit aujourd'hui à conseiller une pratique réfléchie et adaptée de ces exceptions et interprétations. Les premières confèrent plus de souplesse aux licences¹¹⁷, les secondes assurent une plus grande sécurité juridique. Un bon usage de celles-ci peut permettre de prévoir et de résoudre a priori la plupart des situations préjudiciables : problèmes d'incompatibilité, failles au sein des licences, etc.

3.1.2 TYPES DE LICENCES OPEN SOURCE

Les développements qui suivent visent à étudier de manière générale les licences *Open Source* par typologie et d'effectuer un zoom sur la GNU *General Public License*.

3.1.2.1 TYPOLOGIE DES LICENCES

La rédaction des licences *Open Source* n'étant pas toujours l'œuvre de juristes et émanant de systèmes juridiques hétérogènes, on s'aperçoit que certaines clauses sont inutiles au regard des règles de droit d'un pays, alors que d'autres doivent être interprétées pour être contextualisées.

¹¹³ Chaque nouveau composant apportant une nouvelle fonctionnalité autour d'un noyau. De nombreux projets *Open Source* adoptent cette structure qui permet notamment de répartir efficacement les tâches de développement.

¹¹⁴ En effet, la MPL permet l'adjonction de greffon (plug-in) ou toute autre extension au logiciel tant que chaque fichier contenant du code sous MPL soit exclusivement sous MPL. Cette tolérance permet une plus large diffusion du logiciel.

¹¹⁵ La conséquence logique d'une telle multilicence est d'ailleurs que le *copyleft* final est celui le moins contraignant de l'ensemble des licences.

¹¹⁶ Ceci valant bien sûr pour les licences dites permissives, mais aussi pour toute licence *copyleft* puisque le *copyleft* ne perpétue que les termes *stricto sensu* de la licence.

¹¹⁷ La troisième version de la GNU GPL invite même dans un article dédié à user de cette faculté sur ses propres contributions (en distinguant d'une part les permissions additionnelles qui peuvent être ajoutées sans limitation et les termes supplétifs à tirer parmi une liste de sept types de clauses).

Parmi les licences *Open Source* et afin d'avoir une vision plus globale du système, plusieurs classifications peuvent être croisées¹¹⁸.

1 Classification classique : licence copyleft versus permissive¹¹⁹

L'utilisation du terme **copyleft** s'entend des licences qui rendent persistantes les libertés consenties en astreignant les utilisateurs subséquents à concéder systématiquement les mêmes libertés. Concrètement, une cession non exclusive est demandée, les contributeurs restant titulaires de leurs droits et libres d'exploiter leur contribution par ailleurs.

On parle indifféremment de réciprocité ou de *copyleft*. Dans cette situation, c'est l'intérêt de l'utilisateur final qui prévaut sur la liberté de celui qui diffuse l'œuvre. Deux mécanismes sont aujourd'hui utilisés :

- imposer l'utilisation d'une licence particulière (à l'instar de la GNU GPL, à son propre compte)
- obliger à conférer les mêmes libertés (comme le fait la LAL¹²⁰ : les droits cédés devront se retrouver dans la licence finale — qu'il s'agisse de la LAL ou de toute autre). Il en résulte une relation de confiance qui sécurise et favorise les collaborations entre professionnels.

À l'inverse, on parle de licence **permissive** lorsque seules les obligations de celui qui reçoit l'œuvre — comme celles en matière de brevets pour la licence Apache — doivent être transmises, laissant le contributeur libre d'en ajouter d'autres lors du transfert aux utilisateurs ultérieurs (le logiciel redistribué perd souvent les libertés qui lui étaient attachées). Ces licences sont traditionnellement assimilées à des renoncements et le statut des œuvres est souvent proche de celui des œuvres tombées dans le domaine public, puisqu'elles n'imposent en règle générale que le respect de la paternité (avec les habituelles clauses d'exclusion ou limitation de garanties et de responsabilité). Cette relative liberté les fait coexister sans anicroche puisqu'il est très simple pour ces licences d'être compatibles en perpétuant simplement les obligations initiales.

2 Classification historique

• **Les licences GNU / philosophiques** : il s'agit des licences publiées par la *Free Software Foundation* et plus généralement toutes celles qui partagent son esprit et sa philosophie. Les plus utilisées sont celles de la FSF : la première de la famille est la *GNU General Public License* — publiée en 1989, modifiée en 1991 et 2007, sa petite sœur est la *GNU Library General Public License* — renommée *lesser GPL* — et leur cousine à destination de la documentation, la *GNU Free Documentation licence*. Il y a deux nouvelles entrantes : la *GNU Affero GPL* depuis fin 2007 et la *Simpler FDL* qui devrait suivre prochainement. À l'origine, le langage de ces licences, très proche de celui des développeurs, est empreint d'une intention forte et d'une portée parfois complexe à définir. La réécriture récente des licences ajoute à cette famille de licences des versions beaucoup plus juridiques et complexes. Elles s'opposent à toute réappropriation du code grâce à leur *copyleft* qui impose que tout logiciel dérivé — basé sur, ou constituant un tout avec le logiciel — soit lui-même soumis à cette licence. **Les sociétés intéressées par l'alternative du « Libre » hésitent souvent à recourir à l'utilisation de ces licences aux implications extensives et parfois incertaines.**

• **Les licences académiques / universitaires.** Elles sont en large partie à l'origine du développement de l'infrastructure d'Internet. Pour exemple, le système de nom de domaine BIND, le protocole TCP-IP et Sendmail sont tous des standards de *facto*, issus de ces licences permissives. On y retrouve l'idée d'un partage des connaissances « sans condition » issu des universités américaines. Elles sont fréquemment formulées d'une façon courte et claire. Elles consistent généralement en l'énumération de la totalité des droits conférés, une obligation de respecter la paternité de l'œuvre et une exclusion de responsabilité et de garantie. Un bon exemple est la licence BSD — pour *Berkeley Software Distribution*.

• **Les licences communautaires.** Elles sont principalement issues de projets libres qui, devenus populaires, choisissent de rédiger et d'utiliser leur propre licence. Très spécifiques puisqu'intimement liées à un projet et à son déroulement, elles sont en principe peu juridiques et trop souvent susceptibles d'interprétations hasardeuses. Les deux principales sont la licence *Artistic* et la licence Apache. Ce sont essentiellement des licences permissives, mais leur spécificité les rend difficiles — voire impossibles — à concilier avec la plupart des licences *copyleft*.

• **Les licences institutionnelles** : Elles furent introduites par des sociétés intéressées par le développement coopératif de leur produit selon le modèle de l'*Open Innovation*. La première, la plus symptomatique est la *Mozilla*

¹¹⁸ Une description des familles de licences Open Source et de leurs grandes caractéristiques figurent dans un tableau reproduit en Annexe 3.

¹¹⁹ Voir : (petit) Guide à l'usage des licences libres, description des différents types de licences libres et des conséquences qui leur sont attachées, Intervention de Benjamin JEAN, lors de la Matinée Juridique du Syntec Informatique du 14 mars 2008.

¹²⁰ Licence Art Libre, Voir : <http://artlibre.org/>

Public License (MPL), rédigée par la firme Netscape pour la libération du code de son navigateur. Ces licences sont précises et très complètes, ont une étendue définie et emblématique du mouvement *Open Source*.

3 Par domaine

Les licences *Open Source* sont aujourd'hui utilisées dans de nombreux domaines : les logiciels bien sûr, mais aussi les encyclopédies (on pense à Wikipedia), les livres, la musique et bientôt tout type d'œuvres. La majeure partie des licences *Open Source* trouve leur fondement dans une application particulière pour un domaine artistique bien déterminé. De ce fait, il est très délicat et déconseillé d'utiliser des licences rédigées pour un domaine dans un autre. Par exemple, le formalisme extrêmement contraignant de la GNU GPL n'est pas du tout adapté à la musique (le texte entier de la licence doit être attaché ; et imaginez qu'il faille distribuer le master et les sources en plus du CD...).

Inversement, des licences comme les Creative Commons ont été écrites pour la musique, le film, les livres, etc., mais pas pour le logiciel. Très logiquement, licencier un logiciel sous *Creative Commons* — par exemple une CC-By-SA pour prendre une licence *Open Source* assez similaire à la GNU GPL — n'est pas conseillé puisque tous les aspects propres au logiciel sont inexistantes : la distribution du code source n'étant notamment pas envisagée.

Enfin, quelques licences, plutôt rares, ont été rédigées avec l'ambition de s'étendre à l'ensemble des créations couvertes par le droit d'auteur, voire toute la propriété littéraire et artistique : l'OSL et la LAL sont deux très bons exemples avec des origines bien différentes.

Il est néanmoins nécessaire dans certaines situations, notamment en présence d'œuvres dites multimédias — regroupant de nombreux types d'œuvres — de réfléchir à l'opportunité d'utiliser une licence pour le tout dans un objectif d'uniformisation ou de privilégier une approche modulaire (en utilisant une licence pour chaque type d'œuvre).

Il n'y a encore une fois ni de bonnes ni de mauvaises réponses : tout dépend de l'espèce.

4 Par liberté

Sous la coprésidence de Valérie-Laure BENABOU et Joëlle FARCHY, la commission spécialisée du CSPLA portant sur la mise à disposition ouverte des œuvres de l'esprit a publié en juin 2007 un rapport reprenant une distinction issue des travaux de thèse sur les œuvres libres de Mélanie CLEMENT-FONTAINE. On y retrouve la classification selon le degré de libertés conférées :

- **Les licences qui offrent une liberté pérenne.** Il s'agit des licences disposant d'un *copyleft* : l'œuvre et ses dérivées sont libres ;
- **Les licences qui offrent une liberté fragile.** Il s'agit ici des licences permissives qui autorisent la propriétérisation par un tiers d'une œuvre dérivée — les contributeurs n'ont donc aucune garantie de pouvoir à leur tour bénéficier des contributions ultérieures ;
- **Les licences qui offrent une liberté asymétrique.** Il s'agit ici des licences qui créent un déséquilibre au profit de celui qui a initialement le choix de la licence : avec l'exemple de la licence CC-By-NC qui interdit l'exploitation commerciale de l'œuvre *Open Source*, mais présente une souplesse supérieure dans la réutilisation.

3.1.2.2 L'EXEMPLE DE LA GNU GPL¹²¹

Pour comprendre l'évolution des licences *Open Source*, il faut suivre l'évolution de la GNU *General Public Licence*.

S'appliquant évidemment au logiciel et à la documentation technique l'accompagnant, la licence GPL fixe les conditions de diffusion : les logiciels sous licence GPL peuvent librement être copiés, adaptés et distribués de quelque manière que ce soit, à la seule condition de rester soumis aux mêmes termes. La question de l'interprétation de la GNU GPL v2 pose problème en ce qu'elle émane d'un système de droit anglo-saxon, mais la troisième version se détache de ce droit au profit d'une terminologie contractuelle plus conforme au droit français. L'objectif est de créer un « *fonds commun auquel chacun peut ajouter, mais duquel personne ne peut retrancher* », comme aime à le répéter Eben Moglen, l'avocat de la FSF et directeur de la SFLC.

- Le code source du logiciel diffusé sous licence GPL est communiqué à l'utilisateur. Mais une condition s'impose : le code source, auquel l'utilisateur a eu accès, doit être également transféré en cas de diffusion. Cet utilisateur ne saurait de sa seule initiative, et donc sans l'autorisation du créateur du logiciel, distribuer celui-ci sans son code source. En d'autres termes, il ne saurait introduire, dans la chaîne de diffusion du logiciel, une forme d'appropriation de *facto* en se réservant le code source. Il irait ainsi contre la volonté du premier programmeur et ne respecterait pas les termes de la licence.

¹²¹ Voir la version originale sur <http://www.gnu.org/copyleft/gpl.html> et l'analyse de ses différentes versions sur <http://www.vvlibri.org/>

- Compte tenu de la possibilité qui lui est faite d'accéder librement au code source, l'utilisateur peut modifier celui-ci, voire modifier le logiciel dans son entier. C'est ici une considérable exception apportée à l'interdiction de la décompilation traditionnelle, comme nous le verrons plus loin, interdiction visant précisément à empêcher toute modification du code source. Par contre, cette possibilité d'examen et de modification du code source implique une obligation : l'utilisateur doit préciser dans une notice les adaptations dont il est l'auteur, la date de ses apports, et son identité. Chaque programmeur successif participe à l'enrichissement du logiciel, et par principe son droit au nom sera respecté.
- Un logiciel ainsi modifié peut également être distribué, mais cette nouvelle diffusion doit également respecter les stipulations de la licence GPL : elle doit être diffusée sous les mêmes termes : une fois qu'un logiciel a été cédé en vertu de ses stipulations, il est interdit de réintroduire un quelconque élément « propriétaire », c'est-à-dire qu'on ne peut se réserver le résultat d'une modification, ni la diffuser à titre onéreux, ni même introduire des portions de code faisant l'objet d'une réservation (par le brevet ou le droit d'auteur) dans les adaptations successives, dès lors que l'on distribue ces dernières.

Cette licence se veut la plus ouverte, mais comporte bel et bien des obligations pour l'utilisateur :

- **Obligation**, pour toute redistribution et toute modification du logiciel initial, d'adopter la licence GPL et de la diffuser en même temps que le logiciel ;
- **Obligation** pour un programmeur travaillant à partir d'un logiciel libre cédé sous cette licence, de soumettre sa création dérivée à la licence GPL ;
- **Obligation** de porter à la connaissance des tiers et de tout cessionnaire le nom des programmeurs qui sont déjà intervenus sur le code source ;
- **Obligation** d'informer tout cessionnaire ultérieur du logiciel ou de ses modifications des règles de la licence GPL.

Deux vagues successives (la première le 29 juin, et la seconde le 9 novembre 2007), ont respectivement donné lieu à publication des nouvelles versions¹²² des GNU GPL (v3), GNU LGPL (v3) et de la GNU Affero GPL (v3) (dite aussi AGPL).

Sur la forme, les trois licences s'affichent avec une compatibilité parfaite au profit de la GNU GPL et s'offrent une modularité qui permet de faciliter leur utilisation conjointe. Si les améliorations ont pesé sur la lisibilité des licences, la FSF a fait attention à conserver leur similitude. Ainsi, la GNU LGPL apparaît finalement comme une exception à la GNU GPL (en se concentrant sur les droits supplémentaires accordés, et renvoyant à la GNU GPL pour le reste de ses termes). De même, l'Affero GPL ne se distingue de la GNU GPL que par une clause qui lui permet de s'adapter aux spécificités des logiciels voués à être utilisés via le réseau sans avoir à être distribués. Ces derniers devront alors permettre aux utilisateurs de bénéficier de la version du logiciel utilisée par ledit service, comblant ainsi ce que l'on avait dénommé l'« ASP *loophole* ». Enfin, un travail de rédaction a aussi permis d'internationaliser les licences et d'assurer leur validité pour l'ensemble des systèmes juridiques.

- Sur le fond, les changements sont nombreux et ne peuvent qu'être listés ;
- Une « obligation de cohérence » interdit à l'utilisateur de limiter par d'autres voies, ou droits exclusifs concurrents, les libertés qu'il concède par la licence : que ce soit à l'égard des Mesures Techniques de Protection, des limitations matérielles (Tivoïsation) ou encore des brevets ;
- Un encadrement des accords parallèles portant sur des brevets¹²³ agit tant du côté du promettant (qui s'obligerait à ne pas opposer ses brevets) qu'à celui qui en bénéficierait ;
- Une modularité est incorporée afin de permettre d'intégrer d'autres briques logicielles sous licence *a minima* libre (disposition profitant à la compatibilité avec des licences comme LaTeX, etc.) ;
- Des adaptations technologiques (comme la distribution par peer to peer).

Garante du rôle qu'elle s'est donnée, la *Free Software Foundation* assure par ces mises à jour la pérennité des libertés qu'elle prône, sans aucune concession.

¹²² Voir la « *Rétrospective 2007 sur le logiciel libre et les sujets afférents* » publiée par l'April : <http://www.april.org/articles/divers/retrospective-2007.html#ToC23>

¹²³ En réponse à l'accord Novell/Microsoft

3.1.3 ENJEUX DE LA COMPATIBILITÉ ENTRE LICENCES¹²⁴

3.1.3.1 LICENCE UNIQUE

La question « particulière » est celle de la **compatibilité des licences**. Si la démarche qui précède permet de s'orienter vers le choix optimal d'une licence, celle-ci peut se révéler impossible à mettre en œuvre si le produit « final » intègre des composants soumis à d'autres licences, aux termes incompatibles lors de la distribution ou de l'utilisation du logiciel.

Il n'existe pas de *facto* de licences dominantes : toutes ont la même force obligatoire et le licencié doit renoncer à exploiter une création lorsqu'il ne peut pas respecter l'ensemble de ses engagements. La seule condition pour distribuer sous licence unique un projet comprenant des composants sous d'autres licences¹²⁵ est que cette distribution soit permise par l'ensemble des autres licences.

Après avoir été un frein majeur aux échanges entre projets libres pendant de longues années, le problème de l'incompatibilité entre licences semble se résorber progressivement grâce aux licences « de nouvelles générations » :

- La situation la plus commode réside en la **compatibilité expresse** : soit la licence prévoit que le logiciel peut indifféremment être redistribué selon ses propres termes ou selon ceux d'une licence déterminée¹²⁶, soit la compatibilité est limitée aux seules situations où l'ajout d'un composant sous une autre licence *Open Source* impose de redistribuer le tout sous cette unique autre licence¹²⁷.

- En l'absence d'une telle compatibilité, il convient de procéder à une analyse logique du contexte : l'enjeu consiste à gérer les flux de droits et d'obligations de la société pour savoir ce qu'elle peut faire ou ne pas faire : **compatibilité induite**. Lorsque l'on souhaite céder une contribution pour laquelle on est seulement licencié, la question se résume à s'assurer que l'on est bien en mesure de céder tous les droits que la licence confère et à veiller que l'on n'oblige pas moins l'utilisateur final que l'on est soi-même obligé¹²⁸.

La dernière version de la GNU GPL (V. 3) s'appuie sur ce principe pour accueillir en son sein une modularité qui lui permet d'assurer la compatibilité à l'égard d'autres licences *Open Source* auparavant incompatibles¹²⁹ : en permettant, sur chaque nouvelle contribution, d'une part, l'ajout de « permissions additionnelles » et, d'autre part, l'ajustement par l'utilisation de certains « termes supplétifs » limitativement autorisés. Le problème n'est néanmoins pas tout à fait résolu puisqu'il faudra ensuite confronter chaque licence aux différentes clauses, au cas par cas, afin d'être certain de leur compatibilité¹³⁰.

Cette réflexion peut se traduire mathématiquement grâce à la théorie des ensembles sous la forme qui suit¹³¹ : la licence B est dite compatible (c'est-à-dire qu'elle peut être utilisée pour redistribuer le logiciel en respectant les termes de la licence originaire A) **si et seulement si** l'ensemble des droits de la licence B est inclus dans la licence A et que l'ensemble des obligations de la licence A est inclus dans la licence B. Ainsi, une licence est compatible avec elle-même, une licence *copyleft* n'est pas forcément compatible avec une licence permissive et une licence permissive ne l'est forcément pas à l'égard d'une licence *copyleft*.

Attention, la version des licences constitue dans certains cas un élément déterminant : même si la plupart permettent une compatibilité ascendante au bénéfice des nouvelles versions de la licence, certaines autorisent aussi aux auteurs de licences *Open Source* de figer la licence à une version spécifique¹³². Il y a alors un *copyleft* en deux temps : le premier caractérisant l'étendue de la licence et le second déterminant la version sur chaque contribution.

La situation la plus confortable consiste à rester dans la même « famille de licences » : il en existe plusieurs et celles-ci couvrent en théorie un large panel de choix tout en assurant une compatibilité parfaite entre elles¹³³.

¹²⁴ Sur cette question, voir notamment : Benjamin JEAN, « Option Libre : compatibilité entre contrats », sous la direction de Michel VIVANT, ERCIM, 2006

¹²⁵ Par exemple, distribuer sous GNU GPL v3 une solution qui comporte des composants sous licence BSD et Apache.

¹²⁶ C'est notamment ce que faisait la LGPL v2 au bénéfice de la GNU LGPL.

¹²⁷ Ce que l'on retrouve notamment dans l'European Public Licence au profit de cinq licences que sont l'OSL v. 2.1 et v. 3.0, la CPL v. 1.0, l'EPL v. 1.0, la CeCILL v. 2.0 et la GNU GPL v. 2.0

¹²⁸ Cette compatibilité est aujourd'hui de mieux en mieux expliquée, comme le démontre un papier récent de la Software Freedom Law Center qui détaille la procédure permettant de conserver la licence BSD sur du code distribué sous Licence GNU GPL.

¹²⁹ Comme les licences plus permissives comme les licences Apache, LaTeX, etc.

¹³⁰ Avec le risque de divergence d'interprétation, comme c'était déjà le cas les licences GPL et Apache.

¹³¹ Sur les questions de compatibilité entre licences, voir Benjamin JEAN, Option Libre : « compatibilité entre contrats », Op. Cit., 2006.

¹³² Si on prend l'exemple de la GNU GPL et du noyau Linux : toutes les contributions de Linus TORVALD sont distribuées sous GNU GPL « version 2 seulement », ainsi, même si d'autres parties autorisent la distribution sous GNU GPL version 3, le noyau Linux ne pourra pas être distribué sous la troisième version de la GNU GPL tant que les composants sous GPL v2 n'auront pas été réécrits ou que leur auteur aura consenti à modifier la licence !

¹³³ La plus connue est certainement la famille des licences GNU (GPL, LGPL, AGPL, etc.), mais bien d'autres peuvent être citées pour l'exemple : CeCILL (-A, -B, -C), OSL (OSL, AFL), etc.

3.1.3.2 CAS D'INCOMPATIBILITÉ ET SOLUTIONS ALTERNATIVES

Certains conflits ne peuvent contractuellement être résolus. Si l'œuvre litigieuse s'avère trop complexe pour être réécrite, il faudra alors renoncer à la distribution ou à l'utilisation du tout¹³⁴.

Il reste néanmoins quelques autres solutions : contacter le ou les auteurs d'un composant litigieux pour leur demander une dérogation, une exception ou un changement de licence ; acheter une licence commerciale à ces derniers ; se faire céder la titularité de leurs droits¹³⁵ ou contourner matériellement le problème posé par la licence.

Le premier expédient possède l'avantage de la simplicité, mais son succès dépendra du nombre d'auteurs du composant, de leur personnalité et de ce que l'entreprise est prête à investir dans le projet. Les deux autres mesures dépendront aussi totalement de la volonté des titulaires de droits. Le dernier procédé se révèle le plus précaire puisqu'il nécessite une connaissance parfaite des licences afin d'estimer ce qui peut matériellement être permis et ce qui ne le serait pas (certaines formes de distribution¹³⁶, certains modes de distribution¹³⁷ et certaines constructions du logiciel¹³⁸ permettent par exemple d'alléger les contraintes).

En résumé, le processus visant à vérifier la compatibilité expresse entre une licence *Open Source* A (d'origine) et une licence B (licence finale) est le suivant :

- Considérons que les droits ne posent pas de problème puisqu'a minima la Licence A est libre.
- Penchons-nous sur les obligations :
 - 1) la licence B contient toutes les obligations de la licence A - modulation des variations éventuellement tolérées par la licence,
 - 2) la licence A n'oblige pas à la redistribution sous une licence particulière OU elle permet expressément la redistribution sous la Licence B (ou une licence qui lui soit compatible).
- Considérons une licence *copyleft* comme une licence qui a comme obligation de conserver les droits lors de la redistribution, alors : si la licence A est une licence *Open Source copyleft*, la licence B devra aussi être libre et *copyleft*.

Plus généralement, la condition de compatibilité est donc atteinte si l'ensemble des droits accordés par la licence absorbante B est inclus dans l'ensemble des droits conférés par la licence compatible A et que l'ensemble des obligations imposées par la licence compatible A est inclus dans l'ensemble des obligations imposées par la licence absorbante B.

3.2 RÉDIGER LE CONTRAT ADAPTÉ À LA LICENCE

Après avoir adopté une démarche permettant de sécuriser la licence du produit vendu, il est impératif de sécuriser le contrat global de services. L'attention doit porter sur deux aspects :

- La propriété du logiciel et des droits de propriété intellectuelle (logiciel en tant que création),
- Les services offerts, la description de(s) l'engagement(s) pris et l'étendue de la responsabilité.

Par conséquent, pour rédiger les clauses relatives aux garanties, à la maintenance, à la propriété, il est indispensable de se référer aux licences d'utilisation des logiciels composant la solution vendue au client. Pour maîtriser et comprendre les contrats de services, il faut connaître :

- L'étendue du droit d'auteur sur le logiciel rappelé en introduction du présent document,
- La particularité des licences *Open Source* qui organisent une cession de droits d'auteur **non exclusive** et **gracieuse** afin d'opérer un partage de l'exploitation d'une œuvre. La non exclusivité des cessions détermine tout le système

¹³⁴ Sauf bien sûr à trouver un remplacement pour la partie litigieuse.

¹³⁵ On utilise souvent l'appellation américaine de *Copyright assignment* pour désigner ces cessions de droits. Il en existe de multiples sortes : La cession pure et simple sans contrepartie (par exemple comme condition à l'intégration d'une contribution dans un projet) ou les cessions finalisée (c'est-à-dire que l'utilisation par le cessionnaire est clairement délimitée par la cession : par exemple, toute distribution dont la contribution doit être réalisée sous licence open source).

¹³⁶ Par exemple exécutable, tant que les sources des modifications sont disponibles par ailleurs.

¹³⁷ On pense notamment aux logiciels dont seul le client est distribué alors que le serveur est hébergé chez l'éditeur.

¹³⁸ Par exemple la construction modulaire pour l'utilisation de composants sous licence du type MPL, voire LGPL.

contractuel afférent aux licences *Open Source*.

Une précision s'impose : la rémunération traditionnellement accordée à l'éditeur du logiciel dans le schéma dit « propriétaire » se trouve transférée vers l'entreprise qui offre un service. La valeur est transférée de la propriété incorporelle vers le service.

C'est l'aménagement contractuel de ce service qui nous intéresse ici.

Les demandes du client reposent souvent sur des confusions ou une méconnaissance de la propriété intellectuelle en matière de logiciel (Cf. *supra* 1.2.). Bien communiquer sur les droits de propriété intellectuelle relatifs au logiciel et, en particulier au logiciel libre sur la nature et l'étendue des droits conférés au client, facilite la négociation des projets. Cette communication est un préalable indispensable à toute relation contractuelle et ceci d'autant plus que tout prestataire est tenu d'informer son cocontractant (obligation d'information).

Les développements qui suivent se concentrent sur les principales clauses des contrats de services et sur les éléments à mettre en exergue lors des négociations contractuelles.

De plus, soulignons que les licences *Open Source* contiennent souvent des clauses qui ne relèvent pas de la licence au sens strict (déterminer l'étendue du droit d'utiliser le logiciel), mais du contrat commercial (service offert, garantie...).

3.2.1 LA CLAUSE « DÉFINITIONS »

Il convient d'ajouter à la clause « définitions » ou, de préciser dans le contrat, les définitions suivantes :

- **Logiciel Libre** : un logiciel est dit libre, ou *Open Source*, lorsque son code est publié et disponible pour être étudié et utilisé par celui qui dispose d'une copie, et qu'il est soumis à une licence *Open Source*.
- **licence *Open Source*** : contrat de cession non exclusive de droits d'auteur permettant gracieusement au licencié de copier, modifier et distribuer le logiciel et son code source. Les licences certifiées par l'*Open Source Initiative*, et publiées comme telles sur son site internet (<http://www.opensource.org>) en date du <.....>, sont considérées comme conformes à cette définition.
- **Licence Compatible** : est dite compatible toute licence de logiciel qui se substitue à une autre en respectant l'ensemble de ses termes lors de la distribution du logiciel ; elle permet généralement de distribuer un ensemble de composants logiciels sous une seule licence.

3.2.2 LA GARANTIE D'ÉVICTION

3.2.2.1 LE PRINCIPE

La **garantie d'éviction** ou **garantie de l'utilisateur en cas de contrefaçon** implique l'obligation pour le concédant (éditeur, société de services, etc.) de ne pas perturber, par son fait personnel, le licencié dans la jouissance de la chose cédée et de faire en sorte qu'aucun tiers ne vienne perturber cette libre utilisation.

La garantie d'éviction est légale, en droit français. Ce qui signifie qu'elle s'applique même si elle n'est pas stipulée dans le contrat de cession. Elle consiste pour le concédant à garantir le cessionnaire (bien souvent le client) contre toute réclamation d'un tiers prétendant que le logiciel licencié/cédé porte atteinte à ses propres droits de propriété intellectuelle (notamment en cas de contrefaçon).

- L'éditeur d'un logiciel est en mesure de garantir qu'il est titulaire de tous les droits de propriété intellectuelle applicables aux modules qu'il a intégré ou qu'il a obtenu les droits des tiers lui permettant d'intégrer le module à sa solution et de les redistribuer ensembles.
- Lors de l'utilisation de composants *Open Source*, cette garantie n'est techniquement pas possible. L'ensemble des licences *Open Source* prévoient le transfert du logiciel « en l'état », c'est-à-dire sans garantie quelle qu'elle soit.

Un contributeur peu scrupuleux peut avoir inséré du code propriétaire, parfois protégé par un brevet, dans la solution développée. Il diffusera ce code « contrefaisant » sous licence *Open Source*, sans respecter le monopole d'exploitation de son auteur. Faute de pouvoir maîtriser la chaîne des contributeurs, les licences *Open Source* ne contiennent aucune garantie pour le licencié contre une éviction provenant d'un tiers. **Dans les faits, les logiciels libres commercialisés sont développés par des communautés organisées par des sociétés commerciales qui encadrent ce risque.**

Pour conclure, il faut admettre que les risques attachés à l'utilisation de composants *Open Source* sont réels. Cela dit, le risque est identique dans le modèle propriétaire. Il convient juste de le gérer¹³⁹.

Si la garantie d'éviction est légale pour être effective encore faut-il qu'un contrat soit conclu avec un intermédiaire, assuré et responsable, telle qu'une société de services informatiques. Le risque du client est donc couvert par l'éditeur et/ou la société de services.

3.2.2.2 DANS LE CONTRAT DE SERVICES

L'éditeur doit cette garantie légale qu'est la garantie d'éviction. Il déclare être le légitime détenteur de tous les droits de propriété intellectuelle qu'il cède par contrat.

À ce titre, l'éditeur garantit expressément au Client la jouissance pleine et entière des droits qu'il a cédés aux termes du Contrat contre tout trouble, revendication, éviction ou réclamation quelconques.

“L'éditeur garantit le Client contre toutes réclamations, oppositions relatives au Produit, émanant de tout tiers invoquant la violation d'un droit quelconque cédé par le contrat, et notamment contre toute action en contrefaçon et/ou en concurrence déloyale et/ou parasitaire intentée par tout tiers. Il en supportera tous les frais et dommages-intérêts y afférent.

Dans le cas où l'interdiction d'utilisation du produit serait prononcée en conséquence d'une telle action ou résulterait d'une transaction signée avec le demandeur de l'action pour de telles réclamations, l'éditeur devra à ses frais :

- *Obtenir le droit pour le Client de poursuivre l'utilisation du Produit,*
- *A défaut remplacer ou modifier le produit de façon à écarter ladite action, tout en conservant le même niveau de fonctionnalités, de performance et de pertinence,*
- *A défaut rembourser au Client le montant total global du Contrat.*

Ces dispositions survivront en principe à la résiliation ou à l'expiration du contrat, quelle qu'en soit la cause.”

3.2.2.3 LES CONSÉQUENCES LIÉES AUX SOUS LICENCES (LIBRES)

La chaîne contractuelle qui lie le Licencié au(x) titulaire(s) de droits peut être de deux sortes :

- soit directe et unique lorsque les licences *Open Source* autorisent les sous-licences (chaque détenteur subséquent de l'œuvre se voit céder le droit de céder),
- soit distribuée lorsque les licences impliquent que chaque titulaire de droit cède au licencié final les droits relatifs à leur propre contribution.

Il n'y a pas de consensus sur le sujet et il est possible de trouver les deux types de licence. Par exemple, l'OSL 3.0 est clairement dans la première catégorie (chaîne unique) alors que les GNU GPL (version 2 ou 3) opèrent une distribution.

Plusieurs conséquences sont directement liées à ces enjeux :

- Lorsque la licence met en place un mécanisme de sous-licence, tout membre de la chaîne peut agir en justice en cas de non respect de la licence et, en cas de rupture de la chaîne de cession des droits, l'utilisateur final qui serait inquiété pourrait se retourner pour le tout contre chacun d'entre eux ayant cédé la partie du logiciel qui est litigieuse (à leur charge ensuite de remonter la chaîne).
- En l'absence de sous-licence, chaque contributeur cède les droits sur sa seule contribution sur le logiciel et, dans le cas où l'une des contributions serait contrefaisante, seul le contributeur de cette dernière pourra se faire assigner.

¹³⁹ Voir notamment sur ce sujet, l'intervention de Sandrine RAMBAUD lors de l'European Opensource Law Event 2008 : « GPLv3 – Warranties and Liability ».

3.2.3 GARANTIE CONTRACTUELLE ET MAINTENANCE

Il faut bien distinguer :

- La correction des anomalies qui s'inscrit dans le cadre de l'obligation de conformité et qui est une obligation essentielle du contrat de développement (la plupart du temps couverte par une garantie de courte durée correspondant à une maintenance gratuite),
- Des obligations de maintenance ou d'assistance.

Par ailleurs, il existe toujours une obligation technique quant à la performance du logiciel (et conformité au cahier des charges) qui n'est pas spécifique au « Libre ».

3.2.3.1 LES GARANTIES TECHNIQUES SUR LE LOGICIEL

Les licences propriétaires contiennent fréquemment des clauses de garantie. Ces clauses sont en principe valides. Les licences *Open Source* quant à elles comportent souvent des clauses de non garantie, sauf dispositions légales contraires. En effet, dans le modèle libre, et compte tenu de la pluralité des contributeurs et de la rédaction de la licence, il n'est pas possible de conférer aux clients les garanties classiques. En ce qui concerne les licences GNU, la notion est même incompatible.

Dans les cas où la licence ne contient pas de garantie, le client qui conclut un contrat de services souhaite obtenir une garantie.

Lorsque les entreprises recourent à une société de services (intégrateurs...) auxquels elles confient l'installation, le paramétrage, l'adaptation et la maintenance du logiciel, la différence entre logiciel libre et logiciel propriétaire s'efface. Le niveau de garantie et de responsabilité sera alors négocié auprès de la société de services afin d'assurer la sécurité juridique et économique du projet.

3.2.3.2 LA MAINTENANCE

Afin de palier le risque de fork, il est possible de demander au prestataire de garantir au client un support qui réponde à la demande de ce dernier tout en restant cohérent avec le développement communautaire du logiciel.

De SSLL proposent de la tierce maintenance applicative appliquée aux Logiciels Libres. Elles disposent des compétences permettant d'utiliser les outils de développement collaboratifs utilisés par les communautés et peuvent donc assurer un haut niveau de support voire d'adaptation aux besoins spécifiques.

Plus classiquement, certaines entreprises définissent les niveaux d'anomalies et interviennent en fonction du niveau concerné.

ANNEXE 1 : LEXIQUE TECHNIQUE ET JURIDIQUE

Bibliothèque : (Souvent appelé « librairies » en raison du faux ami anglais) En informatique, une bibliothèque ou librairie logicielle (ou encore, bibliothèque de programmes) est un ensemble de fonctions utilitaires, regroupées et mises à disposition afin de pouvoir être utilisées sans avoir à les réécrire. Les fonctions sont regroupées par leur appartenance à un même domaine conceptuel (mathématique, graphique, tris, etc). Les bibliothèques logicielles se distinguent des exécutables dans la mesure où elles ne représentent pas une application. Elles ne sont pas complètes, elles ne possèdent pas l'essentiel d'un programme comme une fonction principale et par conséquent ne peuvent pas être exécutées directement. Les bibliothèques peuvent regrouper des fonctions simples (par exemple le calcul d'un cosinus, ou l'inversion d'une matrice) comme des fonctions complexes, avec de nombreuses fonctions internes non accessibles directement. L'intérêt des bibliothèques réside dans le fait qu'elles contiennent du code utile partageable que l'on ne doit pas avoir à réécrire à chaque projet.

Composant : Matériau constituant le logiciel

Code source : Suite d'instructions en un langage informatique. Ce programme est compilé sur la machine en un code objet, puis lié avec des bibliothèques en un code exécutable aussi appelé programme exécutable.

Le code source (ou les sources, voire le source) est un ensemble d'instructions écrites dans un langage de programmation informatique de haut niveau, compréhensible par un être humain entraîné, permettant d'obtenir un programme pour un ordinateur.

Composants : http://fr.wikipedia.org/wiki/Composants_communs

CPI : Code de la Propriété Intellectuelle

Documentation : Ensemble de documents élaborés par l'Éditeur de progiciel et le fournisseur de système et comprend notamment le guide utilisateur et la documentation technique.

Forge : En informatique, une forge désigne un système de gestion de développement collaboratif de logiciel¹⁴⁰.

Fork : Parfois nommé embranchement. L'utilisation, parfois critiquée à bon escient, de l'anglicisme *fork* dans le contexte de projet informatique est une utilisation imagée du mot fork utilisé en programmation : on crée un nouveau projet à partir d'un autre à l'identique, sans détruire celui-ci. Cela suppose que les droits accordés par les auteurs le permettent : ils doivent autoriser l'utilisation, la modification et la redistribution du code source. C'est pour cette raison que les embranchements se produisent facilement dans le domaine des logiciels libres¹⁴¹.

Gouvernance : Ensemble de méthodologies, méthodes et outils utilisés pour contrôler un processus

Licence compatible : Est dite compatible toute licence de logiciel qui se substitue à une autre en respectant l'ensemble de ses termes lors de la distribution du logiciel ; elle permet généralement de distribuer un ensemble de composants logiciels sous une seule licence.

Licences copyleft (« gauche d'auteur » jeu de mots par opposition à droits d'auteur) : l'auteur d'une adaptation s'engage à ce que son adaptation soit elle-même libre de droits. Alors que l'auteur d'une adaptation d'un logiciel non *copylefté* peut protéger le programme dérivé sans avoir à reverser quoi que ce soit à la communauté du logiciel premier.

Module : Un module en programmation désigne un espace de noms. Chaque module peut exporter ou importer certains symboles comme des variables, des fonctions ou des classes. Les modules peuvent se regrouper en package éventuellement hiérarchique¹⁴².

Référentiel : « Ensemble de bases de données contenant les « références » d'un système d'information »¹⁴³.

Versions : Un logiciel est susceptible de changer de forme, car il connaît différentes versions, par traduction de langage, par évolution de ses fonctionnalités, par adaptation à son environnement matériel et aux besoins des utilisateurs. Tant que la création originale est reconnaissable sous les évolutions, il s'agit d'une seule et même œuvre.

¹⁴⁰ Voir http://fr.wikipedia.org/wiki/Forge_%28informatique%29

¹⁴¹ Voir http://fr.wikipedia.org/wiki/Fork#Embranchement_d.27un_projet_informatique

¹⁴² Voir : http://fr.wikipedia.org/wiki/Module_%28programmation%29

¹⁴³ http://fr.wikipedia.org/wiki/R%C3%A9f%C3%A9rentiel_%28BDD%29

ANNEXE 2 : FICHE SYNTHETIQUE DECRIVANT UNE LICENCE ANNEXEE AU CONTRAT

Licence X		
Informations		
	Auteur	
	Date de création	
	Langue	
	Version	
	Traduction	
Domaine (logiciel, documentation, etc.)		
Liberté		
	Libre	
	Interdit la commercialisation	
	Interdit la modification	
Obligations		
	Obligation de céder les mêmes droits (Copyleft)	
	Marques	
	Brevets	
	Paternité	
	Autre	
	Obligations additionnelles tolérées	
Portée de la licence		
	Type	
	Exceptions	
Licence de redistribution		
	sous la même licence	
	Possible sous une autre licence	
	Sous certaines licences mentionnées	
Condition de réutilisation		
	Mention du copyright	
	Citation des auteurs	
	Conditions relatives à la marque	
	Autre	

ANNEXE 3 : ANNEXE CONTRACTUELLE DE DESCRIPTION D'UNE FOURNITURE LOGICIELLE

NOM ET DESCRIPTIF FONCTIONNEL SOMMAIRE

Il s'agit ici de mentionner :

- Le nom du logiciel ou de l'application livrée
- Le niveau de version (ou d'état technique, niveau de tag, etc.) livré
- Une description fonctionnelle sommaire (laquelle pourra renvoyer aux spécifications fonctionnelles)
- La liste des principaux modules ou composants, surtout quand ces modules font l'objet d'une facturation additionnelle. Cette liste doit correspondre notamment aux éléments de facturation pour limiter les risques de litige en matière de recette et donc de règlement. Cette liste pourra également comprendre un tableau de correspondance avec la liste des répertoires et fichiers de la livraison, si le nommage des répertoires de sources et/ou de livrable de ces modules n'est pas le même que celui utilisé dans les contrats de licence/maintenance/fourniture.

Remarque : Il arrive fréquemment que le nommage des sources diffère du nommage « marketing » des modules, pour des raisons d'historique, parce qu'à un module commercial correspondent plusieurs modules techniques, etc.

- Toute information complémentaire éventuellement nécessaire pour préciser le périmètre de cette livraison.

Remarque : Dans la mesure où ce logiciel/application est l'objet principal du contrat de licence, d'utilisation et/ou de maintenance, il mérite d'être soigneusement décrit.

LISTE DES COMPOSANTS EXTERNES UTILISÉS

Au niveau du processus de Build

Remarque : la quasi-totalité des applications livrées à des utilisateurs finals comporte un minimum de composants non propriété du prestataire : *Open Source*, composants d'infrastructure nécessaire au déploiement ou à l'exécution (bibliothèques « run-time"...). Il convient de les lister précisément, tout particulièrement quand le contrat de fourniture prévoit transfert total ou partiel de propriété au client.

Liste des composants utilisés par le processus de *build* en précisant :

- Nom et niveau de version du composant, nom du propriétaire pour les logiciels propriétaire (le propriétaire pouvant être le prestataire lui-même qui réutilise des composants d'autres applications et dont il veut absolument conserver la pleine propriété),
- Type de logiciels : *Open Source*, propriétaire
- Statut des licences : licence propriétaire, GPL, LGPL, etc.
- Remarque : « licence gratuite » n'est pas un statut au sens juridique...
- État technique de ces composants : état natif ou modifié, corrigé, complété par le prestataire...

Au niveau du processus d'intégration/packaging

Liste des composants utilisés par le prestataire dans le processus d'intégration en précisant :

- Nom et niveau de version du composant, nom du propriétaire pour les logiciels propriétaire (le propriétaire pouvant être le prestataire lui-même qui réutilise des composants d'autres applications et dont il veut absolument conserver la pleine propriété)
- Type de composant : documentation, données insérées en base (par exemple données cartographiques, liste des codes postaux, images, etc.), logiciels...
- Statut des licences : licence propriétaire, GPL, LGPL...
- Support de livraison : fourniture sur CD, téléchargement avec nom du site...
- État technique de ces composants : état natif ou modifié, corrigé, complété par le prestataire...

Au niveau du processus de déploiement

Liste des pré requis de déploiement pour tous matériels et composants non fournis par le prestataire mentionnant :

- Environnement matériel y compris des dispositifs spéciaux tels que lecteurs de cartes, configuration de disques et plus généralement tout ce qu'il incombera au client de préparer pour la phase de déploiement (que celle-ci soit faite par lui-même ou par le prestataire)
- Nom de chaque composant logiciel : système d'exploitation, moteur de bases de données, dispositifs de connexion à distance pour la télémaintenance...
- Niveau de version de chaque composant logiciel
- Le cas échéant, procédure spécifique d'installation et de paramétrage pour être compatible avec la livraison...

Remarque : Cette partie pourra faire référence à la documentation d'installation ou en reprendre les éléments concernés. Rappelons que cette annexe est un document contractuel qui fera foi en cas de contestation, de litige, d'arbitrage.

Remarque : Cette information est nécessaire quand l'application est installée chez le client utilisateur. Mais elle peut être également « rassurante » quand l'application est fournie en mode SAAS/ASP et déployée chez un hébergeur.

Remarque : environnement matériel doit être précisé. D'autant plus important qu'il y a de plus en plus de logiciels. En terme d'inventaire, il faut avoir les mentions des micrologiciels (tBIOS) : indissociable du matériel (ne peuvent fonctionner sans). Ces informations doivent aussi être conservées.

DESCRIPTION DU CONTENU DE LA LIVRAISON À UN UTILISATEUR/CLIENT

Il s'agit ici de donner l'arborescence du contenu du support de livraison en listant les principaux répertoires, sous répertoires et fichiers, avec une description sommaire de leur contenu, de telle sorte que le client puisse raisonnablement rapidement identifier :

- Les composants externes qui lui sont fournis avec la livraison et nécessaires pour le déploiement et/ou le fonctionnement de l'application (ceux-ci ayant été décrits ci-dessus),
- Les modules livrables correspondant aux modules listés dans son contrat de licence,
- Tout ce qui concerne le déploiement, le peuplement et le paramétrage initial des bases de données,
- Des jeux de tests à dérouler après déploiement pour s'assurer que la procédure d'installation s'est déroulée normalement (que ces jeux de test soient déroulés par le prestataire ou par le client lui-même),
- Les documentations techniques dont il aura besoin : pour le déploiement, pour l'exploitation, la maintenance, le support, etc. (par exemple : manuel utilisateur, manuel d'exploitation, ...)

Remarque : Il ne s'agit pas d'une liste exhaustive des répertoires et fichiers, mais d'une description de la livraison permettant au client d'y retrouver rapidement l'essentiel des informations et éléments dont il aura besoin, soit pour déployer et valider, soit pour prononcer sa recette.

Remarque : Il arrive parfois que le client ne respecte pas scrupuleusement les procédures de déploiement et de test, parce qu'il ne prend pas la peine de lire les documentations fournies, parce qu'il fait des économies sur la formation de son personnel etc. Il peut être intéressant pour le prestataire d'apporter alors la preuve de la qualité, complétude et validité des documentations, procédures, informations fournies au client lors de la livraison.

DESCRIPTION DU CONTENU DE LA LIVRAISON DESTINÉE À UN SÉQUESTRE

Un certain nombre de contrats de fournitures logicielles prévoient une clause dite « de séquestre/défaillance » par laquelle le fournisseur s'engage à déposer sous séquestre les éléments nécessaires à la reprise de maîtrise en maintenance de l'application livrée par toute personne compétente, en cas de défaillance en maintenance du fournisseur initial.

Cela suppose que soit ajoutés à la liste fournie ci-dessus, un certain nombre de composants et documentations non fournis normalement à l'utilisateur, mais nécessaires à cette reprise de maîtrise en maintenance dans des conditions raisonnables.

La liste de ces éléments complémentaires peut varier en fonction des objectifs des parties, des types de déploiement ou de commercialisation. On pourra donc trouver les cas suivants (non exhaustifs) :

- Commercialisation en mode SaaS/ASP,
- Solutions financières avec un *leaser*.
- Fourniture à un intégrateur,
- Système de preuve vis-à-vis d'une autorité de tutelle (sanitaire, sécurité, fiscale, traçabilité, Objectif patrimonial,
- Système de preuve en matière de droits d'auteur.

Commercialisation en mode SaaS/ASP :

Ce cas est également assimilable aux solutions *outsourcing*. En pareil cas, l'utilisateur est dépendant de son fournisseur, non seulement en ce qui concerne la maintenance évolutive, mais également toute la production (exploitation, sauvegardes, sécurité...). Dans la plupart des cas, le fournisseur utilise les services d'un « hébergeur ». En cas de besoin, l'utilisateur aura besoin d'informations et de composants complémentaires tels que :

- Nom et coordonnées de l'hébergeur,
- Descriptif détaillé des composants fournis par l'hébergeur : type de machine (dédiée ou partagée), système d'exploitation (nom, niveau de version, procédure d'installation et de paramétrage), mots de passe d'accès au serveur, nature des composants spécifiques matériels et logiciels complémentaires fournis par l'hébergeur, dispositifs de sécurité, etc.
- Procédure détaillée de déploiement et de paramétrage de l'application client sur l'environnement d'hébergement, mots de passe d'accès, de redirection de noms de domaines, etc.
- Procédures complémentaires d'exploitation utilisées par le fournisseur (gestion de profils utilisateurs, purge des bases de données, procédures de mises à jour de l'application, etc.)
- Systèmes de sauvegarde (*mirroring*), localisation des *back-up*...
- Plus généralement, toute information permettant au client comme au fournisseur de basculer la production sur un autre site en cas de défaillance de l'hébergeur, par exemple.

Solutions financières avec un Leaser (Société de leasing, de crédit-bail) :

Ce cas est assez proche de celui du déploiement en mode SaaS, à ceci près qu'une société financière s'intercale entre le fournisseur et le client final et que c'est elle qui perçoit les mensualités couvrant la licence, la maintenance applicative, l'hébergement et même souvent le matériel installé en local. Cette société financière peut alors se trouver engagée financièrement sur une application et a besoin de sécuriser ses encaissements (autant par des procédures de secours et de sécurité que par des contrats d'assurance). Le « financier » a alors besoin d'être assuré de disposer de tous les moyens pour relancer la production en priorité puis la maintenance applicative dans des délais compatibles avec ceux couverts par son contrat d'assurance par exemple ou dans les délais prévus avec les clients et au-delà desquels des pénalités de retard pourraient lui être imposées. Il peut alors avoir besoin d'informations complémentaires telles que :

- Liste complète des mots de passe de redirection des noms de domaine des clients concernés,
- Liste des mots de passe d'accès aux machines locales des clients pour intervention en maintenance et mise à jour des applications déployées en local,
- Copie des factures de vente et/ou bons de garantie des matériels financés ...

Fourniture à un intégrateur

L'intégrateur a a priori vocation à reprendre la maintenance et le développement en cas de défaillance du fournisseur d'origine. C'est souvent pour cette raison qu'un « grand compte » préfère travailler avec un intégrateur, même s'il lui impose d'intégrer telle ou telle solution qu'il a validée comme correspondant à ses besoins. En pareil cas, les contraintes de reprise de la maintenance peuvent imposer des délais plus courts et des pénalités de retard.

L'intégrateur doit alors disposer de toutes les informations techniques lui permettant de reprendre la maîtrise de l'application dans les délais impartis. Il devra alors disposer des informations telles que :

- Normes internes de développement et de nommage,
- Spécifications fonctionnelles et techniques,

- Dossiers d'architecture,
- Scripts de création peuplement et paramétrage initial des bases de données, descriptions des tables, index, etc.
- Quand il y a des composants hardware tels que des cartes spécifiques, nom des sous-traitants, dossiers de conception et environnements de développement correspondants (y compris ceux utilisés par les sous-traitants et dont les licences peuvent être très coûteuses), sources, documentations de conception, fichiers résultants, etc.

Système de preuve vis-à-vis d'une autorité de tutelle

A la demande d'une autorité de tutelle, il peut être demandé de produire des sources, des plans de test et dossiers de validation, que ce soit pour des raisons fiscales, de santé publique, de risques humains (logiciels embarqués sur des matériels roulants ou volants) etc. Le fournisseur aussi bien que son intégrateur peuvent être amenés à fournir solidai-
rement des informations complémentaires telles que ;

- Dossiers et plans de test,
- Dossier de validation pour mise en conformité avec les exigences de la CNIL, de la FDA, de normes aéronautiques, etc.
- Systèmes de traçabilité,
- Systèmes de sécurisation des données et des transmissions, etc.

Dans le but de prouver qu'ils ont respecté les normes imposées et/ou qu'ils ont testé leurs applications de façon professionnelle et à la hauteur des risques potentiellement encourus pas leurs utilisateurs

Système de preuve en matière de droits d'auteur

En matière de « Droits d'auteur » et pour autant que cela soit nécessaire en matière de « logiciels libres », l'administration de la preuve est dite « de libre parcours ». En d'autres termes, tout ce qui contribue à renforcer le « faisceau de preuves de propriété » est utile. Dans ce cadre, le propriétaire de l'application (notamment quand il s'agit de code source interprété et donc fourni à l'utilisateur) pourra ajouter un certain nombre d'informations complémentaires telles :

- Justification des choix d'environnements de développement et de déploiement, articulation de ces composants, etc. En effet ; si le fournisseur n'est pas propriétaire des environnements qu'il a utilisé, le choix qu'il en a fait n'est pas dû au hasard et sa cohérence d'avec le contenu de son patrimoine en renforcera nécessairement le système de preuve.
- Bases de données de sources (CVS, VSS, Perforce, etc.) donnant l'historique détaillé de tout ce qui a été développé, modifié, ajouté au fil des mois et des années à une application initiale ou entre deux versions de cette application,
- Documents préparatoires au développement, etc.

ANNEXE 4 : MODÈLE DE DOCUMENT DE DIFFUSION LINAGORA : LICENCE, DIFFUSION ET CONTRIBUTEURS

LICENCE

Ce document est licencié cumulativement sous licences GNU FDL 1.3 et CC-BY-SA 3.0.

La **GNU FDL** est une licence libre *copyleft* calquée sur la GNU GPL, parfaitement adaptée aux documentations et qui nécessite que soit annexé systématiquement le texte de la licence.

La **CC-BY-SA** est une licence libre *copyleft* parfaitement adaptée aux contenus multimédias. Sa grande modularité permet de mixer les réalisations.

Cette double licence permet un usage du document qui soit conforme à l'une ou l'autre des licences. Plusieurs avantages peuvent être avancés :

1. Le contenu sous licence est dès lors compatible avec la totalité des licences qui lui sont adjointes ;
2. L'étendue de la double licence est limitée à celle de la licence la plus permissive ;
3. L'utilisation d'au moins une licence française sécurise la double licence au regard des dispositions françaises.

EXCEPTIONS

Par dérogation au paragraphe précédent, certaines exceptions peuvent être apportées à la cession de droits telle que consentie par la licence. Les éléments concernés par ces limitations sont les suivants :

Élément	Titre et/ou description	Licence	Remarques

DIFFUSION DU DOCUMENT

Par dérogation aux paragraphes précédents, la diffusion du document est limitée de la manière qui suit :

Mention de diffusion : **Restreinte**

Nom	Organisme	Pour	Média
Tous les collaborateurs	Groupe Linagora	Information	GED, Wiki

LISTE DES CONTRIBUTEURS

Prénom NOM, Prénom NOM, Prénom NOM.

ANNEXE 5 : RÉFÉRENT *OPEN SOURCE*

Exemple de référent : le « Responsable *Open Source* » d'Orange Labs (transverse à une organisation)

Intitule du poste : Responsable *Open Source*

Mission :

L'utilisation grandissante de logiciels issus de _____ dans les nouveaux produits et services a un impact positif sur les coûts des développements, mais augmente les risques sur la propriété intellectuelle, à la fois en terme juridique, en terme de capacité de protection des services France Télécom et de potentiel de valorisation. Dans l'objectif de favoriser l'utilisation de l'*Open Source* tout en maîtrisant les risques associés et minimisant les impacts sur la propriété intellectuelle, la division R&D a défini sa politique de l'*Open Source*.

La mission du responsable *Open Source* est de piloter la mise en œuvre de cette politique dans la production et l'utilisation des logiciels à la Division R&D. Cette mission s'exerce en coopération avec la Direction de la Propriété Industrielle et Valorisation et la Direction Juridique.

Activités principales :

Contribuer à décliner la politique *Open Source* dans le processus de production de la Division R&D. La politique *Open Source* doit être intégrée en amont du processus et suivie tout au long du cycle de vie du logiciel.

Diffuser la connaissance des pratiques et des règles dans l'utilisation et la production de logiciels dans le contexte de l'*Open Source*, faire connaître la politique *Open Source* de la Division et la politique de propriété intellectuelle et leur déclinaison dans le processus.

Apporter, en amont, un soutien aux projets dans les décisions liées à l'*Open Source* pour :

Évaluer les avantages et les risques liés à l'utilisation ou la production de logiciels *Open Source* en termes de coût, délais, différenciation, maîtrise, solidité technique, maturité, pérennité, popularité de techniques ou d'usages, image de France Télécom, etc.

Éliminer tous les risques ayant un impact négatif sur la potentialité de revenu de valorisation de brevets ou sur la capacité de protection des services de France Télécom, sur leurs distributions et les obligations attachées aux licences.

Instruire, en coopération avec l'ensemble des acteurs impliqués à R&D et dans le Groupe, les dossiers *Open Source* en prenant en compte les contextes de développement dans les projets (prototypes, produits), d'utilisation et de distribution dans le Groupe (y compris les filiales françaises ou étrangères), de protection des services de France Télécom et de valorisation externe et les obligations juridiques liées aux licences *Open Source* utilisées. Plus particulièrement, en cas de risque conséquent sur la valorisation de la propriété intellectuelle ou la capacité de protection de services, lancer une étude avec les Ingénieurs Brevet et solliciter le cas échéant un comité d'arbitrage. La politique *Open Source* s'appliquant également aux développements réalisés en externe, une attention particulière sera portée dans les dossiers intégrant des sous-traitances de logiciels, des relations avec des partenaires industriels, des achats « sur étagère » de composants.

Contribuer à la recommandation des composants référencés dans le catalogue des logiciels France Télécom et à la décision de publication dans la communauté *Open Source*.

Identifier des cas génériques et toutes les améliorations permettant de simplifier le processus d'instruction des dossiers *Open Source* en matière de propriété intellectuelle.

Compétences techniques et relationnelles :

Compétence approfondie de l'ingénierie logicielle, expérience significative dans le développement de logiciel dans un contexte industriel,

Bonne connaissance générale des domaines informatique et télécom,

Expérience opérationnelle dans le monde de l'*Open Source* (utilisation de composants, contribution voire pilotage d'une communauté *Open Source*), expérience de travail en réseau,

Capacité à appréhender les problèmes juridiques et de propriété industrielle,

Capacité d'analyse et de synthèse, capacité à la négociation, à convaincre, à communiquer,

Anglais indispensable

ANNEXE 6 : TABLEAU DE CLASSIFICATION ET DE CROISEMENT DES LICENCES

Attention : ressource distribuée à titre purement indicatif

Partant du principe que toutes les licences *Open Source* se rejoignent sur le plus grand nombre de leurs clauses, ce tableau vise à les distinguer par les obligations à la charge du licencié qui diffèrent.

Par ailleurs, les licences *Open Source* consacrent l'existence d'une sphère privée dans laquelle les licenciés bénéficient de tous les droits ? sans être contraints de respecter d'obligation .

Les *Creative Commons* sont déconseillées pour un usage sur des logiciels, notamment parce qu'elles sont modifiées pour chaque système juridique (on perd cette notion d'internationalisation) et la dernière version (3.0) n'est justement pas encore d'actualité en France.

Tableau 1 : redistribution sous une autre licence (compatibilité)

Lecture du tableau : Peut-on, à partir d'une licence A (licence d'origine), distribuer sous une autre licence B (licence de distribution)

		Licences B : utilisée pour la distribution																		
		Copyleft												Permissif						
		Propriétaire	Affero GPL	GPL v3	GPL v2	LGPL v3	LGPL v2.1	CeCILL	CeCILL-C	MPL	OSL	EUPL	CPL	EPL	BSD	BSD non Modifiée	Apache	Latex	Academic Free Licence	CeCILL-B
Licences A : d'origine	Propriétaire	*	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
	Affero GPL	✗	●	*	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
	GPL v3	✗	*	●	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
	GPL v2	✗	V	V	●	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
	LGPL v3	✗	*	●	✗	●	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
	LGPL v2.1	✗	V	V	●	V	●	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
	CeCILL	✗	✗	●	●	✗	✗	●	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
	CeCILL-C	✗	✗	?	?	✗	✗	●	●	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
	MPL	✗	✗	✗	✗	✗	✗	✗	✗	●	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
	OSL	✗	✗	✗	✗	✗	✗	✗	✗	✗	●	✗	✗	✗	✗	✗	✗	✗	✗	✗
	EUPL	✗	✗	✗	●	✗	✗	●	✗	✗	●	●	●	✗	✗	✗	✗	✗	✗	✗
	CPL	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	●	✗	✗	✗	✗	✗	✗	✗
	EPL	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	●	✗	✗	✗	✗	✗	✗
Permissif	BSD	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
	BSD non Modifiée	*	?	?	✗	?	✗	?	?	✗	✗	✗	✗	✗	*	●	*	*	*	●
	Apache	●	*	*	✗	*	✗	✗	✗	✗	✗	✗	✗	✗	*	*	●	*	*	*
	Artistic Licence	●	*	*	✗	*	✗	✗	✗	✗	✗	✗	✗	✗	*	*	*	●	*	*
	Academic Free Licence	●	✗	*	✗	✗	✗	✗	✗	✗	●	✗	✗	✗	*	*	*	*	●	*
	CeCILL-B	●	✗	*	✗	✗	✗	●	●	✗	✗	✗	✗	✗	*	*	*	*	*	●

"source : Benjamin Jean, Compatibility / Incompatibility, European OpenSource & Free Software Law Event, 2009 -
licence : LAL 1.3, CC-BY-SA 3.0 and GFDL 1.3"

Tableau 2 : Redistribution sous une autre licence (Documentation)

Note : ce n'est plus seulement le droit d'auteur qui est encadré par l'usage des licences *Open Source*, mais tous les droits de propriété intellectuelle.

		Licences B : utilisée pour la distribution														
		Propriétaire	Copyleft									Permissif				
			GNU FDL	GNU GPL	CC By-SA	CC By-NC-SA	LAL	DSL	OPL	OdbL	OSL	BSD	CC-0	CC-BY	CC-BY-NC	CC-BY-ND
Licences A : d'origine	Propriétaire															
	GNU FDL															
	GNU GPL															
	CC-BY-SA															
	CC By-NC-SA															
	LAL															
	DSL															
	OPL															
	OdbL															
	OSL															
	BSD															
	CC-0															
	CC-BY															
	CC By-NC															
	CC By-ND															



Une incertitude existe



il est nécessaire d'ajouter une clause/d'adapter la licence



Non



Oui

"source : Benjamin Jean, Compatibility / Incompatibility, European OpenSource & Free Software Law Event, 2009 --
licence : LAL 1.3, CC-BY-SA 3.0 and GFDL 1.3"

ANNEXE 7 : LISTE DE RESSOURCES UTILES RELATIVES AUX LICENCES *OPEN SOURCE*

- Décision américaine Jacobsen v. Katzer

Elle concerne la licence *Open Source* Artistic (principalement utilisée pour Perl) et sanctionne une violation de celle-ci : Est une contrefaçon l'irrespect des conditions d'une licence *Open Source*.

<http://www.ca9c.uscourts.gov/opinions/08-1001.pdf>

L'analyse suivante est particulièrement conseillée : <http://lawandlifesiliconvalley.blogspot.com/2007/08/new-open-source-legal-decision-jacobsen.html>

- Le guide publié par la SFLC (branche détachée de la FSF, sous la direction d'Eben Moglen, qui s'occupe des questions légales et qui organise la rédaction des licences GNU) sur la GPL.

- <http://www.softwarefreedom.org/resources/2008/compliance-guide.html>

- Foss Bazaar : Initiative qui porte justement sur le bon usage du Libre pour les entreprises.
<https://fossbazaar.org/>

- Blog VVL : A propos des licences *Open Source* et des clauses contractuelles des contrats annexes.

- <http://blog.venividilibri.org/> et le site : <http://wiki.venividilibri.org/>

- European Opensource Lawyers Event (EOLE) 2008 : organisation annuelle d'un séminaire dédié aux logiciels libres, *Open Source*, et aux questions juridiques afférentes. La première édition eut lieu le 24 septembre 2008 à Paris.
<http://www.eolevent.eu/> et http://eolevent.eu/?q=Slides_Speakers

ANNEXE 8 : DÉFINITION DE L'OSI

COMMENTAIRES/REMARQUES

_ OSI (Open Source Initiative) : Cette définition centrée sur l'objet de droits, le logiciel, a ensuite été développée dans l'*Open Source Definition*¹⁴⁴ qui définit les critères qu'une licence doit vérifier afin d'assurer à chacun et sans discrimination ce partage du monopole d'exploitation. Est donc *Open Source* une licence qui vérifie :

1. La libre redistribution du logiciel — elle ne peut, par exemple, exiger le paiement d'une redevance supplémentaire ;
2. Le code source doit être fourni ou être accessible ;
3. Les dérivés des œuvres doivent être permis ;
4. L'intégrité du code doit être préservée — un tiers ne peut pas s'approprier le travail d'un autre et les contributions de chacun sont clairement attribuées (les modifications peuvent n'être éventuellement distribuées que sous forme de patch, séparément : *distinguo* que ne tolère pas la FSF) ;
5. Pas de discrimination entre les personnes ou les groupes — toute personne détentrice d'une copie du logiciel bénéficie des termes de la licence tant qu'il s'y conforme lui-même ;
6. Pas de discrimination entre les domaines d'application — la licence se limite à la propriété intellectuelle : elle ne peut en aucun cas réguler un autre domaine « politique » ;
7. La licence s'applique sans dépendre d'autres contrats — ex : on ne peut pas ajouter un NDA (accord de confidentialité) lors de la cession du logiciel ;
8. La licence ne doit pas être propre à un produit — elle est attachée au code et non à un logiciel particulier : un composant peut resservir dans un logiciel différent, voire concurrent ;
9. La licence d'un logiciel ne doit pas s'étendre à un autre — Remarque : la large étendue de la GPL (et c'est la raison pour laquelle certains utilisent le terme de viralité ou de propagation) est conforme à ce critère puisqu'elle ne s'étend qu'au programme envisagé comme un tout ;
10. La licence doit être neutre technologiquement — c'est-à-dire ne pas dépendre d'une technologie.

¹⁴⁴ Voir : <http://www.opensource.org/docs/definition.php>

ANNEXE 9 : LICENCE ET PROCHAINE VERSION DU DOCUMENT

9.1 LICENCES

« GUIDE *OPEN SOURCE* : RÉFLEXIONS SUR LA CONSTRUCTION ET LE PILOTAGE D'UN PROJET *OPEN SOURCE* », réalisé sous la direction de Benjamin Jean et Olivia Flipo — Copyright © 2010

Ce document est licencié cumulativement sous licences **GNU FDL 1.3** et **CC-BY-SA 3.0**.

La **GNU FDL** est une licence libre *copyleft* calquée sur la GNU GPL, parfaitement adaptée aux documentations et qui nécessite que soit annexé systématiquement le texte de la licence.

La **CC-BY-SA** est une licence libre *copyleft* parfaitement adaptée aux contenus multimédias. Sa grande modularité permet de mixer les réalisations.

Cette double licence permet un usage du document qui soit conforme à l'une ou l'autre des licences. Plusieurs avantages peuvent être avancés :

1. Le contenu sous licence est dès lors compatible avec la totalité des licences qui lui sont adjointes ;
2. L'étendue de la double licence est limitée à celle de la licence la plus permissive ;
3. L'utilisation d'au moins une licence française sécurise la double licence au regard des dispositions françaises.

9.2 PROCHAINE VERSION DU DOCUMENT

Première brique d'un guide *Open Source* dont nous sommes certains de l'utilité, ce document sera amené à évoluer. Chaque nouvelle version fera l'objet d'une numérotation distincte.

N'hésitez pas à nous contacter si vous êtes intéressés et que vous souhaitez rejoindre le groupe de travail.

9.3 LICENCE GNU FDL

GNU Free Documentation License
Version 1.3, 3 November 2008

Copyright (C) 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <<http://fsf.org/>>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright

holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the

conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant

Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:
Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

REMERCIEMENTS

Nous tenons à remercier chaleureusement les membres du groupe de travail et toutes les personnes ayant contribué à la réalisation de ce guide, tout particulièrement les pilotes de ce guide et les experts interrogés.

Les pilotes :

Olivia FLIPO – CABINET STAUB & ASSOCIES
Benjamin JEAN - LINAGORA

Les contributeurs :

Sylvie BENARD
Thomas BEAUGRAND – CABINET STAUB & ASSOCIES
François BESSAGUET – STERIA
Glenn BRAULT - NOVELL
Francois BERTIN - BULL
Charles CACHERA – REDHAT
Olga CHARMOILLE - CSC
Bruno CORNEC - HP
Frédéric GERAUD DE LESCAZES - MICROSOFT
Marisela GRATEROL - LOGICA
Vincent HANNIET - MIA-SOFTWARE Groupe SODIFRANCE
Pascale HATE - UPERTO / Groupe DEVOTEAM
Henri DE HAUTECLOCQUE – LOGITAS
Jean-Pierre LAISNE - BULL
Christophe LAPLUME - MICROSOFT
Valérie PAROT ALIBERT - SAGE
Coralie PRADOLIN - BULL
Eric SOARES - INGRES
Olivier THIRARD - ORANGE FT GROUP
Sophie THEPAULT - DEVOTEAM

Les membres du comité de rédaction et de relecture :

Thomas BEAUGRAND - CABINET STAUB & ASSOCIES
Francois BERTIN - BULL
Francois BESSAGUET – STERIA
Bruno CORNEC - HP
Olivia FLIPO – CABINET STAUB & ASSOCIES
Frédéric GERAUD DE LESCAZES - MICROSOFT
Marisela GRATEROL - LOGICA
Vincent HANNIET - MIA-SOFTWARE Groupe SODIFRANCE
Benjamin JEAN – LINAGORA
Henri LECLERC DE HAUTECLOCQUE - LOGITAS

Les « grands témoins » :

Alexandre ZAPOLSKY, PDG de LINAGORA
Hervé GUYOMARD, Black Duck France
Loïc DACHARY, FSF France
Luc GRATEAU, Responsable du service de propriété intellectuelle et valorisation, INRIA
Philippe CARRE, Program Manager, Alcatel-Lucent
Bruno CORNEC, FOSSology/FOSSBazaar

Remerciements du Groupe de travail aux sociétés et personnes qui ont manifesté un intérêt aux travaux.

L'Open source constitue une **forme technico-juridique originale** dans l'environnement du droit d'auteur et du copyright anglo-saxon, au sein duquel elle s'est développée. Issue d'une démarche à la fois philosophique et militante, cette construction s'est propagée dans le monde entier - non sans rencontrer un certain nombre de résistances et d'interrogations - mais avec un succès et des avantages aujourd'hui incontestables.

Si les développeurs et techniciens ont immédiatement saisi les avantages de l'**Open Source**, il en va différemment du marché et des juristes. Ceux-ci comprennent progressivement que le cœur du patrimoine des éditeurs logiciels (le code source) et leur avantage concurrentiel par excellence ne doit plus être systématiquement « *réservé* » mais au contraire, « *propagé* ».

C'est dans ce contexte que Syntec informatique et la FniLL (*Fédération Nationale de l'Industrie du Logiciel Libre*) ont décidé de mettre en place un groupe de réflexion commun afin de permettre à leurs membres (chefs d'entreprise, développeurs, juristes, consultants...) de partager et de mutualiser leurs expériences et compétences.

Le **Guide** qui a résulté de leurs travaux est à la fois une analyse détaillée de ses principes et de ses effets, et un *vade-mecum* qui pourra servir aussi bien aux services juridiques qu'aux services techniques et marketing de l'entreprise. Il répond à toutes les questions qui se posent, tant à l'éditeur logiciel qui souhaite se lancer dans l'aventure de l'*Open source*, qu'à l'entreprise cliente qui souhaite recourir à cette forme de solutions informatiques.

L'étude menée par le groupe de travail sous les auspices de Syntec Informatique et de la FniLL a également permis de dégager un principe essentiel : l'adoption des logiciels *Open Source* implique une **traçabilité** parfaite des composants logiciels employés, d'une part, et une **collaboration totale entre les services techniques, juridiques et commerciaux** de l'entreprise, d'autre part. Le **Guide** s'adresse donc successivement aux **techniciens**, aux **juristes**, aux **commerciaux**, aux **ressources humaines** et à la **direction générale**, qui est la mieux placée pour décider des principes et les mettre en application.

Le **Guide** présente à ces différents lecteurs la **typologie des licences libres** existantes, des plus permissives aux plus « virales » (notamment la fameuse licence GNU GPL), et expose dans le détail la philosophie et les conséquences pratiques de chacune des grandes caractéristiques des licences libres, afin de permettre à l'entreprise de choisir, en connaissance de cause, celle(s) des licences libres qui conviennent le mieux à ses besoins ou à sa stratégie commerciale.

Le **Guide** fournit ainsi les **principes d'analyse de compatibilité entre licences** et des recommandations et alternatives dans ce travail complexe. Il propose également les **précautions rédactionnelles** à prévoir dans les contrats informatiques afin de prendre en compte les spécificités du Libre.

La notion d'**inventaire**, de panorama exhaustif et détaillé des logiciels employés par l'entreprise ou intégrés dans ses produits, devient capitale. Cet inventaire permet d'une part de contrôler les prérogatives permises par chaque licence libre en jeu, et d'autre part de définir les prérogatives qui seront ensuite transmises aux utilisateurs avec le code source. Un tel inventaire, s'il est établi par les développeurs qui peuvent choisir les composants qu'ils intègrent au produit, doit également être partagé avec les juristes, qui vont définir les licences applicables, contrôler leur compatibilité, et vérifier leur adéquation avec la stratégie commerciale de l'entreprise.

La **stratégie commerciale** de l'entreprise, elle aussi, se modifie sous l'impact des licences libres. Dès lors que l'éditeur diffuse le code source gratuitement, il ne peut plus compter sur les redevances de licences pour rentabiliser ses travaux de Recherche & Développement effectués en amont. L'éditeur qui fait le choix du libre doit ainsi affiner son usage des droits de propriété intellectuelle (« *quelle licence libre proposer ?* ») et prendre l'avantage sur d'autres plans de compétition (« *quel business model adopter ?* »).

Plusieurs modèles se sont donc développés, fondés sur un « *dual licensing* » mêlant le régime propriétaire et le régime libre, ou encore sur un « *bouquet de services* » à **valeur ajoutée** que l'entreprise commercialise une fois que son logiciel librement diffusé a su lui attirer les faveurs du marché. D'où la spécialisation progressive des sociétés du secteur en faveur de modèles hybrides d'édition de logiciel, de distribution et d'intégration spécifique très orientés sur une économie de services.

Enfin, la distribution ou le recours aux logiciels libres implique également de sensibiliser la **clientèle** à ses avantages. Les commerciaux de l'entreprise, qu'ils procèdent à la diffusion du logiciel de l'éditeur ou qu'ils évoluent au contraire au service « achats » d'une entreprise utilisatrice, doivent savoir renoncer à certains « *habitudes* » pour adopter de nouveaux repères : remplacer la **maintenance de l'éditeur** par **l'expertise de la communauté**, remplacer la **garantie d'éviction** par la **transparence du code source**, remplacer la **licence personnelle et payante** par un **processus ouvert de diffusion à valeur ajoutée**, permet en réalité de substituer la **dépendance** vis-à-vis de l'éditeur par la **confiance** dans sa capacité à fournir un logiciel toujours adapté aux besoins quelle que soit leur évolution.

Le Guide propose ainsi un rappel des grands principes juridiques applicables au logiciel, et présente leur application originale dans le monde du Libre, ainsi que les **précautions techniques et juridiques** à respecter, puis les différentes **organisations commerciales et industrielles** qui peuvent être adoptées afin de garantir une bonne maîtrise du monde Open Source .

Il appartient donc au secteur informatique de parfaitement saisir les enjeux, les contraintes et les avantages des Logiciels Libres, qui permettent de passer de modèles économiques compartimentés et parfois inadaptés aux nouvelles technologies, vers de nouveaux modèles plus ouverts, plus collaboratifs.