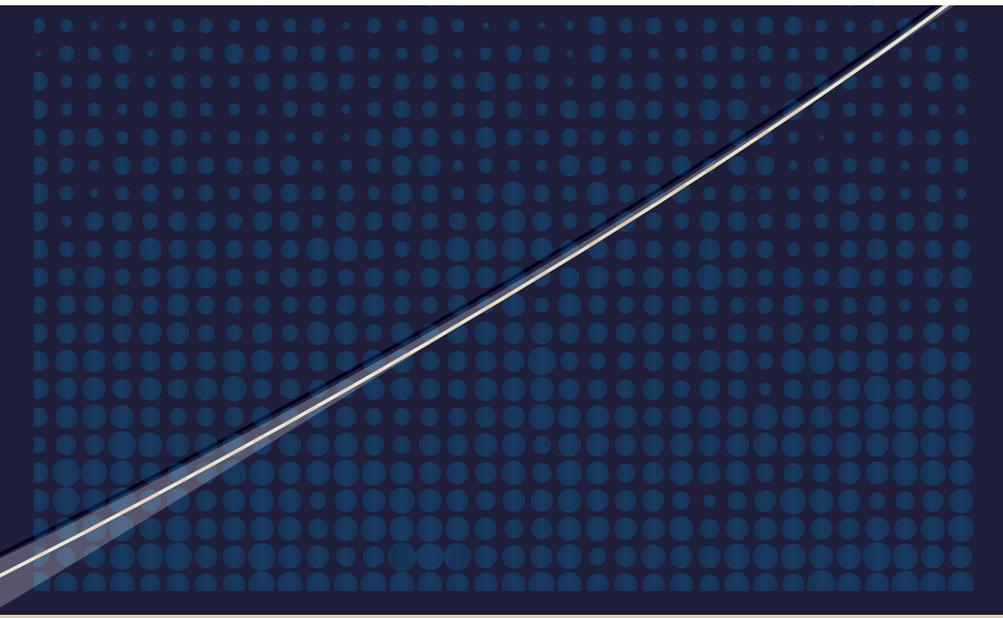


LOGICIELS  
LIBRES  
ET OUVERTS  
**Guide de référence**



Cette publication a été réalisée par  
le Sous-secrétariat aux ressources informationnelles et  
bureau du dirigeant principal de l'information  
et produite par la Direction des communications.

Vous pouvez obtenir de l'information au sujet  
du Conseil du trésor et de son Secrétariat  
en vous adressant à la  
Direction des communications  
Secrétariat du Conseil du trésor  
5<sup>e</sup> étage, secteur 500  
875, Grande Allée Est  
Québec (Québec) G1R 5R8

Téléphone : 418 643-1529  
Sans frais : 1 866 552-5158  
Courriel : [communication@sct.gouv.qc.ca](mailto:communication@sct.gouv.qc.ca)  
[www.tresor.gouv.qc.ca](http://www.tresor.gouv.qc.ca)

Dépôt légal - 2013  
Bibliothèque et Archives nationales du Québec  
Bibliothèque et Archives du Canada

ISBN 978-2-550-67217-3 (en ligne)  
Tous droits réservés pour tous les pays.  
© Gouvernement du Québec - 2013

## Sommaire exécutif

En décembre 2010, le gouvernement du Québec a rendu publique la Politique-cadre sur la gouvernance et la gestion des ressources informationnelles des organismes publics, dont l'un des objectifs est de doter les organismes publics d'outils leur permettant de tirer profit des logiciels libres. Le présent guide de référence constitue un élément important de cette stratégie, puisqu'il fournit des réponses à plusieurs des questions de nature juridique et opérationnelle auxquelles est confrontée l'administration publique en ce qui a trait à l'utilisation des logiciels libres et ouverts.

Ce guide tente d'abord d'expliquer en termes clairs et concis ce qui distingue les logiciels libres et ouverts des autres types de logiciels. Dans ce but, on y aborde en premier lieu les notions de base du droit d'auteur et les différents mécanismes permettant le transfert de ces droits. Par la suite, une présentation des modèles d'affaires en résultant permet de cadrer ces mécanismes de gestion de droit avec la réalité de l'industrie du logiciel.

Les logiciels libres et ouverts ayant suscité toute une série d'interrogations sur le plan légal, les principaux enjeux juridiques y sont aussi énumérés et discutés. Bien que quelques incertitudes subsistent toujours quant à la validité de certaines licences libres et ouvertes, quant à la capacité d'établir avec certitude la chaîne de titres du code qui leur est soumis ou quant aux conséquences de leurs clauses, ce guide démontre que le risque juridique théorique doit être relativisé en considération de sa mise en application pratique. Dans les faits, les tribunaux ont jusqu'ici accepté de reconnaître l'environnement juridique mis en place par les licences libres et ouvertes.

Finalement, des conseils pratiques visent à encadrer le recours aux logiciels libres et ouverts par l'Administration. Le déroulement du processus d'acquisition est revu sous cet angle, allant de l'opportunité de sélectionner un logiciel libre et ouvert jusqu'aux particularités du contrat de services qui en résulte. Différents scénarios sont envisagés, qu'il s'agisse de la simple utilisation d'un logiciel libre et ouvert, de son adaptation par le personnel de l'administration publique ou de son éventuelle redistribution sur Internet.

# Table des matières

Historique.....	5
1. Introduction.....	6
1.1. Qu'est-ce qu'un logiciel libre et ouvert? .....	6
1.2. Droit d'auteur et licences logicielles .....	9
1.3. Les modèles d'affaires du libre .....	17
2. Principaux enjeux juridiques .....	20
2.1. Validité juridique.....	20
2.2. Établissement de la chaîne de titres .....	25
2.3. Brevets logiciels.....	27
2.4. Qualification du contrat de licence .....	29
2.5. Obligations des parties.....	31
2.6. Clauses d'exclusion de responsabilité .....	34
3. Principaux enjeux opérationnels .....	36
3.1. Coût total de propriété.....	36
3.2. Assistance technique.....	36
3.3. Fiabilité et sécurité .....	38
3.4. Interopérabilité .....	40
3.5. Maturité et longévité .....	41
4. Acquisition.....	44
4.1. Quand choisir les logiciels libres et ouverts? .....	44
4.2. Évaluation .....	45
4.3. Approvisionnement.....	46
4.4. Contrats de services .....	48
5. Utilisation .....	51
5.1. Licences privilégiées .....	51
5.2. Conditions et restrictions .....	53
6. Adaptation et intégration.....	55
6.1. Conditions et restrictions .....	55
6.2. Identification du code source .....	56
6.3. Compatibilité des licences .....	57
7. Distribution.....	61
7.1. Distributions internes.....	61
7.2. Distributions externes .....	62
7.3. Contributions aux communautés.....	63
8. Conclusion .....	66
9. Contacts.....	67

## Historique

<b>Auteur</b>	<b>Description</b>	<b>Date</b>	<b>Version</b>
Lexum Pierre-Paul Lemyre Marie-Odile Désy	Création initiale (V1)	Mars 2007	0.9
Secrétariat du Conseil du trésor (SCT)	Vérification finale (V1)	Mars 2007	1.0
Lexum Pierre-Paul Lemyre Marie-Odile Désy Richard Willemant Mouhamadou Sanni Yaya	Version 2	Décembre 2010	1.1
Lexum	Commentaires SCT	Mars 2011	1.2
Secrétariat du Conseil du trésor (SCT) Direction de l'architecture des services gouvernementaux (DASG)	Vérification finale	Avril 2011	2.0

# 1. Introduction

[1] L'objectif du présent guide est de fournir de l'information détaillée ainsi que de définir de bonnes pratiques en ce qui concerne le recours aux logiciels libres et ouverts par les organismes publics du gouvernement du Québec. Il constitue l'un des outils destinés à appuyer l'administration publique dans son mandat de tirer profit des logiciels libres, établi par la Politique-cadre sur la gouvernance et la gestion des ressources informationnelles des organismes publics.

[2] Ce guide ne vise pas à diriger l'administration publique vers une stratégie technologique centrée sur les logiciels libres et ouverts, mais plutôt à encourager leur utilisation lorsqu'ils s'avèrent être le meilleur choix. Il cherche à favoriser et à encadrer leur intégration dans le contexte des processus d'approvisionnement existants. La prise en compte des lignes directrices qui y sont incluses devrait mener à une meilleure compréhension du phénomène des logiciels libres et ouverts et favoriser une utilisation optimale de l'ensemble des possibilités logicielles.

[3] Les logiciels libres et ouverts étant principalement le fruit d'une gestion alternative de la propriété intellectuelle, et plus particulièrement du droit d'auteur, le guide aborde en détail les impacts de cette gestion sur l'Administration et propose des stratégies pour en tirer le meilleur parti. En ce sens, il fixe le cadre juridique essentiel à une saine intégration des logiciels libres et ouverts par l'administration publique. Il traite également des enjeux qui en découlent, ainsi que de quelques questions techniques qui y sont liées.

## 1.1. Qu'est-ce qu'un logiciel libre et ouvert?

### Définition

[4] Un logiciel libre et ouvert est avant tout un logiciel dont l'utilisation, l'adaptation et la distribution sont soumises à des conditions permissives. Les utilisateurs de logiciels libres et ouverts bénéficient d'une liberté substantielle. Cette liberté constitue le cœur du mouvement du libre et repose sur le principe fondamental de la libre circulation des connaissances.

[5] Techniquement parlant, les logiciels libres et ouverts ne sont pas différents des autres types de logiciels. Leur caractère distinctif réside plutôt dans la méthode innovatrice utilisée par leurs auteurs pour gérer la propriété intellectuelle. En effet, plutôt que d'avoir recours aux protections accordées par le régime du droit d'auteur pour contrôler l'utilisation de leurs œuvres, ces derniers les utilisent pour faciliter et promouvoir leur distribution à très grande échelle. Ainsi, le terme « libre » fait référence à la liberté accordée à leurs utilisateurs. Cependant, cela ne signifie pas que ces logiciels soient libres de tous droits. Contrairement à ce que certains pourraient croire, les logiciels libres et ouverts ne font pas tous partie du domaine public. L'acquisition de la plupart d'entre eux fait l'objet d'un contrat de licence et c'est ce dernier qui définit les droits et les obligations qui y sont associés.

### Logiciels libres vs logiciels ouverts

[6] L'expression « logiciels libres » a initialement été popularisée par la Free Software Foundation (FSF). En 1998, une rupture s'est opérée au sein de cette communauté. Certains programmeurs, principalement sous l'influence de M. Éric Raymond et de M. Bruce Perens, se sont distancés de l'idéologie de la FSF, qu'ils jugent mal adaptée à la réalité économique d'aujourd'hui. Selon eux, l'expression « logiciel libre » n'était pas susceptible d'inciter l'industrie du logiciel à adopter leur modèle de développement parce qu'elle implique la prédominance de notions éthiques et morales. Ils adoptèrent donc une nouvelle stratégie fondée sur la notion de logiciels ouverts (*open source*).

[7] Tout comme la définition de la FSF, celle de l'Open Source Initiative (OSI) protège les droits d'utilisation et de redistribution, la disponibilité du code source et le droit de le modifier. En fait, les logiciels libres et les logiciels ouverts sont théoriquement identiques. Seule la terminologie a été modifiée afin de mettre l'accent sur la disponibilité du code source plutôt que sur la liberté de l'utilisateur. Aussi, la distinction est surtout philosophique, le modèle ouvert mettant l'accent sur le pragmatisme alors que le modèle libre repose sur l'éthique.

[8] Toutefois avec le temps, d'un point de vue pratique, certaines différences sont apparues entre les deux mouvements. Celles-ci concernent principalement les entreprises qui offrent des logiciels en dévoilant leur code source tout en restreignant les autres droits essentiels de l'utilisateur. Ces pratiques ont parfois été acceptées par les promoteurs des logiciels ouverts alors qu'elles sont absolument rejetées par les partisans des logiciels libres. Ce fut le cas, par exemple, de la version initiale de la Apple Public Source License (APSL) retenue par Apple pour son système d'exploitation Mac OS X. Celle-ci, en plus d'obliger les licenciés à publier toute version modifiée du programme informatique qu'ils ont déployée et à en notifier Apple, prévoyait la possibilité pour Apple de révoquer la licence dans l'hypothèse où l'entreprise serait poursuivie pour contrefaçon d'un droit d'auteur ou d'un brevet. Ces restrictions de la liberté des licenciés satisfaisaient les critères de l'OSI, mais non ceux de la FSF. Le débat a pris fin avec les modifications apportées à l'APSL lors de la publication de sa version 2.0.

[9] D'un point de vue juridique, il n'existe pas de différence entre les logiciels libres et les logiciels ouverts. Pour s'en convaincre, il suffit de constater que les deux mouvements reposent sur la même idéologie et les mêmes documents juridiques. Pour cette raison, les termes « libres et ouverts » sont adoptés tout au long de ce guide pour les qualifier.

*Critères*

[10] De façon générale, un logiciel peut être considéré « libre et ouvert » s'il est possible :

1. de l'exécuter pour n'importe quel usage;
2. d'avoir accès à son code source;
3. d'en étudier le fonctionnement et de l'adapter à des besoins spécifiques;
4. d'en redistribuer des copies originales ou modifiées.

*Code source*

[11] Une caractéristique essentielle des logiciels libres et ouverts est l'accessibilité du code source. Contrairement à la forme binaire du logiciel, qui n'est intelligible que par l'unité centrale d'un ordinateur, le code source est une version du logiciel compréhensible par l'être humain. En y ayant accès, des personnes indépendantes de son concepteur peuvent étudier, analyser et modifier le code source.

[12] Traditionnellement, les éditeurs de logiciels ont restreint l'accès à leur code source dans le but de protéger leur propriété intellectuelle. Dans ces circonstances, le logiciel est distribué uniquement sous sa forme binaire exécutable. Il en résulte que pour obtenir toute information concernant le fonctionnement du logiciel il faut étudier son comportement lors de l'exécution. Cette technique s'avère beaucoup moins révélatrice que l'étude directe du code source.

*Origine*

[13] Le mouvement des logiciels libres et ouverts origine principalement des États-Unis. Sa naissance est souvent associée avec le projet GNU amorcé par M. Richard Stallman au début des années 1980. Ce projet avait pour but de créer un système d'exploitation libre de type UNIX. Le logiciel libre et ouvert le plus connu aujourd'hui est le système d'exploitation GNU/Linux (plus connu sous le nom de Linux), dérivé de ce projet.

*Évolution*

[14] Au cours des vingt dernières années, le marché des logiciels libres s'est peu à peu transformé. D'un mouvement idéologique soutenu par une poignée de spécialistes, il s'est peu à peu ouvert aux besoins et aux exigences hétérogènes des multiples acteurs de

l'industrie du logiciel. Au début des années 2000, des investissements substantiels de plusieurs grands éditeurs de logiciels dont IBM et SunMicrosystem l'ont propulsé à l'avant de la scène et ont accéléré son développement. Les sommes ainsi investies ont favorisé la mise en place d'un marché du logiciel libre et ouvert, actif et diversifié. Depuis, on constate une consolidation progressive avec le reste de l'industrie. D'un côté, les entrepreneurs initialement issus du mouvement du libre ont dû apprendre à développer des modèles d'affaires susceptibles d'assurer la pérennité de leurs initiatives. De l'autre, de nombreux acteurs de l'industrie traditionnelle du logiciel ont su tirer profit de cet apprentissage.

*Exemples*

[15] Un écosystème libre et ouvert complet est aujourd'hui en place et il existe peu de domaines d'activité qui n'ont pas été bouleversés par l'arrivée massive de logiciels libres et ouverts sur le marché. Alors que le système d'exploitation Linux était auparavant réservé à une utilisation côté serveur, la distribution Ubuntu propose désormais une solution libre et ouverte pour les stations de travail. Parmi les projets grand public qui connaissent un succès croissant se trouvent le navigateur Internet Firefox et la suite de bureautique OpenOffice. En parallèle, plusieurs des environnements de programmation les plus populaires (Java, Perl, PHP, etc.) reposent eux-mêmes sur des logiciels libres et les grands projets associés au développement d'Internet (Apache, OpenSSH, etc.) continuent toujours d'évoluer en ce sens.

*Développement  
collaboratif*

[16] L'une des principales raisons qui poussent de nombreuses entreprises à s'ouvrir aux logiciels libres et ouverts est leur volonté de bénéficier des avantages associés au mode de développement collaboratif qui les caractérise. En effet, les termes permissifs des licences libres et ouvertes et le développement d'Internet ont favorisé l'apparition d'une nouvelle forme de développement logiciel, où chaque utilisateur est en mesure d'apporter sa contribution. L'évolution des logiciels libres et ouverts s'opère en ligne et toute personne intéressée est en mesure d'y prendre part. Cette approche incite des acteurs aux intérêts variables à joindre leurs énergies. Ainsi, des individus à la recherche de reconnaissance y côtoient des employés d'entreprises qui espèrent en tirer un certain profit et des utilisateurs dont le principal souci est d'économiser. Ce mode de développement ouvert encourage la collaboration entre ceux-ci afin de concevoir, de déboguer et d'optimiser les logiciels qu'ils partagent et utilisent en commun.

*Communautés*

[17] Plus souvent qu'autrement, ce mode de développement collaboratif pousse les différents utilisateurs concernés à se regrouper. Sous l'impulsion d'un promoteur (généralement l'auteur original du logiciel), ces regroupements prennent la forme de communautés informelles qui naissent et qui meurent en fonction de l'intérêt commun pour une solution logicielle. Ce sont ces communautés qui voient à :

1. définir les orientations et la stratégie d'évolution du logiciel;
2. organiser son développement et son débogage;
3. assister ses utilisateurs.

[18] Chacune d'elles est régie par ses propres règles de gouvernance et la collaboration qui en résulte peut prendre plusieurs formes, dont la mise en place de forums et de listes de discussion, de sites Web spécialisés, etc.

Tableau 1 – Principales différences entre les droits associés aux logiciels libres et ouverts et aux logiciels propriétaires

	Logiciels libres et ouverts	Logiciels propriétaires
<b>Accès au code source</b>	✓	
<b>Utilisation autorisée</b>	✓	✓
<b>Modification autorisée</b>	✓	
<b>Distribution autorisée</b>	✓	
<b>Redevance financière</b>		✓
<b>Exclusivité d'un fournisseur</b>		✓

Références

AFUL, « Qu'est-ce qu'un logiciel libre et ouvert? », 2011, Web : <<http://aful.org/ressources/logiciel-libre>>

CHEVALIER (B.), *Logiciels libres, Open source : qu'est-ce que c'est?*, H & K, Paris, 2005

ELIE (F.), *Économie du logiciel libre*, Paris, Eyrolles, 2009

OPEN SOURCE INITIATIVE, " The Open Source Definition (Annotated) ", 2011, Web : <<http://www.opensource.org/docs/definition.php>>

ST-AMANT (K.), STILL (B.), *Handbook of Research on Open Source Software : Technological, Economic, and Social Perspectives*, Hershey, IGI Publishing, 2007

WILLIAMS (S.), STALLMAN (R. M.), MASUTTI (C.), *Richard Stallman et la révolution du logiciel libre. Une biographie autorisée*, Paris, Eyrolles, 2010

## 1.2. Droit d'auteur et licences logicielles

Nature de la protection

[19] De façon générale, l'auteur d'un logiciel est présumé être le premier titulaire des droits sur celui-ci. Puisqu'une fois fixé sur un support, le logiciel possède avant tout une forme littéraire, ce sont les règles du droit d'auteur qui en assurent la protection. Ce droit de propriété sur l'immatériel s'est développé au cours des siècles afin de favoriser la création artistique et l'avancement des connaissances, en garantissant aux innovateurs un monopole sur l'exploitation de leurs créations.

Protection internationale

[20] À l'international, le droit d'auteur est régi par la Convention de Berne pour la protection des œuvres littéraires et artistiques (Convention de Berne). Ce document établit un régime minimal de protection des œuvres littéraires et artistiques qui assure la conservation des intérêts de l'auteur. Celui-ci se voit accorder des droits sur l'expression de sa création, mais

non sur l'idée qui en est à l'origine. Bien que le traitement des infractions aux droits d'auteur demeure de compétence nationale, la Convention de Berne permet d'étendre les droits de l'auteur au territoire de tous les États qui en sont membres.

*Réforme du droit d'auteur*

[21] Au Canada, le droit d'auteur relève d'une compétence de juridiction fédérale et c'est la Loi sur le droit d'auteur qui en précise les règles. Cette loi, qui fait présentement l'objet d'une réforme, est en voie d'être révisée par le Projet de loi C-32 sur la modernisation du droit d'auteur. Bien que ce projet de loi propose des mesures destinées à assurer une gestion adéquate du droit d'auteur à l'ère d'Internet, par exemple en interdisant le contournement des mesures techniques de protection ou en créant une exception d'utilisation pour du contenu non commercial généré par un utilisateur, il ne modifie en rien les principes fondateurs du droit d'auteur applicables au Canada. En ce sens, l'adoption ou non de la présente réforme du droit d'auteur ne devrait changer en rien l'analyse présentée dans ce guide.

*Œuvre originale*

[22] En fonction de ces règles, l'auteur d'un logiciel est présumé être le premier titulaire des droits de propriété intellectuelle sur celui-ci. Toutefois, au Canada, pour que son auteur puisse bénéficier du régime de protection accordé par la Loi sur le droit d'auteur, il faut que le logiciel possède les caractéristiques d'une œuvre originale. À cet égard, les critères formulés par la Cour suprême dans l'arrêt CCH Canadienne Ltée c. Barreau du Haut-Canada ont trait à l'exercice d'un talent et d'un jugement dans la production. En fait, il suffit tout simplement que le logiciel ait été créé de façon indépendante. Ainsi, deux logiciels permettant d'effectuer le même calcul mathématique peuvent avoir droit à cette protection, à la condition qu'aucun des deux n'ait été copié sur l'autre. Les objectifs, les sources d'inspiration et même le code informatique peuvent être identiques, l'originalité de chacun des logiciels dépendra de l'effort indépendant ayant été investi.

*Opposabilité*

[23] L'objectif fondamental du droit d'auteur est d'accorder un monopole d'exploitation à la personne qui est titulaire des droits, qu'il s'agisse d'un logiciel ou de tout autre type d'œuvre. Il protège avant tout un certain nombre de droits, dits patrimoniaux, qui limitent les actions susceptibles d'avoir un impact économique sur le titulaire des droits. Quelques-uns touchent particulièrement les logiciels, dont le droit de l'exécuter, de le reproduire, de le convertir à un autre langage informatique, de le communiquer au public et de le louer; et ce, peu importe que ce titulaire soit l'auteur lui-même, son employeur, ou une autre personne s'étant fait céder les droits. Dans tous les cas, le monopole octroyé au titulaire est opposable à toute personne soumise à la loi canadienne. Cela permet aux développeurs de logiciels libres et ouverts de faire valoir les protections accordées par la Loi sur le droit d'auteur aux tiers, c'est-à-dire à toute personne n'ayant pas consenti aux termes de la licence qu'ils privilégient. De la même façon, les règles génériques prévues par la loi devraient trouver application dans l'hypothèse où la licence serait invalidée.

*Fixation sur un support*

[24] Pour être protégé, un logiciel doit nécessairement être fixé sur un support quelconque, qu'il s'agisse d'un fichier informatique, d'une bande magnétique ou d'un coin de nappe. À ce propos, peu importe qu'il y soit fixé sous la forme de code source (version lisible par un être humain) ou sous la forme binaire (version exécutable par un ordinateur), puisque les deux font l'objet d'une protection. Par contre, tant qu'il demeure dans l'esprit de son concepteur, un logiciel reste hors de portée du champ d'application de la Loi sur le droit d'auteur par manque de support matériel consignait l'expression de cette idée. D'ailleurs, il ne faut pas confondre la propriété sur ce support qui est matérielle avec la propriété de l'œuvre qui est immatérielle. Bien que l'acheteur d'un cédérom soit propriétaire de ce disque, il ne possède a priori aucun droit d'auteur sur le contenu.

*Marquage*

[25] Dès qu'un logiciel répond aux conditions décrites ci-dessus, le régime du droit d'auteur lui est automatiquement appliqué, sans aucune formalité. Il n'est pas nécessaire de l'enregistrer ou de le marquer, le droit d'auteur existe dès sa création. Cependant, cela ne veut pas dire que ces opérations soient inutiles. Dans les deux cas, elles visent à faciliter la

preuve de l'existence d'un droit d'auteur, ainsi que la titularité de l'auteur. En conséquence, l'ajout d'un en-tête mentionnant le nom du développeur et l'existence d'un droit d'auteur dans le code source d'un logiciel empêchent les tiers l'ayant téléchargé d'invoquer leur ignorance à ce sujet.

*Exceptions*

[26] Par ailleurs, les protections accordées par le droit d'auteur ne sont pas absolues. Ainsi, il demeure toujours possible de reproduire un logiciel à des fins d'étude privée ou de recherche. De la même façon, on peut le reproduire dans le cadre d'une critique ou d'un compte rendu dans la mesure où l'auteur est mentionné. Ces exceptions permettent, par exemple, de faire circuler un logiciel libre et ouvert au sein d'une entreprise ou de citer du code dans un article scientifique, sans nécessairement reproduire la licence à laquelle il est soumis. La réforme du droit d'auteur en cours prévoit notamment la création de quelques exceptions supplémentaires à ce niveau.

[27] En matière de logiciel, la loi prévoit également deux exceptions spécifiques autorisant sa reproduction sans le consentement du titulaire des droits. La première autorise le propriétaire d'un exemplaire à en faire une copie, soit par adaptation, modification ou conversion, afin d'assurer la compatibilité du logiciel avec un ordinateur en particulier. La seconde lui permet d'effectuer une copie de sauvegarde, qui doit toutefois être détruite dès qu'il n'est plus propriétaire de l'exemplaire. Pour le reste, le recours à ces exceptions a peu d'applications potentielles en matière de logiciels libres et ouverts, vu le caractère permissif de leurs licences qui, le plus souvent, autorisent déjà les actes en question.

*Reproduction partielle*

[28] Par ailleurs, comme l'entend la loi canadienne, le droit de reproduction concerne la totalité ou une partie importante du logiciel. Chaque ligne de code, prise individuellement, n'est donc pas protégée et il est toujours possible de copier des extraits d'un logiciel. Ainsi, dans l'affaire *Delrina Corp. v. Triolet Systems Inc.* un juge a établi que 60 lignes de code ne constituent pas une partie importante d'un logiciel qui en compte plus de 14 000, particulièrement si leur réécriture aurait réclamé environ 20 minutes de programmation. Le code source des logiciels libres et ouverts étant rendu disponible, ceux-ci sont particulièrement sujets à être ainsi copiés. L'évaluation de l'importance de la reproduction est donc déterminante pour assurer leur protection. Les critères généralement reconnus à ce niveau sont :

- ⤴ Le caractère distinct de la partie copiée (Est-ce qu'elle a nécessité des efforts, du talent ou de l'ingéniosité?);
- ⤴ Le degré de protection mérité par un logiciel de ce type;
- ⤴ Les impacts sur l'exploitation du logiciel;
- ⤴ L'enrichissement injustifié de celui qui a reproduit une partie du logiciel aux dépens de son auteur;
- ⤴ L'existence d'un seul marché où les deux logiciels sont en compétition.

*Fin du droit d'auteur*

[29] Les droits accordés sont également limités dans le temps. Ainsi, au Canada, le droit d'auteur cesse d'exister cinquante ans après la mort de l'auteur. Dans le cas des logiciels créés en collaboration, comme c'est souvent le cas des logiciels libres et ouverts, cette période s'étend à cinquante ans après la mort du dernier auteur. Toutefois, lorsque des auteurs étrangers ont contribué au développement du logiciel, ceux-ci ne sont pas autorisés à se prévaloir, au Canada, d'une période de protection plus longue que celle qui existe dans leur pays d'origine. Après cette période, le logiciel tombe dans le domaine public et peut donc être utilisé par tous sans aucune contrainte. Il est vrai que vu la disproportion entre le temps de protection accordé et la durée de vie moyenne des logiciels, ces règles ont peu de chance d'avoir un jour un impact réel sur les logiciels libres et ouverts.

*Contrefaçon*

[30] Enfin, l'article 27 de la Loi sur le droit d'auteur précise que l'accomplissement d'un geste réservé au titulaire de l'un des droits énumérés ci-dessus constitue une violation du droit d'auteur. En ce qui concerne les logiciels, la jurisprudence et la doctrine ont déterminé

que cette contrefaçon peut être établie en effectuant un test en cinq étapes :

1. Abstraction : séparer l'idée de l'expression;
2. Évaluation : le logiciel est-il original?;
3. Comparaison : y a-t-il assez de similitudes?;
4. Filtration : ces similitudes sont-elles justifiées?;
5. Quantification : les similitudes protégées portent-elles sur une partie importante du logiciel.

*Recours civils*

[31] Lorsque ce test s'avère positif, le titulaire des droits sur un logiciel libre et ouvert se voit octroyer toute une série de recours afin de faire respecter son droit d'auteur auprès de la personne ayant commis l'infraction. Le plus connu de ces recours est certainement l'injonction qui permet de forcer le contrevenant à respecter les termes de la licence ou à cesser ses agissements. Le recours en dommages et intérêts, régi par le droit civil québécois, est également disponible quoique le dommage puisse parfois s'avérer difficile à prouver pour certains développeurs distribuant le code source de leur logiciel gratuitement sur Internet. La loi prévoit également des dommages statutaires permettant à un juge de remplacer les dommages et intérêts par un montant variant entre 500 \$ et 20 000 \$. L'octroi d'une proportion des profits réalisés par le contrevenant est un autre recours susceptible de résulter en un avantage économique pour le titulaire des droits. Ceci est particulièrement vrai dans le cas d'un logiciel libre qui aurait été illégalement incorporé à l'intérieur d'un logiciel propriétaire ayant connu un certain succès.

*Infraction criminelle*

[32] En plus des recours civils auxquels il s'expose, le contrevenant à la Loi sur le droit d'auteur commet également une infraction pénale. Cet élément constitue peut-être la principale protection octroyée aux logiciels libres et ouverts, puisque celui qui est reconnu coupable de contrefaçon encourt une amende pouvant aller jusqu'à un million de dollars et un emprisonnement maximal de cinq ans.

*Démembrement*

[33] Au-delà de ces considérations générales, la principale particularité du droit d'auteur réside dans la possibilité de le démembrer à l'infini. Il appartient donc au titulaire des droits sur un logiciel de déterminer qui pourra en disposer, de quelle façon, pour quelle durée, à quel endroit, etc. Deux techniques permettent de transiger les protections accordées par le droit d'auteur. L'une d'elle est la cession, l'autre est la concession.

*Cession*

[34] La cession est une technique qui permet au titulaire des droits sur un logiciel de les céder partiellement ou intégralement à un tiers. La cession implique le transfert de la titularité de certains ou de tous les droits sur un logiciel. Ainsi, une fois le transfert opéré, l'ancien titulaire se retrouve dépourvu des droits concernés qui appartiennent désormais au cessionnaire.

*Concession (licence)*

[35] Cependant, lorsqu'il s'agit d'exploiter un logiciel, la concession s'avère être la méthode la mieux adaptée. La concession, exprimée dans une licence autorise les tiers à accomplir certains actes qui autrement auraient enfreint le droit d'auteur, tout en conservant sa titularité. À cause de ces particularités, les logiciels sont presque exclusivement offerts aux utilisateurs dans le cadre d'une licence.

[36] Les licences de droit d'auteur ont pour effet d'établir une relation contractuelle entre le titulaire des droits et l'utilisateur du logiciel. D'un côté, le donneur de licence exprime son intention de concéder une partie de ses droits exclusifs sur le logiciel, alors que de l'autre, le licencié manifeste son consentement à utiliser le logiciel dans le respect des conditions précisées par la licence. Ce consentement peut être constaté de multiples façons : par la pression d'un bouton s'affichant à l'écran ou encore par la simple utilisation du logiciel.

La principale conséquence découlant de l'établissement de cette convention est l'assujettissement de la licence au droit des contrats. Ainsi, c'est le droit privé, autrement dit le droit civil du Québec et non la loi canadienne sur le droit d'auteur, qui trouve ici application.

[37] Puisque le régime du droit d'auteur accorde à l'auteur d'un logiciel l'exclusivité des droits, la rédaction des licences de logiciel bénéficie d'une très grande flexibilité. En conséquence, il existe un nombre quasi illimité de licences différentes. Certaines sont extrêmement restrictives et d'autres très libérales. Il est cependant possible de les regrouper à l'intérieur de quelques catégories qui représentent assez fidèlement le contexte dans lequel les logiciels sont généralement mis à la disposition de leurs utilisateurs.

*Licences  
propriétaires*

[38] Traditionnellement, les licences propriétaires visent à limiter autant que possible les droits concédés aux utilisateurs. Ces licences interdisent de copier, de modifier ou de redistribuer la version du logiciel qui est remise. En agissant de la sorte, le titulaire des droits se réserve l'exclusivité de l'exploitation du logiciel. Les licences propriétaires n'accordent donc au licencié qu'un droit d'utilisation du logiciel, auquel se greffent de nombreuses restrictions. Les autres limitations ont trait, par exemple, aux garanties accordées, aux questions de responsabilité, au choix de la juridiction, etc. Et, question d'assurer l'efficacité de ces restrictions, les logiciels propriétaires sont généralement distribués sous forme d'exécutables binaires incompréhensibles pour l'être humain; ils peuvent donc être utilisés, mais seront difficilement modifiés.

*Techniques de  
distribution*

[39] Les éditeurs de logiciels propriétaires ont recours à différentes techniques de distribution sur Internet. Par exemple, la redistribution des gratuits (*freewares*) est généralement autorisée, car ils sont distribués gratuitement dès l'origine. Toutefois, leur utilisation demeure limitée et leur modification reste interdite. De la même façon, les partagiciels (*sharewares*) peuvent être redistribués et utilisés gratuitement pendant une certaine période de temps. Cependant, à l'expiration de ce délai, le licencié doit verser une redevance au donneur de licence pour continuer à utiliser le logiciel. Il n'en demeure pas moins que les éditeurs de ces logiciels maintiennent un contrôle serré de leur propriété intellectuelle et évitent d'octroyer aux utilisateurs les libertés caractéristiques des logiciels libres et ouverts.

*Licences semi-  
libres*

[40] Enfin, certains donneurs de licence, bien que séduits par les principes qui sous-tendent les logiciels libres et ouverts, ne sont pas prêts à subir toutes les conséquences qu'entraîne le recours à une licence de ce type. Cette hésitation a causé l'apparition de toute une variété de licences dites semi-libres. Ces licences sont beaucoup plus permissives que les licences propriétaires et prévoient toute la mise à disposition du code source, mais posent certaines conditions aux libertés fondamentales, caractéristiques des logiciels libres et ouverts.

[41] Dans certains cas, il s'agit de logiciels dont la licence accorde des droits d'utilisation, de redistribution et de modification, mais qui en limite l'exercice aux usages sans but lucratif ou exige le paiement d'une redevance dans le cas d'un usage commercial. Ces licences à source publique permettent d'éviter que des tiers réalisent des bénéfices en utilisant ou en distribuant un logiciel libre et ouvert sans que le titulaire des droits d'auteur puisse en bénéficier. C'est le cas, par exemple, de la Aladdin Free Public License (AFPL) à laquelle sont soumises les versions gratuites des technologies Ghostscript. Ce type de licence enfreint la première des libertés recherchées par les logiciels libres et ouverts, soit celle concernant la non-discrimination quant à l'utilisation du logiciel.

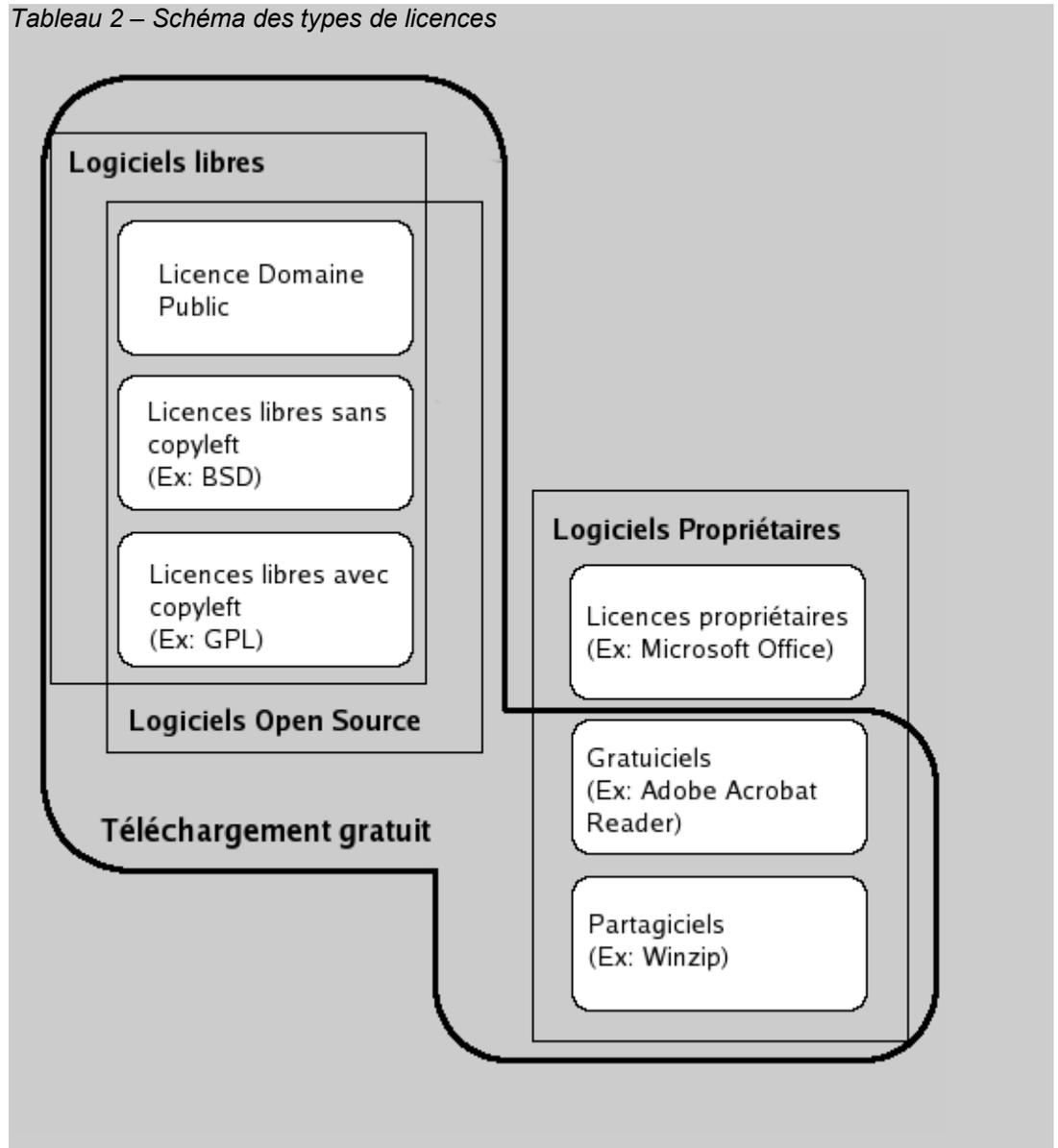
[42] Parfois, il s'agit de logiciels dont la licence autorise la modification du code source, mais à la condition que toute implémentation de ce code modifié respecte certains standards définis par le donneur de licence. Cela équivaut donc à soumettre la modification

et la distribution de ce code à l'approbation de ce dernier.

C'est, entre autres, le scénario privilégié par Sun Microsystems dans le cadre de la Sun Community Source License (SCSL).

[43] En réponse aux demandes provenant de sa clientèle, même Microsoft a développé un mode de concession semi-libre par l'intermédiaire du partage de code source. Sa principale caractéristique est de permettre aux licenciés d'étudier une partie du code mis au point par Microsoft. Toutefois, la modification et la redistribution de ce code sont fortement restreintes.

Tableau 2 – Schéma des types de licences



*Licences libres et  
ouvertes sans  
« copyleft »*

[44] Les licences libres et ouvertes sans « copyleft » (voir [49]) autorisent généralement l'utilisation, la modification et la redistribution du code source en imposant très peu de conditions à l'utilisateur. Mis à part l'obligation de divulguer l'existence de droit d'auteur sur le logiciel et de reproduire la licence, elles précisent simplement que le titulaire des droits ne fournit aucune garantie et ne saurait être responsable des dommages résultant de son utilisation.

[45] Les licences libres et ouvertes sans « copyleft » n'interdisent pas d'inclure le code source auquel elles s'attachent à l'intérieur de solutions logicielles régies par une licence différente. Cependant, il faut que les quelques conditions mentionnées plus haut soient respectées. Ainsi, tout licencié peut modifier un logiciel de ce type et le redistribuer sous la licence de son choix, y compris une licence propriétaire beaucoup plus restrictive.

[46] Le statut juridique des licences libres et ouvertes sans « copyleft » se rapproche de celui des logiciels du domaine public. Ceci s'explique notamment par l'étendue des droits concédés par ces licences. En effet, elles autorisent l'utilisation, la modification et la redistribution du code source, sans imposer d'autres restrictions. Toutefois, les logiciels qui leur sont soumis ne font pas partie du domaine public, puisque les licences sans « copyleft » précisent que le titulaire des droits d'auteur conserve ses droits. D'ailleurs, elles contiennent souvent l'obligation de divulguer l'existence de ces droits à l'intérieur des versions modifiées du logiciel.

*Apache*

[47] Apache 2.0 est l'exemple par excellence d'une licence libre et ouverte sans « copyleft ». Celle-ci s'applique à l'ensemble des projets de la Apache Software Foundation (ASF), ce qui inclut de nombreux logiciels largement distribués, comme le serveur Apache Web, le moteur de recherche Lucene et le système de gestion de version Subversion. Celle-ci autorise l'utilisation, la modification et éventuellement la redistribution du code source de ces projets sous toute autre licence, mais requiert une reconnaissance de l'ASF et l'inclusion du texte de la licence, et interdit l'utilisation des marques de commerce de l'ASF.

*Licences libres et  
ouvertes avec  
« copyleft »*

[48] Les licences libres et ouvertes avec « copyleft » permettent également au licencié d'utiliser, de modifier et de redistribuer le logiciel concerné. Toutefois, leur particularité consiste à attacher les mêmes libertés à tous les logiciels dérivés du code source original, grâce à l'inclusion d'une clause obligeant le licencié à distribuer ses modifications sous les termes de la même licence. Par cette technique, les licences avec « copyleft » empêchent le licencié de redistribuer une version modifiée du logiciel sous des termes plus restrictifs.

[49] Le terme « copyleft » fut mis en avant par la FSF et s'oppose au copyright, équivalent anglophone du droit d'auteur. L'objectif du « copyleft » est d'utiliser les protections accordées par le droit d'auteur afin de garantir la liberté d'utilisation et de modification des logiciels.

*GPL*

[50] La General Public License (GPL), la licence ayant introduit le concept du « copyleft », est sans contredit la licence libre et ouverte la plus utilisée à travers le monde. Elle grève l'ensemble des distributions de Linux, qui reposent toutes sur des modifications apportées au noyau original du système d'exploitation, lui-même soumis à la GPL. Une grande partie des logiciels concernés sont toujours couverts par la version 2 de la GPL, rédigée en 1991, bien que la version 3 de la licence ait été lancée officiellement en 2007.

Tableau 3 – Liste des licences libres et ouvertes et semi-libres les plus populaires

Rang	Licence	% des projets
1	GNU General Public License (GPL) 2.0	45,99 %
2	Artistic License (Perl)	8,50 %
3	GNU Lesser General Public License (LGPL) 2.1	8,32 %
4	MIT License	7,83 %
5	GNU General Public License (GPL) 3.0	6,46 %
6	BSD License 2.0	6,24 %
7	Apache License 2.0	4,59 %
8	Code Project Open 1.02 License	2,57 %
9	Microsoft Public License (Ms-PL)	1,68 %
10	Mozilla Public License (MPL) 1.1	1,15 %

Données par Black Duck Software, 27/01/2010

*Domaine public*

[51] Finalement, certaines licences font entrer les logiciels qui leur sont soumis dans le domaine public. L'utilisation de ces logiciels n'est soumise à aucune condition particulière. Il s'agit de logiciels dont l'auteur et le titulaire des droits, le cas échéant, ont renoncé à leurs droits. Selon ce type de licence, le titulaire des droits ne conserve aucun privilège particulier sur le logiciel, qui peut ainsi être utilisé, modifié et redistribué sans aucune restriction.

*Doubles concessions*

[52] Par ailleurs, chaque licence n'étant qu'une concession individuelle octroyée à un utilisateur spécifique, le titulaire de droits d'un logiciel jouit d'une grande flexibilité. Rien ne l'empêche d'autoriser un autre licencié à utiliser le logiciel à d'autres conditions. Il est donc possible qu'un même logiciel soit soumis à différentes licences, ce qui permet parfois aux utilisateurs de choisir celle qui leur convient le mieux. Aussi, le mode sous lequel un logiciel est licencié est susceptible de changer avec le temps. C'est le cas lorsqu'une éventuelle évolution des droits concédés dans le cadre d'une seule et même licence est prévue, ou si un logiciel est distribué sous des conditions différentes à divers moments.

*Références*

Apple Computer c. Mackintosh Computers Ltd., (1987) 1 C.F. 173

BLACK DUCK SOFTWARE, "Top 20 Most Commonly Used Licenses in Open Source Projects", 2011, Web : <<http://www.blackducksoftware.com/oss/licenses#top20>>

CCH Canadienne Ltée c. Barreau du Haut-Canada, 2004, CSC 13, par. 24 <<http://www.canlii.org/fr/ca/csc/doc/2004/2004csc13/2004csc13.html>>

DEEK (F.) & MCHUGH (J. A.), *Open Source : Technology and Policy*, Cambridge, Cambridge University Press, 2007

Delrina Corp. v. Triolet Systems Inc. (199347 C.P.R. (3rd) 1

FREE SOFTWARE FOUNDATION, "GNU General Public License", 2011, Web : <<http://www.gnu.org/copyleft/gpl.html>>

IBM Corp. c. Ordinateurs Spirales Inc., (1985) 1 C.F. 190

*Loi sur le droit d'auteur*, L.R., 1985, ch. C-42, Web : <<http://www.canlii.org/fr/ca/legis/lois/lrc-1985-c-c-42/derniere/lrc-1985-c-c-42.html>>

OPEN SOURCE INITIATIVE, "The BSD License", 2011, Web : <<http://www.opensource.org/licenses/bsd-license>>

VAVER (D.), *Copyright Law*, Irwin Law, Toronto, 2000

WIKIPÉDIA, « Logiciel libre », 2011, Web : <[http://fr.wikipedia.org/wiki/Logiciel\\_libre#Licences](http://fr.wikipedia.org/wiki/Logiciel_libre#Licences)>

### 1.3. Les modèles d'affaires du libre

#### *Évolution*

[53] Nouveautés technologiques, les logiciels libres et ouverts se sont imposés en l'espace de vingt ans comme l'un des piliers de l'industrie informatique. Issu d'un environnement idéologique caractérisé par une forte croissance financé par du capital de risque et destiné à un marché de niche, le milieu du logiciel libre et ouvert a su s'intégrer à l'économie de marché globale. Ce succès a incité de nombreuses entreprises à s'y investir et, par le fait même, à développer de nouvelles façons de monnayer leur expertise dans un environnement où le code source qu'elles produisent circule librement. Du modèle classique de paiement à la copie, les entreprises œuvrant dans le domaine se sont graduellement tournées vers une offre plutôt orientée vers les services.

#### *Types d'entreprises*

[54] De façon générale, l'industrie du logiciel est caractérisée par deux types d'entreprises. D'un côté, les éditeurs de logiciels proposent un produit. Ceux-ci ciblent généralement le marché mondial avec une offre générique. À l'inverse, les sociétés de services ciblent le marché local en proposant une offre personnalisée. Le marché des logiciels libres et ouverts n'échappe pas à cette règle et cette séparation entre produit et services continue d'y avoir cours.

#### *Modèles d'affaires*

[55] Ceci dit, il existe de nombreux modèles d'affaires soutenant les activités des éditeurs et des sociétés de services ayant choisi d'intervenir dans un environnement libre et ouvert, et plusieurs d'entre eux peuvent se chevaucher dans la même entreprise. Il demeure qu'au fil des ans un certain nombre d'entre eux ont su démontrer leur capacité à assurer la viabilité économique des entités qui s'y soumettent. Parmi eux, les modèles les plus répandus sont les suivants :

- ♣ Doubles licences;
- ♣ Noyau ouvert;
- ♣ Logiciel-service;
- ♣ Services à valeur ajoutée;
- ♣ Hybride.

#### *Doubles licences*

[56] Le modèle des doubles licences s'appuie sur la possibilité offerte par le droit d'auteur de concéder une même œuvre sous diverses licences. Il est possible d'offrir une version du code source d'un logiciel sous une licence libre avec « copyleft » de façon à promouvoir son adoption à grande échelle tout en empêchant son intégration dans des solutions propriétaires. En parallèle, le même code peut être distribué sous une licence propriétaire flexible pour les licenciés qui ne veulent pas s'engager à respecter les termes du « copyleft », et cela en contrepartie d'une redevance financière. C'est, entre autres, l'approche adoptée par MySQL AB qui distribue son système de base de données à la fois

sous la licence GPL (MySQL Community Server) et sous une licence propriétaire (MySQL Enterprise). Bien qu'initialement fort populaire, l'utilisation de ce modèle est aujourd'hui en décroissance.

*Noyau ouvert* [57] Plutôt que de distribuer la totalité de leur code source sous une licence libre et ouverte, d'autres entreprises choisissent d'en ouvrir uniquement le noyau. L'objectif est une fois de plus d'intéresser les masses en promouvant une large adoption du logiciel principal. Une fois cet objectif atteint, l'entreprise dispose d'une vaste clientèle susceptible d'acquérir des modules supplémentaires qui eux requièrent un déboursé. C'est le modèle d'affaires retenu par Red Hat qui fournit une distribution Linux de base sous GPL tout en offrant à ses clients une version commerciale accompagnée d'une multitude d'outils administratifs d'entreprise supplémentaires.

*Logiciel-service* [58] On fait souvent référence au modèle du logiciel-service sous les appellations de Software as a Service (SAAS) ou de Service Application Provider (SAP) qui n'est pas exclusif au domaine du logiciel libre et ouvert. Ce modèle est basé sur la prémisse que de nombreuses entreprises préfèrent impartir les fonctions de leur infrastructure informatique qui se situent à l'extérieur de leur champ de compétence principal. Aussi, plusieurs d'entre elles préféreraient s'abonner aux services d'hébergement et d'opération offerts par un opérateur spécialisé plutôt que de déployer un logiciel à l'interne. Pour l'opérateur qui fonde son offre sur un ou des logiciels libres et ouverts, il devient ainsi possible de coopérer avec les communautés de développeurs, tout en étant rémunéré pour son expertise spécialisée. Le succès des logiciels SugarCRM et Word Press repose sur cette approche. Celle-ci a présentement la cote et explique en grande partie le succès croissant des logiciels libres et ouverts au cours des dernières années.

[59] Cette approche est parfois combinée à celle du noyau ouvert, par l'offre de services supplémentaires fondée sur des modules qui eux ne sont pas disponibles dans le logiciel libre et ouvert original. Cette approche ne peut cependant pas être retenue lorsque la licence utilisée est la GNU Affero GPL, elle-même conçue pour garantir la disponibilité du code source des logiciels utilisés sous le modèle logiciel-service.

*Services à valeur ajoutée* [60] Il existe également une multitude de services connexes susceptibles d'intéresser les utilisateurs de logiciels libres et les entreprises œuvrant dans le secteur ont pris bien soin d'élaborer leur offre en conséquence. Au-delà du logiciel qui, lui, est disponible gratuitement sur Internet, les firmes spécialisées offrent du soutien technique, de la formation, des services-conseils, des services d'installation et d'intégration, etc. Grâce à la libre circulation du code source, de tels services sont tous offerts aussi bien par les distributeurs du logiciel original que par de pures sociétés de service dont la force consiste à fournir une capacité d'intégration personnalisée sur la base d'une plateforme commune. Selon une étude internationale réalisée en 2009, près de 50 % des entreprises actives dans le marché du logiciel libre et ouvert ont adopté ce modèle d'affaires.

*Hybride* [61] Finalement, il convient de noter que de plus en plus d'entreprises issues de l'industrie traditionnelle du logiciel choisissent aujourd'hui d'adapter leur modèle d'affaires dans l'objectif de répondre aux demandes de plus en plus nombreuses de leurs clients qui souhaitent obtenir un accès au code source de leurs logiciels. Bien que ce modèle puisse prendre plusieurs formes, sa mise en œuvre implique très souvent une ouverture du code source couplée à une licence semi-libre ou propriétaire assurant un encadrement plus serré de son utilisation, de sa modification et de sa distribution que s'il était soumis à une licence libre et ouverte.

*Justification* [62] Au-delà de tous ces modèles d'affaires potentiels, il existe une vaste gamme de raisons justifiant la décision prise par ces entreprises de rendre librement et ouvertement accessible le fruit de leurs efforts. À ce sujet, les sources de motivation les plus souvent invoquées sont :

▲ Favoriser la standardisation;

- ✦ Obtenir des ressources supplémentaires de développement;
- ✦ Réduire les coûts d'exploitation;
- ✦ Améliorer la qualité du logiciel;
- ✦ Accroître la confiance des utilisateurs dans le logiciel;
- ✦ Générer une communauté d'utilisateurs.

Marques de  
commerce

[63] Malgré cette diversité des modèles d'affaires et des sources de motivation, un trait commun unit tous les acteurs commerciaux actifs dans le domaine du logiciel libre. Il s'agit de l'importance du droit des marques pour assurer le caractère distinctif de leur offre. Bon nombre d'entre eux considèrent que le succès commercial est intimement lié à la popularité du nom du logiciel autour duquel ils gravitent. Pour cette raison, les marques de commerce couvrant ces noms prennent une valeur importante et sont activement protégées. Dans le même ordre d'idées, de plus en plus de licences libres et ouvertes incluent des dispositions concernant spécifiquement l'usage des marques de commerce s'attachant aux logiciels qu'elles grèvent.

COMMISSION OPEN SOURCE TELECOM VALLEY, « Livre Blanc : Intégration et publication de logiciels sous licences Open Source : Mode d'emploi », Telecom Valley, 2010, Web : <[http://www.telecom-valley.fr/util/news\\_detail.fr.htm?newsid=6713](http://www.telecom-valley.fr/util/news_detail.fr.htm?newsid=6713)>

FLOSSMETRICS, « Free/Libre Open Source Software : a guide for SMEs », 2009, Web : <<http://guide.flossmetrics.org>>

GAUGUIER (J.-J.), *L'industrialisation de l'open source*, Thèse, Université de Paris Dauphine, 2009

LINDMAN (J.), JUUTILAINEN (J.P.), ROSSI (M.), « Beyond the Business Model : Incentives for Organizations to Publish Software Source Code », 5th IFIP WG International Conference on Open Source Systems : Open source ecosystems : diverse communities interacting, Skovde, 2009

OPEN SOURCE STRATEGIES, « Open Source Business Models and Strategies : Research, Articles, and White Papers », 2010, Web : <<http://www.opensourcestrategies.org>>

PROWSE (T.), « What business models are currently used with open source software? », Open Source Business ressource, 2010, Web : <<http://www.osbr.ca/ojs/index.php/osbr/article/view/1157/1107>>

RASTETTER (Y.), *Le logiciel libre dans les PME*, Paris, Hermès Science Publications, 2008

WATSON (R.), BOUDREAU (M.-C.), YORK (P. T.), GREINER (M. E.), WYNN (D. J.), « The Business of Open Source », (2008) 51 Communications of the ACM, Web : <<http://portal.acm.org/citation.cfm?id=1330321&dl=ACM&coll=DL>>

## 2. Principaux enjeux juridiques

[64] Ce chapitre aborde quelques difficultés juridiques, d'ordre général, parfois soulevées lors de l'analyse des solutions fondées sur le recours aux logiciels libres et ouverts. Les risques et les avantages potentiels associés aux logiciels libres et ouverts y sont détaillés, ainsi que les techniques permettant de les balancer aussi objectivement que possible.

### 2.1. Validité juridique

*Enjeux*

[65] Les licences des logiciels libres et ouverts ont une nature et un contenu qui leur sont propres. Certains assimilent même les licences « copyleftées » à des contrats viraux, puisque celles-ci suivent d'elles-mêmes le contenu numérique. En raison de ces particularités, la validité de ces contrats et leur caractère contraignant ont été mis en doute. Cette question est primordiale, car leur validité permet d'assurer la sécurité juridique requise pour envisager leur utilisation. Alors que, pendant de nombreuses années, le droit a tardé à aborder directement les licences libres et ouvertes, on dispose aujourd'hui de quelques décisions de justice reconnaissant la validité de ces licences dans divers pays industrialisés. Bien que les tribunaux québécois n'aient encore rendu aucune décision en ce sens, la confrontation des licences libres et ouvertes avec le droit québécois laisse entrevoir un résultat similaire.

*Doctrine américaine*

[66] Dans le contexte américain, les arguments théoriques invoqués à l'encontre de la validité des logiciels libres et ouverts ont été de plusieurs ordres. Certains auteurs ont d'abord remis en question la viabilité de la philosophie de la FSF dans la mesure où celle-ci irait à l'encontre des principes promus par le droit d'auteur, soit « *promote the Progress of Science and useful Arts* » selon la Constitution américaine. Allant un peu plus loin, certains ont prétendu que les licences avec « copyleft » ne seraient pas exécutoires lorsque le licencié n'a pas contracté directement avec les nombreux donneurs de licence précédents puisque l'utilisateur ignore potentiellement toutes les implications de son geste. La question du consentement, poussée encore un peu plus loin, entraîne également des interrogations sur la capacité, pour un licencié d'un logiciel libre et ouvert, de sous-licencier à son tour le logiciel s'il n'est pas prouvé qu'il dispose des droits lui permettant d'accorder lui-même une licence. Finalement, les clauses de « *de rétrocession ou grantback* » présentes dans les licences avec « copyleft » ont fait douter certains auteurs de la validité de ces contrats en droit américain dans la mesure où celles-ci pourraient être considérées comme un « *copyright misuse* », forme d'abus du droit d'auteur, qui priverait alors le contrat de sa force contraignante.

*Jurisprudence américaine*

[67] Une partie de cet argumentaire a été fortement publicisée dans le cadre du conflit opposant le groupe SCO à plusieurs éditeurs de logiciel dont Novell, Red Hat et surtout IBM. SCO reprochait à IBM, sous le couvert de violations de ses engagements contractuels et du secret commercial, d'avoir introduit des parties du code d'Unix, sur lequel SCO détient les droits de propriété intellectuelle, dans le noyau de Linux. Par ailleurs, SCO soutenait que la GPL viole la Constitution des États-Unis, ainsi que les lois américaines sur le droit d'auteur et les brevets. Selon les prétentions de SCO, le système du « copyleft » équivaldrait à une négation totale du droit en vigueur et serait contraire aux buts et à l'esprit des textes mentionnés, qui protègent avec force les créations intellectuelles. Toutefois, malgré tout le battage médiatique, la cour n'a pas eu l'occasion de se pencher sur ces questions, puisqu'elle en est arrivée à la conclusion que c'est Novell, et non SCO qui détient la propriété intellectuelle sur le code d'Unix. Novell a par la suite abandonné la poursuite contre IBM.

[68] C'est dans le cadre de décisions beaucoup moins publicisées que les tribunaux américains ont eu à se pencher sur la force contraignante des licences libres et ouvertes, et jusqu'ici leur validité a été reconnue. Ainsi, en 2002, dans le contexte d'une injonction provisoire, la Cour du district du Massachusetts dans l'affaire MySQL v. Progress Software a implicitement considéré que la GPL était une licence contraignante bien que le juge n'ait pas fait droit à cette demande en l'absence de preuve d'un préjudice. De façon plus convaincante, en 2008, la Cour d'appel fédérale dans l'affaire Jacobsen v. Katzer est venue confirmer la validité juridique de l'Artistic Licence et l'obligation qu'elle contient de créditer les auteurs du logiciel qui lui est soumis pour leurs contributions antérieures.

*Doctrine française*

[69] En France, une situation similaire prévaut. Les principales difficultés évoquées par la doctrine ont trait à l'incompatibilité des licences libres et ouvertes avec le formalisme du Code français de la propriété intellectuelle. Par exemple, alors que celui-ci requiert une délimitation stricte des droits cédés quant au lieu et à la durée, les licences libres et ouvertes ne contiennent pratiquement jamais de telles précisions. En outre, la clause par laquelle le licencié renonce par avance à ses droits patrimoniaux sur le logiciel dérivé pose également une difficulté en droit français. Sur ce point, la loi interdit toute renonciation à un droit futur et prohibe toute cession globale d'œuvres futures. C'est en considération de cette incertitude juridique que le Commissariat à l'énergie atomique (CEA), le Centre national de recherche scientifique (CNRS) et l'Institut national de recherche en informatique et en automatique (INRIA) ont collaboré à la rédaction d'une licence de logiciel libre et ouverte compatible avec le droit français, la licence CeCILL. Aujourd'hui, la CeCILL se décline en trois versions, l'originale qui recherche une compatibilité avec la GPL, la CeCILL-B de type BSD et la CeCILL-C compatible avec la Lesser GPL (LGPL). Le même raisonnement a mené à la rédaction de la European Union Public Licence (EURL) qui veut respecter le droit des 27 États membres de l'Union européenne, tout en demeurant compatible avec la GPL.

*Jurisprudence française*

[70] Malgré l'insistance de la doctrine à affirmer l'inadéquation de la vaste majorité des licences libres et ouvertes avec le droit français, la Cour d'appel de Paris n'a pas hésité à reconnaître les effets contraignants de la GPL. Dans une décision du 16 septembre 2009, la cour oblige la société Edu4 à fournir à l'Association de formation professionnelle pour les adultes (AFPA), son client, le code source d'une version modifiée du logiciel VNC ayant été déployé dans ses salles de formation. En plus de clairement établir la validité juridique de la GPL en France, cette décision démontre que n'importe quel utilisateur est en mesure de réclamer l'accès au code source puisque les auteurs de VNC n'ont jamais pris part au litige.

*Jurisprudence allemande*

[71] Cette décision vient confirmer l'approche pragmatique retenue par la Cour de District de Munich, en Allemagne, dès 2004 dans l'affaire Netfilter/iptables v. Sitecom Germany GmbH. Il s'agissait alors de la première décision judiciaire se prononçant clairement sur la validité de la GPL et à en avoir ordonné l'exécution forcée. En l'espèce, la société Sitecom distribuait un logiciel de routage d'accès sans fil développé à partir d'un autre logiciel réalisé, lui, par le consortium Netfilter/iptables, et soumis à la GPL. Sitecom offrait son produit sans que son code source soit disponible et sans inclure la GPL, ni même y faire référence. Dans ces circonstances, la cour considéra que Sitecom avait violé les termes de la licence et l'a enjointe de s'y conformer ou de cesser de distribuer le logiciel mis en cause.

*Facteur de rattachement avec le Québec*

[72] Malgré cette reconnaissance internationale, il n'en demeure pas moins que, pour engendrer des effets contraignants sur le territoire québécois, les licences de logiciels libres et ouverts devront le plus souvent respecter le droit applicable au Québec. À cet effet, l'existence d'un facteur de rattachement entre celles-ci et la juridiction québécoise est toutefois essentielle. La désignation de la loi québécoise dans la licence constitue le facteur de rattachement privilégié. En l'absence d'une telle clause, la loi québécoise devrait trouver application uniquement si elle présente les liens les plus étroits avec la licence. Ce

sera le cas si la résidence ou l'établissement du donneur de licence est situé sur le territoire du Québec. Cette solution est compréhensible dans la mesure où l'obligation principale, en vertu d'une licence de logiciel libre et ouvert, consiste, de la part du donneur de licence, à délivrer le logiciel au licencié. Dans la mesure où un tel facteur de rattachement existe, la licence libre et ouverte doit d'abord être compatible avec les dispositions fédérales relatives au droit d'auteur, puis satisfaire les exigences du droit civil québécois.

*Constitutionnalité*

[73] Tout d'abord, il ne saurait être question, au Canada, de mettre en doute les licences libres et ouvertes au motif qu'elles méconnaîtraient les principes fondamentaux du droit d'auteur. Contrairement à la Constitution américaine, la Constitution canadienne ne régit pas l'exercice des pouvoirs du gouvernement fédéral relativement au droit d'auteur. De plus, la Loi sur le droit d'auteur ne contient ni préambule, ni article définissant ses objectifs. Son article 3 précise simplement que « le droit d'auteur sur l'œuvre comporte le droit exclusif de produire ou reproduire la totalité ou une partie importante de l'œuvre [...] ». C'est précisément de l'exercice de ce droit que les licences libres et ouvertes tirent leur signification.

*Formalisme*

[74] De plus, contrairement aux législations en vigueur dans la plupart des pays d'Europe continentale, la Loi canadienne sur le droit d'auteur n'encadre absolument pas les techniques permettant d'autoriser des tiers à accomplir des actes qui sont, à prime abord, interdits par le droit d'auteur. Ainsi, il ne saurait être question, au Québec, qu'une licence de logiciel manque de précision quant à la délimitation du domaine d'exploitation des droits concédés. Le donneur de licence est entièrement libre de rédiger le document servant à concéder ses droits de la façon dont il l'entend, au risque que celui-ci soit flou ou incomplet.

*Exigences de signature*

[75] En fait, la seule formalité requise par la Loi sur le droit d'auteur est que cette licence soit rédigée par écrit et signée par le titulaire du droit. Bien que cette règle unique n'impose qu'un fardeau extrêmement léger aux donneurs de licence, elle pose néanmoins certaines difficultés lorsque confrontée aux licences libres et ouvertes. En effet, ces dernières étant le plus souvent jointes au logiciel à l'intérieur d'un fichier électronique, il n'est pas certain qu'elles répondent de façon universelle à l'exigence concernant la signature. À ce propos, il semble maintenant bien établi que l'exigence d'une signature ne nécessite plus l'apposition d'une marque manuscrite. Un premier exemple est fourni par la Loi sur la protection des renseignements personnels et les documents électroniques qui précise que la signature électronique peut parfois équivaloir à une signature prévue par les lois fédérales. Cependant, cette mesure ne s'applique toujours pas à la Loi sur le droit d'auteur. Le droit civil québécois, qui trouve application de façon supplétive en pareilles circonstances, fournit un second exemple. En effet, la Loi concernant le cadre juridique des technologies de l'information établit la neutralité technologique qui prévaut au Québec en spécifiant que la signature peut être apposée au moyen de tout procédé, dans la mesure où les exigences de l'article 2827 du Code civil du Québec sont respectées. Les exigences de cet article sont de deux ordres : la signature doit permettre l'identification de la personne concernée et représenter une manifestation de son consentement. Ainsi, en fonction des dispositions en vigueur au Québec, la forme de la signature apposée sur une licence n'est pas un élément déterminant de sa validité. Celle-ci doit plutôt être établie afin que le moyen retenu remplisse les fonctions inhérentes à la signature.

*Pratique*

[76] Dans le contexte des licences libres et ouvertes, la pratique veut généralement que le nom du titulaire des droits sur le logiciel soit apposé en entête ou en pied de page de la licence, voire à l'intérieur de chacun des fichiers composant le logiciel. Dans la mesure où le donneur de licence y insère une marque qu'il utilise de façon courante, le critère lié à l'identification est alors aisément respecté. Quant à la manifestation du consentement, il est peu plausible qu'un développeur appose son nom à proximité d'une licence qui elle-même est jointe à son logiciel, sans avoir l'intention d'en respecter les termes. Il semble donc que cette façon de procéder, fortement inspirée par le droit américain, permette de

remplir les conditions de forme imposées par la Loi sur le droit d'auteur.

[77] Par ailleurs, les tribunaux devraient faire preuve d'une certaine flexibilité à cet égard puisque la jurisprudence va parfois jusqu'à reconnaître l'existence de licences implicites ou verbales, malgré l'apparente contradiction avec les termes de la loi. Les décisions de ce type sont généralement rendues lorsqu'une pratique ou une coutume allant dans le même sens peut être constatée. Vu leur relative uniformité en ce qui a trait à l'apposition du nom du donneur de licence et leur utilisation maintenant devenue courante dans l'industrie du logiciel, les licences libres et ouvertes devraient pouvoir bénéficier du régime établi par ces précédents. Ceci est d'autant plus vrai que la jurisprudence fait preuve de plus de souplesse en matière de licences non exclusives.

*Capacité du  
donneur de licence*

[78] Dans l'hypothèse où la forme requise est respectée, la qualité du donneur de licence à concéder les droits concernés doit également être évaluée. À ce sujet, le principe directeur est que l'auteur d'un logiciel en est le titulaire des droits. Cependant, deux exceptions peuvent modifier cet état de fait. Tout d'abord, il est possible que l'auteur ait déjà cédé à un tiers la titularité de ces droits. Ensuite, la Loi sur le droit d'auteur stipule que lorsqu'un employé développe un logiciel dans l'exercice de son emploi, c'est son employeur qui est titulaire des droits. Dans un cas, comme dans l'autre, toute licence concédée par une personne autre que celle possédant effectivement le droit d'auteur serait sans effet. Or, en ce qui concerne les logiciels libres et ouverts, les contributions effectuées par des employés peuvent aisément tomber dans cette situation, notamment si elles sont faites à l'insu de l'employeur. La validité d'une licence de logiciel libre et ouvert en fonction de ce critère devra donc être évaluée, en tenant compte du contexte ayant circonscrit le développement du logiciel concerné.

*Échange de  
consentements*

[79] Dans la mesure où les dispositions canadiennes sur le droit d'auteur ne semblent pas poser d'obstacles à la reconnaissance des licences libres et ouvertes au Québec, il faut que celles-ci satisfassent aux exigences du droit civil québécois relatives aux ententes contractuelles. Au Québec, c'est l'article 1385 du Code civil du Québec qui prévoit les modalités de la formation des contrats. Celui-ci précise que le seul échange de consentement entre les parties suffit à les engager l'une à l'autre, pour autant que l'entente dispose d'une cause et d'un objet respectant l'ordre public.

*Consentement  
exprès*

[80] Ainsi, le critère déterminant réside dans la constatation du consentement, et particulièrement celui du licencié dans le cas des licences libres et ouvertes. Or, la jurisprudence québécoise est extrêmement flexible quand il s'agit de reconnaître la manifestation de la volonté des parties. Cette attitude est compréhensible, eu égard à la liberté contractuelle dont elles jouissent généralement. À cet effet, la Cour suprême dans l'affaire *Dell Computer Corp. c. Union des consommateurs* est venue confirmer qu'un contrat conclu en ligne, même lorsque imposé par l'une des parties sous la forme d'un contrat d'adhésion, peut constituer une acceptation valide, si la procédure utilisée est claire et sans équivoque. Ainsi, lorsque le licencié doit nécessairement activer une procédure d'acceptation multimédia avant de pouvoir utiliser ou télécharger le logiciel, la validité de la licence pose peu de difficulté.

*Consentement  
tacite*

[81] Cependant, l'article 1386 du Code civil du Québec précise que la manifestation de la volonté du licencié n'a pas à être expresse, elle peut très bien être tacite. En fait, la forme d'acceptation généralement privilégiée pour les licences libres et ouvertes (le licencié est présumé en avoir accepté les termes, car autrement il agirait dans l'illégalité) se rapproche beaucoup plus d'un consentement tacite. Le licencié se trouve alors dans une situation similaire à l'internaute qui se voit imposer les conditions d'utilisation d'un site Internet par son webmestre. Or, dans l'affaire *Aspencer1.com*, la cour émet l'opinion qu'une clause « posée » sur un site ne suffit pas à engager ses utilisateurs. Cependant dans de telles circonstances, la nature du service offert est généralement considérée par le juge dans l'évaluation de la clause contestée. Il est donc possible de présumer que face à une licence libre et ouverte, l'amplitude des droits concédés au licencié et la gratuité de la mise

à disposition du logiciel feraient pencher la balance en faveur de la validité de la licence.

[82] En principe, rien en droit civil québécois ne s'oppose donc formellement à la manifestation de la volonté du licencié, comme les donateurs de licences libres et ouvertes le préconisent. Ceci est particulièrement vrai pour les professionnels des technologies de l'information qui, aujourd'hui, ne peuvent plus ignorer le mode de fonctionnement de ces licences.

*Cause de la licence* [83] En ce qui a trait à la cause de la licence, il s'agit de la raison qui détermine les parties à la conclure. Ainsi, la possibilité existe qu'une licence libre et ouverte soit déclarée nulle vis-à-vis d'un licencié québécois, si l'une des parties a transigé dans le but de commettre un acte illicite. Ce serait le cas, par exemple, du donneur de licence qui distribue un logiciel auquel est intégré un cheval de Troie ou du licencié qui a recours à un logiciel d'administration réseau afin d'accéder aux systèmes d'un tiers sans y être autorisé.

*Objet de la licence* [84] L'objet, pour sa part, constitue l'opération juridique envisagée par les parties. La concession de droit d'auteur étant prévue spécifiquement par la loi, cet élément de la formation du contrat n'est pas susceptible de mettre en cause la validité des licences libres et ouvertes.

*Contrats aléatoires* [85] Enfin, l'aspect viral des licences libres et ouvertes avec « copyleft » étant invoqué par certains auteurs comme une cause de nullité, l'analyse de la position du droit civil québécois sur cette question prend une certaine importance. À ce propos, l'article 1382 du Code civil du Québec reconnaît l'existence de contrats aléatoires, à savoir des conventions dont l'étendue des obligations est incertaine au moment de leur conclusion. De la même façon, l'article 1374 du Code civil du Québec prévoit spécifiquement que l'objet d'une obligation peut être un bien futur. Le fait, pour le licencié, de s'engager quant aux modifications et aux distributions qu'il pourrait effectuer dans le futur correspond précisément à ce concept d'incertitude. Ses obligations varient alors en proportion des gestes posés. La jurisprudence québécoise confirme d'ailleurs que la cession du droit d'auteur d'une œuvre future, c'est-à-dire qui n'existe pas encore, prend effet dès que l'œuvre est créée. Bien que la solution soit probablement différente dans les provinces de *common law*, le mécanisme du « copyleft » serait donc parfaitement compatible avec le droit civil québécois.

[86] En définitive, les licences libres et ouvertes semblent parfaitement valides lorsqu'elles sont confrontées au droit applicable au Québec. En fait, la seule réserve importante a trait à l'exigence édictée par la Loi sur le droit d'auteur quant à la formalité de la signature. Malgré tout, l'interprétation susceptible de prévaloir à ce sujet laisse penser que leur validité sera retenue dans l'hypothèse où cette question ferait l'objet d'un litige devant les tribunaux québécois.

*Références* Dell Computer Corp. c. Union des consommateurs, 2007 CSC 34, [2007] 2 R.C.S. 801, Web : <<http://www.canlii.org/fr/ca/csc/doc/2007/2007csc34/2007csc34.html>>

CA Paris, 16 septembre 2009, RG n 04/24298 : SA EDU 4 c. Association AFPA

CEA, CNRS, INRIA, « CeCILL et les logiciels libres », CeCILL, 2011, source : <<http://www.cecill.info>>

CLÉMENT-FONTAINE (M.), « La licence publique générale GNU », Juriscom, 1999, Web : <<http://www.juriscom.net/uni/mem/08/presentation.htm>>

*Code civil du Québec*, L.Q. 1991, c. 64, art. 1385, 1386, 1410, 1411, 3111, 3113, Web : <<http://www.canlii.org/fr/qc/legis/lois/lq-1991-c-64/derniere/lq-1991-c-64.html>>

*Code de la propriété intellectuelle* (France), art. L. 131-3

Diffusion YFB Inc. c. Les Disques Gamma (Québec) Ltée, (1999) C.S. 500-05-047570-997

Jacobsen v. Katzer, 535 F. 3d 1373 (2008),  
Web : <[http://scholar.google.ca/scholar\\_case?case=17776182574171214893&hl=en&as\\_sdt=2,5&as\\_vis=1](http://scholar.google.ca/scholar_case?case=17776182574171214893&hl=en&as_sdt=2,5&as_vis=1)>

*Loi concernant le cadre juridique des technologies de l'information*, L.R.Q., c. C-1.1, art. 39, source : <<http://www.canlii.org/fr/qc/legis/lois/lrq-c-c-1.1/derniere/lrq-c-c-1.1.html>>

*Loi sur la protection des renseignements personnels et les documents électroniques*, L.R.C. (2000), c. 5, art. 43, source : <<http://www.canlii.org/fr/ca/legis/lois/lc-2000-c-5/derniere/lc-2000-c-5.html>>

*Loi sur le droit d'auteur*, L.R.C. (1985), c. C-42, art. 13 (3), 13(4), 13.1,  
Web : <<http://www.canlii.org/fr/ca/legis/lois/lrc-1985-c-c-42/derniere/lrc-1985-c-c-42.html>>

MARKELLOU (M. P.), « EUPL : vers un dialogue harmonieux entre « copyright » et « copyleft »? », *Revue Lamy droit de l'immatériel*, n° 28, 2007

NADAN (C. H.), " Open Source Licensing Virus or Virtue " (2002) 10 *Tex. Intell. Prop. L.J.*349

Netfilter/iptables v. Sitecom Germany GmbH, IFROSS, 2004,  
Web : <[http://www.ifross.de/ifross\\_html/eVWelte.pdf](http://www.ifross.de/ifross_html/eVWelte.pdf)>

Progress Software Corp. v. MySQL AB, (2002) 195 F.Supp.2d 328,  
Web : <[http://scholar.google.ca/scholar\\_case?case=13584730711160488510&q=Progress+Software+Corp.+c.+MySQL+AB,+195+F.Supp.2d+328&hl=en&as\\_sdt=2,5&as\\_vis=1](http://scholar.google.ca/scholar_case?case=13584730711160488510&q=Progress+Software+Corp.+c.+MySQL+AB,+195+F.Supp.2d+328&hl=en&as_sdt=2,5&as_vis=1)>

WIKIPEDIA, " SCO v. IBM ", 2010, Web : <[http://en.wikipedia.org/wiki/SCO\\_v.\\_IBM](http://en.wikipedia.org/wiki/SCO_v._IBM)>

## 2.2. Établissement de la chaîne de titres

*Enjeux*

[87] Les logiciels libres et ouverts, en tant qu'œuvres de l'esprit, sont avant tout protégés par les règles du droit de la propriété intellectuelle. Ces protections, généralement bien adaptées au contexte du développement de logiciels propriétaires en milieu fermé, entraînent des difficultés lorsqu'elles sont confrontées à l'approche collaborative retenue par de nombreux développeurs de logiciels libres et ouverts. Ainsi, la multiplication de concessions de droits successives requiert une étude approfondie de la chaîne de titres dans laquelle l'utilisateur est le dernier maillon.

*Structure de la chaîne de titres*

[88] Traditionnellement, les logiciels résultent des efforts d'un groupe restreint de programmeurs travaillant ensemble à l'intérieur d'une organisation hiérarchisée. À l'inverse, l'évolution des logiciels libres et ouverts est généralement décrite comme reposant sur la contribution volontaire d'une partie de leurs utilisateurs. Ceux-ci, tout en étant dispersés sur la surface du globe, s'y impliquent en fonction de leur expertise et de leurs besoins respectifs. Ce modèle du bazar ne colle cependant pas exactement à la réalité. Tous les projets de logiciels libres et ouverts possèdent un minimum d'organisation. Le plus souvent, la coordination est assurée par le promoteur du projet, qui sert de guide à l'évolution du code informatique. D'ailleurs, l'existence d'une structure est essentielle en cas d'éventuels conflits provoqués par la proposition de solutions contradictoires. Souvent, c'est donc un petit nombre de promoteurs qui contribuent à l'élaboration de la

plus grande partie du code des logiciels libres et ouverts, les nombreux développeurs occasionnels ne fournissant qu'une aide complémentaire.

[89] Aussi, l'idée souvent véhiculée que la structure contractuelle s'établissant entre les divers contributeurs peut être représentée par une chaîne de contrats linéaires semble mal adaptée à la réalité. En pratique, le développement de la quasi-totalité des logiciels libres et ouverts peut être schématisé sous la forme d'une étoile dont le promoteur du projet occupe le centre et les divers contributeurs ses extrémités.

*Œuvres créées en collaboration*

[90] Peu importe la structure contractuelle retenue, une fois rassemblées, les contributions sont intégrées les unes aux autres pour former un logiciel autonome. À partir de ce point, il devient difficile de distinguer la part créée par un contributeur de celles créées par les autres. Or, la Loi sur le droit d'auteur attache à ce genre de réalisation le statut d'œuvres créées en collaboration. Cette même loi étant muette sur les effets de ce statut, c'est vers le droit civil québécois qu'il faut se tourner. En vertu du Code civil du Québec, la plupart des logiciels libres et ouverts devraient être soumis à la copropriété indivise, comme elle est définie à l'article 1010. La principale conséquence de l'indivision est l'existence d'un seul droit d'auteur global sur l'ensemble du logiciel. Il en découle également une présomption d'égalité entre les parts des contributeurs et l'autorisation, pour chacun d'eux, d'accomplir des actes concernant le logiciel, à condition de ne porter atteinte, ni à sa destination, ni aux droits des autres développeurs. Ainsi, dans l'affaire Tremblay c. Nguyen, la Cour supérieure du Québec a confirmé la possibilité, pour un titulaire indivis de droit d'auteur, de s'adresser seul aux tribunaux en cas de violation de ce droit. Dans le même ordre d'idées, toute licence concédée par l'un des titulaires indivis devrait être valide dès que le logiciel était destiné à être distribué sous une licence libre et ouverte. Évidemment, cette indivision n'existe pas lorsque, comme dans le cas des projets gérés par la FSF, toutes les parts sont transférées entre les mains de la même personne.

*Défaut de la chaîne de titres*

[91] Un problème majeur est susceptible de survenir si l'un des maillons de la chaîne de titres fait défaut, c'est-à-dire que la concession de droit octroyée n'est pas effective. Dans un tel cas, tous les licenciés subséquents se trouvent dans une situation irrégulière, car les gestes qu'ils ont posés à l'égard du logiciel ont contrevenu au droit d'auteur des titulaires précédant le maillon défectueux. Deux situations peuvent mener à ce résultat. La première est l'invalidité de l'une des licences faisant partie de la chaîne de titres. La seconde est l'introduction illicite, dans le logiciel, de codes protégés par les droits d'un tiers.

*Invalidité d'une licence*

[92] L'invalidité d'une licence libre et ouverte peut résulter de plusieurs motifs. L'exemple le plus évident est peut-être la contribution effectuée par un employé, sans le consentement de son employeur. De la même façon, un juge québécois pourrait toujours refuser de reconnaître une telle licence en prétextant l'absence de signature. Dans tous les cas, le caractère international que prend le développement des logiciels libres et ouverts fait en sorte que chaque maillon de la chaîne de titres est potentiellement invalide en fonction d'un droit national différent.

*Introduction de codes protégés*

[93] L'introduction illicite de codes protégés par les droits d'un tiers est précisément le fondement du recours qu'avait intenté SCO face à IBM. SCO alléguait qu'IBM aurait introduit dans le noyau de Linux des éléments d'UNIX protégés par le droit d'auteur d'un tiers, justifiant ainsi ses réclamations à l'égard des milliers d'utilisateurs de ce système d'exploitation. Bien que SCO ait échoué à faire valoir ses prétentions, la voie de recours utilisée demeure toujours valable et ce genre de litige pourrait se reproduire dans le futur.

*Compilations*

[94] Finalement, les distributions qui amalgament des logiciels développés indépendamment et les regroupent ensemble sont susceptibles de complexifier encore un peu la chaîne de titres des logiciels libres et ouverts. C'est le cas, par exemple, des nombreuses distributions de Linux. Leur particularité réside dans le fait qu'en plus de la protection individuelle dont bénéficie chacun des logiciels qui y sont inclus, la compilation est elle-même protégée par le droit d'auteur. Par ailleurs, les distributions de logiciels libres et ouverts sont généralement elles-mêmes soumises à une licence autorisant leur

utilisation, leur modification et leur redistribution. L'exercice de ces droits implique cependant de porter une attention particulière au respect de la licence de la compilation, en plus des licences relatives à chacun des logiciels inclus.

Évaluation du  
risque

[95] Dans l'ensemble, la présence d'un défaut dans la chaîne de titres constitue sans aucun doute le plus important risque juridique pour les utilisateurs de logiciels libres et ouverts. En effet, dès qu'un maillon de la chaîne devant les relier à chacun des développeurs est manquant, ceux-ci se trouvent à commettre une infraction au droit d'auteur. Malheureusement, aucune technique ne permet d'établir aisément la fiabilité de cette chaîne de titres et les utilisateurs ne découvriront généralement l'infraction que s'ils font l'objet d'une poursuite. Il est vrai que même s'il existe un défaut, la vaste majorité des titulaires de droits n'ont aucun intérêt à l'invoquer pour, ainsi, paralyser le développement d'un logiciel dont ils ont eux-mêmes besoin. À cet égard, le mode de développement des logiciels libres et ouverts repose donc en grande partie sur la bonne foi des développeurs impliqués.

[96] Compte tenu de la pratique, ce risque doit tout de même être relativisé puisque le litige soulevé par SCO est le seul exemple où la chaîne de titres de l'un d'eux a été mise en doute. De plus, les mêmes questions ont toujours existé en matière de logiciels propriétaires, même si elle se pose à une échelle beaucoup plus limitée, étant donné leur mode de développement plus fermé. Les logiciels libres et ouverts, loin de reposer sur de nouveaux mécanismes, ne font que pousser à leur extrême limite les règles encadrant la concession successive des droits de propriété intellectuelle.

Références

CARON (C.), « Les licences de logiciels dits libres à l'épreuve du droit d'auteur français », (2003) 23 Dalloz 1556

CARRIÈRE (L.), « Protection du logiciel et autres œuvres originales en vertu de la Loi sur le droit d'auteur et conventions usuelles en ce domaine » (1996) Léger Robic Richard, Web : <<http://www.robic.ca/OPublications.aspx>>

COOL (Y.), DE PATOUL (F.), DE ROY (D.), HAOUIDEG (H.), LAURENT (E.) et MONTERO (E.), *Les logiciels libres face au droit*, collection Cahiers du Centre de Recherches Informatiques et Droit, numéro 25, Éditions Bruylant, Bruxelles, 2005

*Loi sur le droit d'auteur*, L.R.C. (1985), c. C-42, art. 2, Web : <<http://www.canlii.org/ca/loi/c-42/>>

MEEKER (H. J.), *The Open Source Alternative: Understanding Risks and Leveraging Opportunities*, Hoboken, Wiley, 2008

ROJINSKY (C.) et GRYNBAUM (V.), « Les licences libres et le droit français », (2002) *Propriété intellectuelle*, juillet 2002/4

Tremblay c. Nguyen, (1977) 24 C.P.R. 289

### 2.3. Brevets logiciels

Brevets logiciels

[97] Un autre risque juridique découlant de l'application des règles de propriété intellectuelle aux logiciels libres et ouverts concerne le recours aux brevets logiciels. L'objectif fondamental du droit des brevets est de protéger les inventions utiles, originales et non évidentes. Tout comme pour le droit d'auteur, il ne s'agit donc pas d'octroyer un droit sur une idée, mais bien de protéger sa mise en œuvre physique, c'est-à-dire la fabrication d'un objet la matérialisant. Ainsi, l'invention est brevetable alors que la découverte ne l'est pas. Au Canada, la loi définit une invention comme une réalisation, un procédé ou une machine

présentant un caractère nouveau ou utile.

*Principe de la non-brevetabilité*

[98] Bien que les logiciels répondent le plus souvent à ces critères, l'article 27 (8) de la même loi ajoute une exception lorsque l'invention repose sur de simples principes scientifiques ou des conceptions théoriques. Le *Recueil des pratiques* du Bureau des brevets formule l'interprétation donnée à cette disposition par les autorités fédérales chargées de l'attribution des brevets au Canada. Il stipule qu'un programme d'ordinateur, lorsque revendiqué en soi, constitue un plan théorique ou un ensemble de règles d'exploitation d'un ordinateur. Ainsi, les logiciels sont en principe exclus du champ des inventions brevetables au Canada. Cette interprétation découle de la décision de la Cour d'appel fédérale dans l'arrêt *Schlumberger Canada Ltd. c. Commissaire des brevets*. La cour avait alors déterminé qu'il n'y avait rien de nouveau dans le fait d'utiliser un ordinateur pour faire des calculs et qu'une formule mathématique n'était qu'un simple principe scientifique ou une conception théorique. Elle ajouta que le même calcul, s'il était réalisé par un être humain, aurait constitué une séquence d'opérations mentales et intellectuelles, non brevetable au Canada.

*Intégration à un élément brevetable*

[99] Il n'en demeure pas moins que le Bureau des brevets reconnaît la brevetabilité d'un logiciel, lorsque la revendication concerne un produit matériel étant lui-même limité par le programme d'ordinateur. Ainsi, lorsqu'un logiciel est intégré à un élément traditionnellement brevetable, comme une imprimante, un système de télécommunication ou un système d'ascenseur, celui-ci bénéficie de la protection au même titre que la portion matérielle de l'invention. Dans certains cas, de « purs » logiciels ont même été brevetés sur le seul argument que l'on pouvait les rattacher à au moins une composante physique. Cette jurisprudence semble donc indiquer une tendance assez libérale de la Commission d'appel du Bureau des brevets en matière de brevet logiciel.

*Facteurs de risque*

[100] Toutefois, en plus des difficultés qu'ils impliquent pour l'ensemble de l'industrie du logiciel, les brevets constituent un risque juridique majeur pour l'avenir des logiciels libres et ouverts en particulier. Plusieurs facteurs permettent d'expliquer cette menace :

- ✦ Les brevets permettent de protéger des techniques anodines, mais qui sont parfois nécessaires aux développeurs de logiciels libres et ouverts;
- ✦ Les possibilités de réécrire un algorithme, sous une forme non protégée, diminuent en proportion de l'augmentation du nombre des brevets logiciels;
- ✦ Avec l'accroissement du nombre de brevets émis, il devient difficile de ne pas utiliser l'un d'eux involontairement;
- ✦ Les coûts d'enregistrement des brevets, ainsi que ceux de leur application, avantagent les grands éditeurs d'un arsenal anticoncurrentiel dont ne peuvent disposer plusieurs développeurs de logiciels libres et ouverts;
- ✦ La brevetabilité des formats de fichiers ou des protocoles de communication permet de monnayer l'interopérabilité;
- ✦ La preuve qu'un brevet a été enfreint est plus facile à faire à l'encontre des développeurs de logiciels libres et ouverts, puisque le code de ces logiciels est mis à disposition et peut être examiné.

*International*

[101] Considérant le caractère international du modèle de développement collaboratif qui caractérise les logiciels libres et ouverts, ces risques prennent un caractère concret dès lors que les brevets logiciels sont reconnus dans certains pays. C'est d'ailleurs le cas aux États-Unis et au Japon où des milliers de brevets sont enregistrés chaque année sur de purs logiciels. En Europe, un débat faisant rage à cet effet a pris fin en 2005, lorsqu'une directive de la Commission européenne destinée à introduire le brevet logiciel dans les pays membres de l'Union a été rejetée.

*Engagements volontaires* [102] Au cours des dernières années, quelques stratégies ont été mises en œuvre par les développeurs de logiciels libres afin de limiter le risque que représentent les brevets logiciels pour leurs projets respectifs. Ainsi, certaines multinationales se sont engagées volontairement à ne pas faire usage de leurs brevets à l'encontre des communautés dont elles font partie. Le Patent Commons Project vise d'ailleurs à recenser l'ensemble de ces engagements en les regroupant au sein d'un répertoire unique.

*Protections contractuelles* [103] De plus, certains organismes voués à la promotion du logiciel libre et ouvert ont réagi en adaptant leurs licences de façon à y intégrer des mécanismes de protection contre les revendications fondées sur les brevets. C'est entre autres le cas de la licence Apache 2.0 qui prévoit que les licenciés s'octroient mutuellement une licence sur tout brevet avec lequel le logiciel est susceptible d'entrer en infraction. Dans l'hypothèse où un licencié entamerait des poursuites pour non-respect de l'un de ses brevets, cette licence tombe et expose le licencié à des poursuites similaires de la part des autres contributeurs. La GPL 3.0 met en place une protection similaire. Cependant, bien que ces clauses règlent relativement efficacement la question au sein de la communauté concernée, elles offrent peu de protection contre les brevets appartenant à des tiers.

*Évaluation du risque* [104] Ici encore, le risque juridique théorique que représentent les brevets logiciels pour l'avenir du logiciel libre doit être relativisé en considération de sa mise en application pratique. L'image négative que projeterait une entreprise poursuivant les promoteurs d'un logiciel libre et ouvert aux seules fins d'obtenir une compensation monétaire semble suffire à dissuader la plupart d'entre elles d'agir de la sorte. Compte tenu du faible nombre de poursuites de ce type ayant pris forme jusqu'ici, que ce soit aux États-Unis ou au Japon, il est raisonnable de présumer que cette situation est peu susceptible de survenir.

*Références* GRAVELLE (L.-P.), « De la vie aux ordinateurs – développements récents en matière de brevetabilité des organismes vivants, des méthodes de traitement et des technologies informatiques », (2000) 138 *Développements récents en propriété intellectuelle* 104, Web : <<http://www.robic.ca/publications/Pdf/254-LPG.pdf>>

LÉVÊQUE (F.), MÉNIÈRE (Y.), "Copyright Versus Patents : The Open Source Software Legal Battle", (2007) 4 *Review of Economic Research on Copyright Issues* 21

LINUX FOUNDATION, « Patent Commons Project », 2011, Web : <<http://www.patent-commons.org>>

*Loi sur les brevets*, L.R.C. (1985), c. P-4, art. 2, source : <<http://www.canlii.org/fr/ca/legis/lois/lrc-1985-c-p-4/derniere/lrc-1985-c-p-4.html>>

OFFICE DE LA PROPRIÉTÉ INTELLECTUELLE DU CANADA, « Recueil des pratiques du Bureau des brevets », (2010) *OPIC*, Web : <[http://www.cipo.ic.gc.ca/eic/site/cipointernet-internetopic.nsf/fra/h\\_wr00720.html](http://www.cipo.ic.gc.ca/eic/site/cipointernet-internetopic.nsf/fra/h_wr00720.html)>

Schlumberger Canada Ltd. c. Commissaire des brevets, (1981) 56 C.P.R. (2d) 204 (C.A.F.)

#### 2.4. Qualification du contrat de licence

*Enjeux* [105] Lorsque l'Administration accepte les termes d'une licence libre et ouverte, une relation contractuelle s'établit avec le ou les titulaires du droit d'auteur sur le logiciel. La licence constitue la principale base de recours dont dispose l'Administration en cas de défaut du logiciel ou de manquement du prestataire. À l'inverse, l'Administration qui distribue elle-même du logiciel libre et ouvert déterminera ses propres obligations. Pour cette raison, l'étude de cette relation contractuelle est en grande partie orientée vers l'établissement des critères qui déterminent la responsabilité des donneurs de licences

libres et ouvertes. Pour engager cette responsabilité, il est nécessaire de démontrer que le donneur de licence n'a pas exécuté une obligation mise à sa charge au titre du contrat de licences ou au titre des autres garanties prévues par la loi. Les licences libres et ouvertes étant spécialement conçues pour limiter la responsabilité civile du donneur de licence, le fardeau de cette preuve peut s'avérer difficile à surmonter.

*Qualification du  
contrat de licence  
libre et ouvert*

[106] À ce titre, la question de la qualification juridique des licences libres et ouvertes, c'est-à-dire leur rattachement au régime de l'un des contrats nommés, conditionne l'application des règles légales relatives à l'opération juridique considérée et peut ainsi influencer sur les droits et les obligations des parties. Malheureusement, même lorsqu'il s'agit de logiciels propriétaires, la question de la qualification juridique des contrats de licences de logiciels ne fait pas l'objet d'une solution ferme et unanime.

*Contrat de vente?*

[107] Une partie de la doctrine privilégie la qualification juridique de contrat de vente pour les logiciels fournis avec du matériel, par application de la théorie de l'accessoire. C'est en ce sens que s'est prononcée, en 1985, la Cour supérieure du Québec dans l'affaire *Olier, Grisé & Cie Ltée c. Équipements de bureau Maskoutan Inc.*, en qualifiant une licence de logiciel de vente pure et simple, considérant que le logiciel faisait partie intégrante du matériel fourni. De manière analogue, plusieurs arrêts de la Cour d'appel de Paris ont fait application du régime de la vente à un ensemble contractuel, en raison de la dépendance du contrat de mise à disposition de logiciels (licences) avec le contrat de fourniture de matériels qui en est le support. Il s'agissait en l'espèce d'actions pour vices cachés fondées sur la garantie des défauts de la chose vendue.

[108] Dans le même sens, en matière de progiciels, c'est le régime juridique de la vente qui a la préférence de certains auteurs. Ces derniers adoptent une approche « matérialiste » et envisagent le régime de la licence par l'intermédiaire de la vente d'exemplaires du logiciel. Par conséquent, de manière générale, le régime de la vente pourrait être appliqué aux contrats (ou groupes de contrats) qui prévoient la fourniture de systèmes informatiques comprenant du matériel, des progiciels, voire des développements spécifiques, à la condition que le matériel et les progiciels constituent une part prépondérante du système fourni. Cependant, cette qualification de vente n'est pas satisfaisante au regard du droit de la propriété intellectuelle : en effet, la plupart des donneurs de licences de progiciels se réservant leurs droits, ces contrats ne transfèrent aucun droit immatériel.

*Contrat de location?*

[109] C'est en raison de cette absence de transfert de droit sur le logiciel que la quasi-unanimité des auteurs considère que le contrat de licence s'apparente davantage au contrat de location. Cette qualification semble pertinente eu égard à l'« économie » du contrat de licence : le donneur de licence concède au licencié la jouissance d'une chose incorporelle (le logiciel), en contrepartie du paiement d'un loyer (redevance), pendant une durée limitée. En outre, la notion de location permettant au donneur de licence de limiter les actes que le licencié est autorisé à accomplir, ce dernier ne dispose pas librement du logiciel. Elle permet également au donneur de licence de recevoir, pendant le temps d'utilisation du logiciel, une rémunération régulière prenant la forme d'une redevance. La qualification de louage apparaît ainsi la plus adaptée au contrat de licence, même si la doctrine ne manque pas de souligner quelques réserves. Il est vrai que l'analogie n'est pas parfaite, car le donneur de licence, à la différence du loueur, ne se dépossède pas du bien qu'il met à disposition, d'autant plus qu'il conserve le droit de le « louer » à d'autres utilisateurs.

*Contrat innommé?*

[110] C'est certainement pour cette raison que les tribunaux québécois préfèrent généralement considérer qu'une licence de logiciels n'a ni la nature d'une vente, ni la nature d'un louage, mais celle d'un contrat innommé conférant un droit personnel d'usage. Cette solution ressort clairement de l'affaire *Informatique L.G.A. Inc. c. Compagnie d'arrimage de Québec Ltée* dans laquelle le juge souligne que la licence ne transfère aucun droit réel, mais confère un simple droit personnel d'utilisation d'une chose incorporelle. La conséquence de cette solution est très importante en pratique : puisque ni

le régime de la vente ni celui du louage ne s'appliquent à la licence, les garanties légales prévues par ces régimes sont automatiquement exclues.

[111] Cette jurisprudence se trouve cependant nuancée depuis 1994 par la décision *Unicel Inc. c. Contalitec Informatique Inc.* À cette occasion, la Cour du Québec confirma son analyse de la licence en contrat innommé, en précisant, toutefois, qu'il présente certaines caractéristiques du contrat de louage (c.-à-d. contrat de location). Les juges ont alors estimé que le donneur de licence devait être soumis aux obligations qu'impose le Code civil du Québec au loueur, c'est-à-dire les garanties légales associées au contrat de location.

*Logiciel spécifique* [112] S'agissant du développement et de la fourniture d'un logiciel spécifique, la jurisprudence québécoise tend à maintenir la qualification de contrat innommé, à la différence du droit français qui y voit un contrat d'entreprise. En effet, selon la doctrine française, il faut distinguer le cas des logiciels dits spécifiques, ce qui pourrait être le cas d'un logiciel libre et ouvert développé spécifiquement pour le licencié qui le commanderait.

*Logiciel gratuit* [113] Enfin, la qualification juridique est encore différente lorsque le logiciel est distribué gratuitement. Dans ce cas, le licencié n'étant tenu à aucune redevance, l'application du droit de la vente ou du louage est exclue. En droit civil québécois, il est alors possible de considérer ce type de contrat comme un prêt à usage. L'obligation de restitution, qui est à la base du contrat de prêt, perd alors de son importance puisque le logiciel est un bien immatériel pouvant être reproduit facilement.

*Ambiguïté* [114] En définitive, l'ensemble de ces opinions doctrinales et de cette jurisprudence non uniforme et variable montrent à quel point le contrat de licence de logiciel est difficilement identifiable à l'un des contrats nommés prévus par le Code civil du Québec. Pourtant, cette opération de qualification juridique n'est pas sans importance. En effet, la liberté contractuelle en l'espèce est doublement limitée : d'une part, la commune intention des parties ne doit pas permettre de faire obstacle aux règles d'ordre public prévues par un régime contractuel donné, d'autre part, il arrive souvent que, pour déterminer cette commune intention, la qualification du contrat permette au juge de disposer des règles applicables, en cas d'ambiguïté, de lacune ou de « zones d'ombre » de la licence.

[115] Appliquée aux logiciels libres et ouverts, la qualification juridique du contrat de licence de logiciel reste donc variable selon les circonstances propres à chaque espèce. Pourtant cette qualification est susceptible d'influer de manière déterminante sur les droits et les obligations des parties et de décider du régime de garanties légalement applicable.

*Références* *Code civil du Québec*, L.Q. 1991, c. 64, art. 1851, Web : <http://www.canlii.org/fr/qc/legis/lois/lq-1991-c-64/derniere/lq-1991-c-64.html>

*Informatique L.G.A. Inc. c. Compagnie d'arrimage de Québec Ltée*, (1991) R.J.Q. 1767 (C.Q.)

*Olier, Grisé & Cie Ltée c. Équipements de bureau Maskoutan Inc.*, (1985) C.S. 680

*Unicel Inc. c. Contalitec Informatique Inc.*, (1994) J.E. 94-1910 (C.Q.)

## 2.5. Obligations des parties

*Enjeux* [116] L'objet des licences de logiciels est de définir dans quelles conditions et dans quelles limites le licencié peut utiliser le logiciel en question. C'est pourquoi elles contiennent généralement plus d'obligations expressément mises à la charge du licencié que d'obligations pour le donneur de licence. Les licences de logiciels libres et ouverts ne font pas exception à cette pratique. Elles délimitent même précisément les conditions dans

lesquelles le licencié peut exercer les libertés qui lui sont accordées, tout en omettant de mentionner une quelconque obligation pour le donneur de licence. Par conséquent, lorsque l'Administration est en mesure de négocier de gré à gré une licence libre et ouverte avec un prestataire, il est à son avantage de rééquilibrer le contrat, dans le sens d'un accroissement des obligations que la licence mettrait expressément à la charge du titulaire des droits. Si, comme c'est plus fréquemment le cas, aucune négociation n'est possible, il n'en demeure pas moins que le donneur de licence doit se conformer à certaines obligations légales.

*Obligations  
spécifiques du  
licencié*

[117] Les obligations spécifiques du licencié varient en fonction des licences libres et ouvertes considérées. En général, elles prennent la forme particulière de « droits avec charge », c'est-à-dire d'autorisation à accomplir un acte particulier sous réserve du respect de certaines conditions ou obligations. Ainsi, les obligations mises à la charge du licencié sont le plus souvent des « obligations de faire » (obligations positives). Par exemple, la GPL oblige le licencié à redistribuer le logiciel sous les termes de la même licence. Mais les contrats comportent aussi des obligations expresses « de ne pas faire » (obligations négatives). Par exemple, la licence Apache 2.0 proscrit spécifiquement l'utilisation de la marque Apache.

[118] Enfin, à ces obligations particulières s'ajoute bien entendu l'obligation de payer le prix de la redevance, si la licence est consentie à titre onéreux.

[119] L'évaluation du respect de ces obligations est avant tout une question de fait dont le résultat dépend des agissements des parties. De plus, la seule sanction contractuelle généralement prévue, en cas de non-respect des termes de la licence, est la fin de la concession établie par celle-ci. Dans pareille situation, le droit de la propriété intellectuelle se trouve donc à prendre le relais du droit civil québécois.

*Obligations légales  
du donneur de  
licence*

[120] Dans le cas du donneur de licence, c'est avant tout la loi et la jurisprudence, plutôt que les termes de la licence, qui mettent à sa charge des obligations. De façon générale, celles-ci peuvent être décrites comme une obligation de délivrance du logiciel libre et ouvert, qui elle-même s'accompagne d'une obligation d'information et d'une obligation de garantie.

*Obligations de  
délivrance*

[121] L'obligation première d'un prestataire est la délivrance de l'objet du contrat. Dans le cas des licences libres et ouvertes, cela signifie que le donneur de licence est tenu de mettre le logiciel à la disposition du licencié. Cette délivrance peut prendre plusieurs formes : remise d'un support physique ou remise en ligne par téléchargement.

*Délivrance du code  
source*

[122] La principale question relative à l'obligation de délivrance découle de la mise à disposition du code source du logiciel libre et ouvert. En principe, le licencié qui accepte les termes d'une licence libre et ouverte peut légitimement s'attendre à obtenir la communication de ce code. Le code binaire perd, lui, de son importance. Il n'est pas essentiel puisqu'il ne permet pas de modifier le logiciel, bien qu'en tout état de cause il puisse toujours être obtenu par compilation. En définitive, il est possible de considérer que l'objet des contrats de licences libres et ouvertes soit précisément la délivrance du logiciel sous la forme de code source. En ce sens, l'obligation de délivrance demeurerait inexécutée tant que celui-ci n'est pas mis à disposition.

*Délivrance de la  
documentation*

[123] L'obligation de délivrance peut également être interprétée de façon à inclure la documentation du logiciel à titre d'accessoire. C'est la solution qu'ont retenue les tribunaux français en matière de licences propriétaires. Il est loin d'être certain que ce principe puisse être appliqué aux licences libres et ouvertes, puisqu'il s'oppose à l'esprit de ces conventions. En effet, la documentation et le programme informatique y sont perçus comme des éléments autonomes.

*Obligation  
d'information*

[124] La seconde obligation essentielle à la charge du donneur de licence vise l'information qu'il doit fournir au licencié. En effet, le droit civil fait parfois reposer une obligation minimale de renseignement sur les épaules du cocontractant qui détient des informations dont la connaissance peut être utile à l'autre partie. Dans le cadre des contrats informatiques, la

jurisprudence et la doctrine estiment généralement qu'une assez lourde obligation de renseignement, d'information et de conseil pèse sur le prestataire. En matière de logiciels libres et ouverts, les renseignements que le donneur de licence doit donner au licencié sont essentiellement relatifs aux conditions et aux risques liés à l'utilisation du logiciel (le matériel requis au bon fonctionnement du logiciel ou les principaux bogues). De même, tant que la première version officielle d'un logiciel n'est pas opérationnelle, il paraît important que le donneur de licence mentionne clairement qu'il s'agit d'une version préliminaire (ou bêta). En cas de manquement à cette obligation de renseignement, le donneur de licence pourrait être condamné à réparer les dommages qu'il a causés.

*Obligation de garantie*

[125] Le donneur de licence peut également être tenu de garantir le logiciel qu'il distribue. Ces garanties peuvent être expresses, ce qui est très rare dans le cas des licences libres et ouvertes. En revanche, il peut toujours conserver à sa charge des obligations implicites découlant de garanties d'origine légale.

*Garanties expresses*

[126] Lorsque les développeurs font la promotion des caractéristiques et des fonctions de leur logiciel, il est possible que les tribunaux considèrent que leurs déclarations constituent des obligations de garantie (garanties expresses). Par exemple, ce pourrait être le cas s'il prétend que le logiciel est compatible avec un standard reconnu. Pour bénéficier de ces garanties, le licencié devra prouver que les déclarations en question sont entrées dans le champ contractuel et ont, par conséquent, une force contraignante pour le cocontractant. Compte tenu des circonstances, le fardeau de cette preuve peut s'avérer difficile à surmonter.

*Garantie contre l'éviction*

[127] Parmi toutes les garanties implicites, d'origine légale, trois d'entre elles sont susceptibles d'être appliquées aux contrats de licences libres et ouvertes. La garantie contre l'éviction est la première. Elle a la valeur de principe général et trouve son application dans le cadre du contrat de vente et du contrat de location. Concrètement, elle assure au licencié que son utilisation du logiciel ne sera pas perturbée, ni en fait ni en droit, par son cocontractant ou par des tiers, et garantit au licencié que le donneur de licence n'aurait pas accordé un droit déjà cédé. Elle empêche donc ce dernier de revenir sur l'objet de la licence ou de priver le licencié de l'utilisation du logiciel, tant que celui-ci n'a pas excédé les droits qu'il tenait de lui. Le risque que le licencié d'un logiciel libre et ouvert se retrouve en violation des droits de propriété intellectuelle d'un tiers se trouve donc atténué par la possibilité de réclamer des dommages et des intérêts ou d'appeler en garantie le donneur de licence.

*Garantie contre les vices cachés*

[128] La garantie contre les vices cachés est la seconde. Elle s'applique essentiellement aux contrats de vente et de louage. Elle peut aussi s'appliquer dans le cadre des contrats de prêt, mais uniquement lorsque le prêteur connaissait le vice affectant la chose prêtée. Cette garantie protège le licencié lorsqu'un défaut majeur affecte le logiciel et le rend impropre à l'usage auquel il est destiné. Ce vice ne devant pas être apparent, l'examen du logiciel devra avoir été raisonnable selon les circonstances, ce qui peut être apprécié à l'aide de facteurs comme la compétence technique du licencié, la nature du logiciel, ainsi que les affirmations, éventuellement trompeuses, qui auraient pu être faites avant la transaction elle-même.

*Garantie de conformité*

[129] La garantie de conformité est la troisième. Celle-ci oblige le donneur de licence à fournir un logiciel libre et ouvert conforme aux spécifications, aux besoins et aux objectifs particuliers du licencié, à condition que ces éléments soient entrés dans le champ contractuel. Compte tenu de cette dernière condition, le recours à la garantie de conformité est assez limité pour les logiciels libres et ouverts, dès lors que la licence possède les caractéristiques d'un contrat d'adhésion. En revanche, son application est envisageable lorsque la licence est négociée ou lorsque la fourniture du logiciel libre et ouvert fait partie d'un ensemble contractuel plus englobant.

Références *Code civil du Québec*, L.Q. 1991, c. 64, art. 1723, 1726, 1729, 1730, 1854, 1858, Web : <<http://www.canlii.org/fr/qc/legis/lois/lq-1991-c-64/derniere/lq-1991-c-64.html>>

LE TOURNEAU (P.), *Théorie et pratique des contrats informatiques*, Éditions Dalloz, Paris, 2000

LUCAS DE LEYSSAC, (C.), « L'obligation de renseignements dans les contrats », Coll. l'information en droit privé, LGDJ, Paris, 1978, p. 305

McJOHN (S. M.), "The Paradoxes of Free Software", (2000) 9 *Geo. Mason L. Rev.* 25

## 2.6. Clauses d'exclusion de responsabilité

- Enjeux* [130] Les garanties implicites constituent le principal risque de responsabilité contractuelle pour les donneurs de licences libres et ouvertes. C'est pourquoi ils tentent d'éviter leur application en ayant recours à des clauses élusives de responsabilité insérées en caractères apparents dans les contrats. Or, en droit civil québécois, la validité des clauses limitatives ou élusives de responsabilité dépend de la qualification donnée au contrat.
- Contrats d'adhésion* [131] Ainsi, les clauses d'exclusion de responsabilité peuvent être remises en question lorsque la relation contractuelle est fondée sur l'application d'un contrat imposé par l'une des parties. Or, l'une des principales caractéristiques des licences libres et ouvertes réside dans le fait qu'elles sont imposées par les donneurs de licence. Le plus souvent, ceux-ci reprennent l'intégralité d'un contrat type (tel que la GPL) et l'imposent à tous ceux qui désirent utiliser leur logiciel. Le licencié n'a donc à priori aucune possibilité d'en négocier les termes. Pour cette raison, ces licences peuvent le plus souvent être considérées comme des contrats d'adhésion. Les clauses d'exclusion de responsabilité qui y figurent pourraient être considérées comme des clauses abusives prohibées par l'article 1437 du Code civil du Québec.
- Contrats nommés* [132] En ce qui concerne les contrats nommés, le régime de la vente est le seul à interdire les clauses de non-garantie dans certaines circonstances, particulièrement lorsque le vendeur est un professionnel. À l'inverse, elles sont généralement permises en matière de louage. En ce qui concerne les garanties contre l'éviction, elles empêchent le licencié de réclamer des dommages et des intérêts, mais son droit à la résiliation de la licence ou à la diminution du prix n'est pas affecté. En matière de vices cachés, rien n'interdit aux parties de faire peser la totalité des risques sur les épaules du licencié. Ce dernier principe vaut également en matière de prêt, dans la mesure où la garantie contre les vices cachés peut y être appliquée.
- Bonne foi* [133] Toutefois, l'exclusion de responsabilité sera toujours inopérante lorsque le donneur de licence a commis une faute lourde ou intentionnelle. La bonne foi et l'absence d'intention de nuire des donneurs de licence sont donc des éléments déterminants lors de l'évaluation de la validité des clauses élusives de responsabilité.
- Préjudice corporel ou moral* [134] Au Québec, les clauses de non-garantie peuvent aussi être inefficaces quant à certains types de dommages. Ainsi, le Code civil du Québec interdit d'exclure ou de limiter sa responsabilité pour le préjudice corporel ou moral causé à autrui. L'exclusion demeure tout de même utile dans ces circonstances, puisqu'elle équivaut à la dénonciation d'un danger, ce qui peut mener à la réduction de la responsabilité du donneur de licence.
- Interventions* [135] De surcroît, même lorsque les clauses d'exclusion sont autorisées, les tribunaux se

*judiciaires* réservent généralement le droit d'en limiter la portée, en tenant compte des circonstances. Par conséquent, il est possible que les clauses exclusives de responsabilité incluses dans les licences libres et ouvertes ne soient pas toujours opposables aux tiers et puissent parfois être privées de leurs effets. Il demeure que, de façon générale, la balance des probabilités pèse en faveur de leur validité.

*Références* *Code civil du Québec*, L.Q. 1991, c. 64, art. 1437, 1474, Web :  
<<http://www.canlii.org/fr/qc/legis/lois/lq-1991-c-64/derniere/lq-1991-c-64.html>>

MackAAY (E.), « Le marché du progiciel – licence ou vente? », (1994) 6 Cah. Prop. Int. 401

VIVANT (M.) et autres, *Lamy droit de l'informatique et des réseaux*, Lamy, Paris, 2001,

n° 1129

### 3. Principaux enjeux opérationnels

[136] Ce chapitre aborde les difficultés généralement soulevées lors de l'analyse des solutions fondées sur le recours aux logiciels libres et ouverts. Les risques et les avantages potentiels associés aux logiciels libres et ouverts y sont détaillés, ainsi que les techniques permettant de les balancer aussi objectivement que possible.

#### 3.1. Coût total de propriété

*Enjeux* [137] Lorsque vient le moment d'acquérir une solution logicielle pour répondre à un besoin technologique, le choix doit être effectué en fonction des coûts engendrés pour l'Administration. Les logiciels libres et ouverts étant généralement téléchargeables gratuitement sur Internet, on peut être porté à croire que ceux-ci n'occasionnent aucun coût. Toutefois, bien que les fournisseurs de logiciels libres et ouverts ne perçoivent aucuns frais de licence, le recours à de tels logiciels est susceptible d'engendrer d'autres types de coûts, qui sont communs à l'ensemble des solutions logicielles.

[138] Dans ce contexte, l'évaluation du coût d'un logiciel spécifique exige de prendre en compte non seulement le coût d'achat initial mais tous les aspects de l'utilisation et de l'entretien du système. Il s'agit alors de calculer le coût total de propriété du logiciel désigné qui variera toujours en fonction des besoins et de l'environnement existant. En agissant de la sorte, il devient possible de déterminer la valeur économique de l'investissement envisagé. Cette notion est d'autant plus importante qu'elle trouve tout autant application aux logiciels libres et ouverts qu'aux logiciels propriétaires. Grâce à elle, il devient possible de comparer les coûts de solutions logicielles fondées sur des modèles d'affaires diamétralement opposés.

*Guide* [139] À cet effet, le Secrétariat du Conseil du trésor met à la disposition de l'Administration un guide de l'utilisateur du coût total de propriété (CTP). En plus d'aborder en détail les différentes composantes du coût total de propriété, cet outil inclut un chiffrier destiné à faciliter l'évaluation au cas par cas du coût total de propriété de logiciels libres et ouverts spécifiques.

*Références* Secrétariat du Conseil du trésor, *Coût total de propriété – Guide de l'utilisateur*, 2010.

#### 3.2. Assistance technique

*Enjeux* [140] La disponibilité de l'assistance technique est sans conteste la préoccupation principale des autorités ayant pour objectif d'acquérir un logiciel libre et ouvert dans un contexte administratif. Ceci est paradoxal dans la mesure où le modèle d'affaires des logiciels libres et ouverts favorise justement une très grande flexibilité quant aux choix des solutions d'assistance technique. En effet, le soutien d'un logiciel libre et ouvert peut être pris en main par l'Administration elle-même ou par tout fournisseur disposant de compétences et de moyens suffisants. Compte tenu de la grande diversité des logiciels libres et ouverts existants, l'offre en matière de service peut être plus ou moins diversifiée pour un logiciel donné. Il s'agit alors d'évaluer attentivement les différentes options possibles et d'effectuer les choix appropriés en fonction des circonstances.

*Communauté des* [141] Il est important de comprendre que les logiciels libres et ouverts sont soutenus et

*utilisateurs*

entretenus, non pas uniquement par un éditeur logiciel traditionnel mais également par une communauté constituée de ses utilisateurs, qu'il s'agisse d'éditeurs, de sociétés de services spécialisées ou d'utilisateurs. Ce sont ces communautés qui décident de l'orientation technologique, de l'adaptation et de l'évolution du code source ainsi que des versions et des mises à jour qui seront rendues disponibles. Un logiciel libre et ouvert évolue proportionnellement au dynamisme de sa communauté et aucune garantie de service ou de qualité ne doit être attendue de celle-ci. Cependant, en présence d'un marché, les fournisseurs se joindront rapidement à la communauté en offrant des services d'assistance technique.

*Assistance interne*

[142] Du point de vue de l'Administration, il est possible d'envisager la prise en charge à l'interne des tâches associées au soutien d'un logiciel libre et ouvert. Évidemment, cela requiert que le personnel concerné possède les compétences et les disponibilités requises pour assurer la gestion en continu de la solution choisie. Dans une telle situation, l'Administration voit à assister ses employés dans leur utilisation du logiciel (assistance de premier niveau), qui sera intégré et adapté à son propre environnement (assistance de deuxième niveau). Par ailleurs, des ressources plus ou moins grandes devraient être également consacrées à l'interaction avec la communauté, afin de contribuer aux tâches de débogage lorsque cela s'avère requis (assistance de troisième niveau).

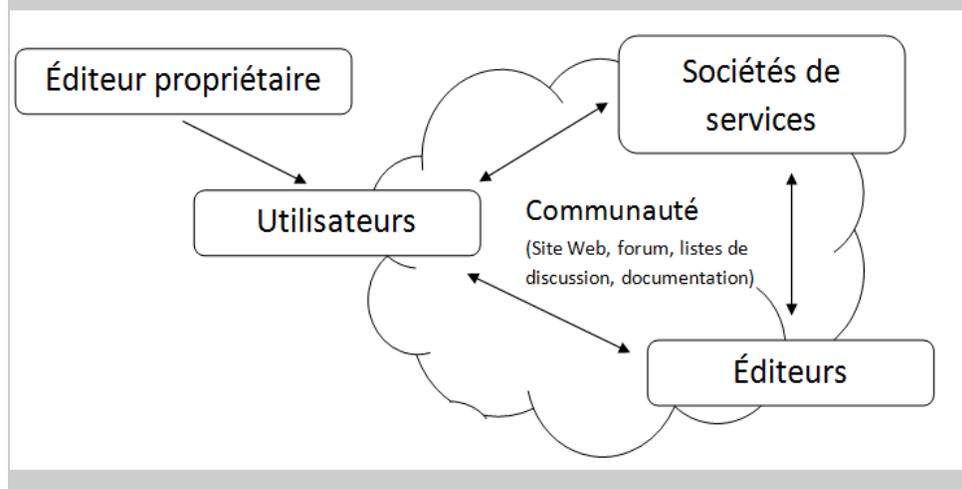
[143] Les ressources disponibles à l'interne pouvant être limitées, un plan de continuité des services devrait toujours être élaboré en parallèle, afin de déterminer les solutions d'assistance externe en cas de besoin.

*Assistance externe*

[144] Les logiciels libres et ouverts les plus populaires bénéficient d'une offre d'assistance technique similaire à ce qui existe en matière de logiciels propriétaires. Ainsi, il est généralement possible pour l'Administration d'acquérir d'un fournisseur de services externe l'ensemble des services d'assistance de premier et de deuxième niveau. Dans ce cas, c'est le fournisseur lui-même qui doit prendre en charge l'interaction avec la communauté du logiciel libre et ouvert concernée qui assure elle-même l'assistance de troisième niveau.

[145] Ce type d'assistance externe peut être fourni par les grands éditeurs internationaux qui supportent presque unanimement les technologies les plus populaires, comme Linux, Apache et MySQL. Ceux-ci offrent généralement plusieurs solutions d'assistance technique distinctes, impliquant un degré d'implication et des prix progressifs. De la même façon, plusieurs sociétés de services québécoises proposent une offre similaire, voire spécialisée dans des champs d'expertise spécifiques. Fait à noter dans le cas de cette offre distribuée, même lorsque l'éditeur original d'un logiciel libre et ouvert disparaît, une société de services pourra prendre sa place et continuer à soutenir l'Administration.

Tableau 4 – Écosystème de l'offre d'assistance technique des logiciels libres et ouverts



Marché compétitif

[146] Cette flexibilité quant au choix des solutions d'assistance technique disponibles est une conséquence directe du modèle de développement collaboratif des logiciels libres et ouverts. L'existence de ce marché compétitif en matière d'offre de service est l'une des caractéristiques fondamentales qui distingue les logiciels libres et ouverts des logiciels propriétaires. Au modèle traditionnel, où une entreprise est en position d'imposer les termes et les conditions de l'assistance technique qu'elle fournira pour son logiciel, s'ajoute maintenant un modèle ouvert où tout fournisseur est en mesure de proposer une offre concurrentielle.

### 3.3. Fiabilité et sécurité

Enjeux

[147] Alors que les logiciels libres et ouverts sont de plus en plus utilisés pour effectuer des tâches critiques, il est important de se questionner sur leur capacité à faire face efficacement aux menaces actuelles en matière de sécurité informatique. Bien que les développeurs de logiciels libres et ouverts de même que les développeurs de logiciels propriétaires aient recours aux mêmes techniques pour améliorer la fiabilité et la sécurité de leur code (révision par les pairs, test de qualité, évaluation des versions, etc.), leurs modèles de développement diamétralement opposés ont un impact important sur ce plan.

Disponibilité du code

[148] Selon le modèle de développement traditionnel, seuls les employés actifs dans le développement du logiciel ont accès à son code source. Pour certains, ce modèle présente l'avantage de rendre difficile l'identification des failles de sécurité et des défauts de programmation potentiellement exploitables. À l'inverse, le code source des logiciels libres et ouverts est publiquement accessible, permettant ainsi à tout un chacun de l'analyser. Pour plusieurs, cette transparence garantit qu'un grand nombre de personnes analyseront le code, accroissant ainsi les chances de découvrir rapidement toute anomalie qui pourrait s'y trouver. Selon ce modèle, c'est justement la transparence qui assure une meilleure sécurité informatique.

[149] Dans les faits, ni le modèle en vase clos ni le modèle collaboratif ne constituent une panacée en matière de sécurité informatique. Bien que la sécurité par l'obscurantisme soit largement rejetée depuis les années 1980, les éditeurs de logiciels propriétaires ont souvent des incitatifs économiques suffisants pour recruter une main-d'œuvre spécialisée capable d'assurer un niveau de sécurité adéquat. De façon similaire, alors que l'accès au code permet théoriquement de miser sur une large communauté de développeurs, il faut

qu'un nombre suffisant d'entre eux soient mobilisés, ce qui peut être particulièrement difficile à réaliser dans des domaines spécialisés d'expertise.

*Importance du  
contexte*

[150] Ainsi, il existe un écart important entre le niveau de sécurité et de fiabilité des différentes solutions libres. Alors que certaines d'entre elles, comme Linux, font l'objet d'efforts soutenus de centaines de programmeurs sur une base quotidienne et sont soumis à des processus formels d'assurance de la qualité, d'autres n'ont fait l'objet d'aucune mise à jour depuis des années. Dans ces circonstances, l'évaluation de la fiabilité et de la sécurité des logiciels doit être réalisée en tenant compte du contexte propre à chacun d'entre eux. Parmi tous les facteurs potentiels, les caractéristiques suivantes sont généralement considérées comme ayant un impact important sur le niveau de sécurité d'un logiciel libre et ouvert :

- ▲ La taille de la base de développeurs et l'étendue de leur distribution;
- ▲ La rapidité du retour d'information de la part des usagers;
- ▲ La fraction des utilisateurs étant aussi des développeurs;
- ▲ La rapidité du cycle de déploiement;
- ▲ Le volume de documentation disponible.

*Flexibilité*

[151] Pour l'Administration, les logiciels libres et ouverts ont cependant l'avantage d'offrir une très large flexibilité quant aux stratégies pouvant être adoptées en matière de sécurité informatique. Ils permettent, entre autres, d'enrichir le code source ou d'effectuer des contrôles de sécurité complémentaires lorsque cela s'avère nécessaire. De telles stratégies peuvent être réalisées à l'interne ou confiées à des fournisseurs externes.

*Indemnisations et  
garanties*

[152] À l'inverse, l'imputabilité qui caractérise le monde corporatif et qui découle de l'application des principes de la responsabilité civile semble désavantager les logiciels libres et ouverts lorsque des dommages découlent d'une faille de sécurité ou d'un défaut de programmation. Toutefois, cette affirmation doit être relativisée, compte tenu des restrictions extensives insérées dans les licences des logiciels propriétaires, du petit nombre d'exemples de dédommagement assuré par des éditeurs de logiciels, et de la capacité potentiellement limitée d'un fournisseur donné à dédommager l'Administration.

CHARPENTIER (R.), Richard CARBONE (R.), « Logiciels libres et ouverts : Survol et guide préliminaire pour le gouvernement canadien », Recherche et développement défense, 2004, Web : <<http://www.abhato.net.ma/index.php/fre/Maalama-Textuelle/Sciences-de-l'information/G%C3%A9n%C3%A9ralit%C3%A9s-et-aspects-theoriques-des-sciences-de-l'information/Aspects-juridiques/Logiciels-Libres/Logiciels-libres-et-ouverts.Survol-et-guide-pr%C3%A9liminaire-pour-le-gouvernement-canadien>>

WHEELER (D.), « High Assurance (for Security or Safety) and Free-Libre/Open Source Software (FLOSS)... with Lots on Formal Methods/Software Verification », 2006, Web : <<http://www.dwheeler.com/essays/high-assurance-floss.html>>

DAMIANI (E.), ARDAGNA (C. A.), IOINI (N.), *Open Source Systems Security Certification*, Springer, New York, 2009

WIKIPEDIA, « Comparison of Windows and Linux », 2011, Web : <[http://en.wikipedia.org/wiki/Comparison\\_of\\_Windows\\_and\\_Linux](http://en.wikipedia.org/wiki/Comparison_of_Windows_and_Linux)>

### 3.4. Interopérabilité

- Enjeux* [153] L'interopérabilité peut être définie comme la capacité qu'ont deux systèmes de se comprendre l'un et l'autre et de fonctionner en synergie. Cette synergie peut prendre plusieurs formes, dont l'échange de données, la lecture et l'écriture dans le même format de fichier ou l'utilisation de protocoles communs. Elle est particulièrement importante pour l'Administration qui manipule énormément de documents électroniques dans le cadre de ses activités alors que l'acquisition des logiciels y est grandement distribuée. Le choix de solutions interopérables est particulièrement pertinent dans le cadre des initiatives du Gouvernement en ligne qui requiert une architecture orientée vers les services. Dans ce contexte, la sélection de toute solution informatique devrait tenir compte de cette considération.
- Standards ouverts* [154] Une caractéristique fondamentale favorisant l'interopérabilité des systèmes informatiques repose sur l'utilisation de standards ouverts. Les standards ouverts sont déposés auprès d'une organisation neutre, plutôt qu'auprès d'un éditeur de logiciels particulier, et sont rendus publiquement accessibles. Ils permettent donc à tous de développer des logiciels basés sur ces spécifications sans courir le risque de violer la propriété intellectuelle d'un tiers. En accordant à l'Administration un choix entre plusieurs technologies concurrentes pour réaliser une même fonction, ils garantissent la pérennité des investissements et des solutions retenues.
- [155] Or, il existe une grande synergie entre les standards ouverts et les logiciels libres et ouverts. En effet, les standards ouverts ont besoin de logiciels libres et ouverts pour assurer une plus grande diffusion, et les éditeurs de logiciels libres et ouverts s'appuient sur ces standards pour s'insérer dans l'architecture des grandes organisations. De ce point de vue, le recours aux logiciels libres et ouverts favorise grandement l'interopérabilité.
- Compatibilité* [156] Toutefois, l'interopérabilité doit être analysée dans le contexte propre à chaque projet informatique. Au sein d'une même organisation, celle-ci peut également être atteinte en uniformisant l'utilisation des technologies. À titre d'exemple, l'intégration d'un logiciel libre et ouvert unique dans un environnement homogène ayant recours à un protocole de communication propriétaire a fort peu de chance de produire les bénéfices escomptés. Il s'agit alors d'assurer la compatibilité entre des systèmes indépendants. Cette compatibilité se trouve renforcée lorsqu'elle repose sur des standards ouverts assurant l'interopérabilité.
- Modularité* [157] Une autre composante de l'interopérabilité tient dans la modularité des solutions retenues. Même reposant sur des standards libres et ouverts compatibles avec un environnement donné, une solution logicielle doit encore être conçue de façon à ce que ses différentes composantes se complètent mutuellement. L'écosystème distribué des logiciels libres et ouverts tend justement à produire un environnement diversifié composé de nombreux modules interchangeable, eux-mêmes soutenus par des communautés indépendantes. L'espace de travail Eclipse en est un très bon exemple. S'appuyant sur Java et délivré par IBM, il sert de fondation aux plateformes de nombreux éditeurs produisant le « *middleware* » servant à relier les applications entre elles.
- Compatibilité des licences* [158] Une limitation inhérente à l'interopérabilité des logiciels libres et ouverts a trait à la compatibilité de leurs licences. Cet aspect est analysé en détail plus loin, mais il est important de noter que des logiciels même libres, techniquement interopérables, ne peuvent pas toujours être intégrés, à cause des restrictions imposées par leurs licences respectives.
- Cadre commun d'interopérabilité* [159] Le gouvernement du Québec s'est récemment doté d'un cadre commun d'interopérabilité destiné à recenser les normes et les standards relatifs aux ressources informationnelles, qui vise à soutenir l'interopérabilité de ses systèmes en matière d'intégration des données, de formats d'échange de documents, d'interconnexion des systèmes de télécommunication et de soutien à la langue française et à l'internationalisation. De façon similaire, le Secrétariat du Conseil du trésor met à la

disposition des ministères et des organismes une collection de notes sur des normes ouvertes en technologies de l'information. Ces outils devraient toujours servir de guide, lorsqu'il est question de sélectionner une solution technologique sur la base du critère de l'interopérabilité.

*Références*

COALLIER (F.), « Étude sur l'architecture gouvernementale ouverte, évolution des normes, des standards de consortium et des logiciels libres », CIRANO, 2005, Web : <<http://www.cirano.qc.ca/pdf/publication/2005RP-10.pdf>>

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, "ISO - International Organization for Standardization", 2011, Web : <<http://www.iso.org>>

SECRETARIAT DU CONSEIL DU TRÉSOR, « Cadre commun d'interopérabilité », 2009, Web <<http://www.msg.gouv.qc.ca/normalisation/interopabilite.html>>

WORLD WIDE WEB CONSORTIUM, "World Wide Web Consortium (W3C)", 2011, Web : <<http://www.w3.org>>

### 3.5. Maturité et longévité

*Enjeux*

[160] Le nombre de logiciels libres et ouverts pouvant être téléchargés sur Internet et immédiatement utilisés est impressionnant. Le site SourceForge met en référence à lui seul plus de 260 000 logiciels libres et ouverts, et cela ne constitue qu'une partie de l'ensemble. Évidemment, tous ces projets n'ont pas atteint le même niveau de maturité technologique. Un grand nombre d'entre eux en sont encore à leurs balbutiements et n'atteindront peut-être jamais un stade de développement suffisamment avancé pour que leur utilisation puisse être avantageuse. À l'inverse, d'autres sont soutenus par des communautés de plusieurs milliers de développeurs et évoluent depuis plus d'une décennie.

[161] Le choix de l'un ou l'autre de ces logiciels peut donc avoir un impact important sur la pérennité d'une solution technologique. Ce choix est d'autant plus difficile à effectuer que les logiciels libres et ouverts sont souvent caractérisés par une pénurie d'informations promotionnelles, comme des fiches techniques, des brochures ou des communiqués de presse. Avant d'y procéder, il est donc essentiel d'évaluer la maturité (stade de développement) et la longévité (capacité à faire face aux évolutions technologiques) potentielle des solutions envisagées. Différents indicateurs facilitent cette évaluation :

- ▲ L'état du cycle de vie du logiciel;
- ▲ Le leadership au sein de la communauté;
- ▲ La vitesse de son développement;
- ▲ La taille de sa communauté;
- ▲ Son inclusion dans un marché compétitif.

*Cycle de vie*

[162] Comme tous les autres, les logiciels libres et ouverts naissent d'une idée, vivent plus ou moins longtemps en fonction du succès de leurs développeurs et finissent par mourir lorsqu'ils deviennent désuets ou cessent de répondre à un besoin. De façon générale, le processus de développement des logiciels libres et ouverts (leur cycle de vie) comporte trois étapes principales :

1. Phase initiale de développement : À cette étape, un développeur publie les fruits des travaux qu'il mène afin de faire face à un problème auquel il est confronté. Le code source est rendu public dans l'objectif d'intéresser une communauté d'utilisateurs faisant face au même problème. Cette phase est caractérisée par des

changements technologiques fréquents et le manque de finition du logiciel;

2. Phase de croissance : Le développeur initial reçoit des contributions de tiers qui améliorent le code du logiciel. Celui-ci devient graduellement plus fonctionnel et générique. La première version entièrement opérationnelle du logiciel (version 1.0) est distribuée;
3. Phase d'organisation : Une équipe se met en place afin de coordonner les efforts de développement et de prendre en compte les besoins de nouveaux utilisateurs. Les tâches de débogage et d'entretien sont réparties entre les membres de la communauté.

[163] Seuls les projets de logiciels libres et ouverts dans leur phase d'organisation présentent une maturité suffisante pour être retenus par l'Administration. Toutefois, lorsqu'une expertise spécialisée est disponible à l'interne et qu'un logiciel présente un intérêt exceptionnel, certains projets peuvent déroger à cette règle. Le risque que le logiciel en question n'atteigne jamais sa phase d'organisation doit alors être considéré.

*Leadership*

[164] Un indicateur important de la maturité d'un logiciel libre et ouvert tient dans le mode de gouvernance de sa communauté. La crédibilité de l'institution hôte y joue pour beaucoup, mais le fait que celle-ci ait des règles clairement établies, qu'elle ait réussi à limiter ou empêcher les fourches (explosion d'un projet en plusieurs projets distincts) par le passé et qu'un plan de développement soit mis de l'avant sont d'autres critères de succès.

*Vitesse de développement*

[165] Un autre indicateur de la maturité d'un logiciel libre et ouvert est la vitesse à laquelle son code source évolue. Celle-ci dépend largement de l'efficacité du mode de gouvernance du projet et peut être constatée par la période plus ou moins étendue entre la mise en production des différentes versions. La capacité d'un projet à préserver un *momentum* est un important facteur de succès et assure la disponibilité de ressources pour son entretien et son assistance technique. Cette caractéristique laisse également entrevoir une capacité à réagir rapidement aux innovations technologiques, garantissant la longévité du logiciel. À défaut, il y a un risque que le logiciel se dirige vers une obsolescence technique, conséquence susceptible de requérir une refonte du logiciel à moyen terme.

*Taille de la communauté*

[166] Évidemment, plus un logiciel libre et ouvert intéresse une large communauté d'utilisateurs et de développeurs, plus il est possible d'avoir confiance en ses résultats. Ceci est particulièrement vrai lorsque d'autres possibilités sérieuses coexistent. Bien que la taille exacte d'une communauté puisse difficilement être calculée, il est possible de l'évaluer en se fiant au nombre de développeurs concernés, au degré de participation aux listes de discussions et de forums sur le sujet, ainsi qu'aux études comparatives indépendantes publiées. La taille d'une communauté varie en fonction de la capacité de chaque logiciel à répondre aux besoins des développeurs, de la façon dont l'information à propos de celui-ci circule entre les projets connexes et des migrations de développeurs entre les différents projets libres et ouverts.

[167] Cependant, la faible taille d'une communauté ne peut pas être automatiquement interprétée comme le signe d'une maturité insuffisante. En effet, le segment de marché de certains logiciels libres et ouverts constitue une niche attirant peu de développeurs. Avec le temps, ceux-ci peuvent tout de même réussir à produire un logiciel de grande qualité.

*Marché compétitif*

[168] Finalement, l'indicateur par excellence de la maturité d'un logiciel libre et ouvert est sans conteste l'existence d'offres de services concurrentes le concernant. Ces offres peuvent prendre la forme de services d'installation, d'intégration, d'assistance technique ou tout simplement de la disponibilité de consultants compétents. L'existence d'un marché compétitif garantit que l'expertise requise par l'Administration est présente et devrait continuer de l'être à moyen terme.

- Références
- KRAFT (B.), HINDERINK (D.), "Evaluating Open Source Communities : Why Open Source is more than Code and why you should care", Transformation and Innovation Conference 2006, 2006, Web : [http://news.typo3.org/fileadmin/user\\_upload/evaluating\\_OS\\_communities.pdf](http://news.typo3.org/fileadmin/user_upload/evaluating_OS_communities.pdf)
- SPINELLIS (D.), *Code quality : the open source perspective*, Addison Wesley, Upper Saddle River, 2006
- WEISS (M.), MOROIU (G.), "Ecology and dynamics of Open Source Communities", dans SOWE (S. K.), *Emerging Free and Open Source Software Practices*, Hershey, IGI Publishing, 2007

## 4. Acquisition

[169] Ce chapitre aborde le contexte particulier qui caractérise l'acquisition de logiciels libres et ouverts par l'Administration, et ce, du moment où un besoin technologique est clairement défini jusqu'à la rédaction d'un contrat de services. Bien que le mode de distribution des logiciels libres et ouverts augmente grandement les possibilités d'échapper aux processus traditionnels d'approvisionnement, les solutions énumérées ci-dessous devraient fournir un encadrement adéquat à toute acquisition de logiciels libres et ouverts.

### 4.1. Quand choisir les logiciels libres et ouverts?

#### *Problématique*

[170] Le principal motif justifiant l'acquisition d'une solution informatique par l'Administration est l'apparition d'un besoin et la volonté d'y répondre. Que la solution soit fondée sur l'utilisation de logiciels libres et ouverts ou propriétaires, les critères de sélection demeurent les mêmes. Dans ce contexte, les principales considérations demeurent l'aptitude d'un logiciel à combler les besoins opérationnels cernés et son rapport qualité/prix.

#### *Besoins opérationnels*

[171] Avant de procéder à toute acquisition technologique, il faut détailler les caractéristiques du projet concerné et les impératifs à prendre en compte. Cette définition des besoins opérationnels et de leur cadre d'application doit être effectuée de la façon la plus inclusive possible afin de ne pas exclure prématurément des solutions potentielles. Un logiciel, qu'il soit libre ou propriétaire, ne devrait être retenu que s'il répond de façon adéquate aux besoins exprimés. Les particularités relatives à l'évaluation des offres de services fondées sur les logiciels libres et ouverts sont abordées plus loin.

#### *Rapport qualité/prix*

[172] La capacité d'une solution technologique à répondre aux besoins circonscrits doit aussi être mise en relation avec son coût. En plus du montant associé à la licence logicielle, il s'agit ici de calculer le coût total de propriété en fonction des modalités précisées plus haut.

#### *Facteurs de risques*

[173] À ces considérations s'ajoute la prise en compte du contexte technologique dans lequel la solution est vouée à s'intégrer. Toute nouveauté entraîne des risques de rejet et ce facteur de risque doit nécessairement être pris en considération. Les expériences menées au gouvernement du Québec illustrent que l'insertion d'un premier logiciel libre et ouvert dans une organisation implique un risque élevé d'échec. Au contraire, la nouveauté technologique crée moins de remous dans les ministères et les organismes l'ayant déjà expérimentée avec d'autres logiciels partageant le même modèle de distribution. Les autres principaux facteurs de risque à considérer sont l'expertise de l'intégrateur, l'adéquation culturelle de l'intégrateur et de l'Administration et le nombre d'utilisateurs concernés.

[174] Le contexte technologique a donc un impact important sur le succès ou l'échec des projets de migration vers les logiciels libres et ouverts. Afin de réduire les frictions potentielles, la mise en place d'expériences pilotes devrait être envisagée.

#### *Projets internes*

[175] Il est également possible d'avoir recours au modèle de développement des logiciels libres et ouverts dans le cadre de projets internes à l'Administration. Toutefois, tous les projets ne sont pas de bons candidats pour le modèle de développement des logiciels libres et ouverts. Ainsi, les projets grandement personnalisés et les applications très spécialisées ont peu de chance d'intéresser une communauté de développeurs désirent y contribuer. De façon similaire, lorsque des délais critiques sont en jeu, la capacité à respecter les échéanciers doit être évaluée attentivement puisque cet élément peut parfois s'avérer difficile à contrôler selon le modèle de développement retenu.

Confidentialité

[176] Dans le même ordre d'idées, le recours aux logiciels libres et ouverts doit être évité lorsque des considérations de confidentialité liées à la distribution du code source entrent en ligne de compte. Il peut d'abord s'agir de la confidentialité du code source appartenant à un tiers et que l'Administration a en sa possession en vertu d'ententes particulières. L'intégration de ce code source avec du code libre et son éventuelle distribution en ligne constituerait une divulgation de secrets industriels. La confidentialité peut aussi viser des données gérées par l'Administration, lorsque la distribution du code source est susceptible de mettre cette confidentialité en péril. Ce serait le cas du code d'un service Web dont l'étude facilite l'accès non autorisé à une base de données correspondante.

Valorisation de la propriété intellectuelle

[177] Dans tous les cas, les choix technologiques de l'Administration devraient tenir compte de son mandat de valoriser la propriété intellectuelle lui appartenant, comme le définit le Cadre de gestion et de valorisation de la propriété intellectuelle du gouvernement du Québec. Ainsi, lorsque l'Administration a l'intention de percevoir des redevances liées à sa propriété intellectuelle sur un logiciel en voie de développement, le choix de technologies libres doit être effectué soigneusement. Inversement, l'objectif de contribuer à l'innovation au Québec peut parfois être mieux servi par la libre circulation sur Internet du code source de logiciels développés par l'Administration. Si le cadre recommande que l'Administration bénéficie d'une contrepartie équitable en échange de l'octroi d'une licence sur sa propriété intellectuelle, rien n'exige que cette contrepartie soit de nature financière. Une contrepartie prenant la forme d'un apport technologique ou d'un savoir-faire de la part d'une communauté de développeurs peut tout aussi bien remplir ce critère.

ABDALLAH (H.), *Impact des logiciels libres sur un département TI : Étude comparative des logiciels développés à l'interne, des logiciels libres et des logiciels propriétaires dans une organisation*, Paris, Éditions universitaires européennes, 2010

ABOUBEKR (M.), Suzanne RIVARD (S.), « Évaluation de risque du projet de migration vers la suite bureautique libre sous Linux », CIRANO, 2005, Web : <<http://www.cirano.qc.ca/pdf/publication/2005RP-09.pdf>>

DÉVELOPPEMENT ÉCONOMIQUE, INNOVATION ET EXPORTATION QUÉBEC, « Cadre de gestion et de valorisation de la propriété intellectuelle résultant des activités scientifiques et techniques des ministères et des organismes publics du gouvernement du Québec », 2010, Web : <<http://www.mdeie.gouv.qc.ca/index.php?id=6156>>

## 4.2. Évaluation

Particularités

[178] Dans les situations où le recours aux logiciels libres et ouverts s'avère approprié, il faut être en mesure d'évaluer les différents choix disponibles. Bien que les principales étapes pour procéder à l'évaluation des logiciels soient relativement génériques, leur mise en œuvre doit être adaptée dans le cas des logiciels libres et ouverts. Ceci s'explique d'abord par les sources d'information à propos des logiciels libres et ouverts qui diffèrent grandement de celles des logiciels propriétaires. Alors que l'information sur ces derniers provient généralement du matériel promotionnel communiqué par les fournisseurs, les caractéristiques des logiciels libres et ouverts peuvent être déduites de l'étude de leur code source, d'analyses effectuées par des tiers, de discussions publiques entre développeurs, etc. Une autre différence majeure réside dans la possibilité pour l'Administration de les modifier et de les redistribuer. Ces éléments ont un impact important sur la flexibilité d'un logiciel, sur la disponibilité d'assistance technique et sur son coût.

*Guide* [179] Afin de faciliter les tâches liées à l'évaluation des logiciels, tout en tenant compte des particularités des logiciels libres et ouverts, le Secrétariat du Conseil du trésor met à la disposition de l'Administration un document intitulé *Portefeuille de logiciels libres et ouverts*. Ce guide propose un modèle d'évaluation de la maturité des logiciels libres et ouverts ainsi qu'une méthodologie de sélection des logiciels.

*Références* SECRÉTARIAT DU CONSEIL DU TRÉSOR, *Portefeuille de logiciels libres et ouverts*, 2010

### 4.3. Approvisionnement

*Acquisition directe* [180] Une grande partie des logiciels libres et ouverts auxquels l'Administration a recours sont obtenus, installés puis utilisés sans même qu'un fournisseur de services participe à l'opération. Cette forme d'approvisionnement inusité était plus ou moins inexistante jusqu'à tout récemment. Évidemment, l'acquisition directe par l'Administration requiert que l'expertise et les ressources pertinentes soient entièrement disponibles à l'interne.

*Téléchargement* [181] Le téléchargement du logiciel à partir d'un site Web constitue un autre type d'approvisionnement possible. Cette procédure simplifiée ne requiert aucune demande de crédits ou de bons de commande et peut être entreprise sans autre formalité. Cependant, cet état de fait ne dégage pas pour autant l'Administration de son devoir d'assurer une prise de décision éclairée et équitable. À défaut, elle est susceptible de s'engager trop rapidement envers certains logiciels libres et ouverts, situation qui mène éventuellement à une prolifération des solutions utilisées et nuit à l'intégrité technologique de son infrastructure. À cet effet, une évaluation des différentes formules envisageables devrait être menée aussi souvent que possible.

*Acquisition par un fournisseur* [182] Par ailleurs, acquérir des logiciels libres et ouverts par l'intermédiaire de contrats de services conclus avec des fournisseurs est une solution qui devrait être privilégiée dès que l'expertise et les ressources requises ne sont pas disponibles à l'interne.

*Types de services* [183] Vu la diversité des modèles commerciaux soutenant le marché des logiciels libres et ouverts, il est possible de faire appel à un fournisseur qui n'en est pas l'auteur, ni l'éditeur. Ceci contribue à créer une diversité d'offres de services distinctes. Certains fournisseurs se spécialisent dans l'installation et la migration vers des environnements libres. D'autres offrent à leurs clients le développement de solutions intégrées et personnalisées. D'autres encore se consacrent à l'entretien et à l'assistance technique. Les plus grands éditeurs de logiciels proposent généralement des solutions clefs en main intégrant l'ensemble de ces services.

*Indemnisations et garanties* [184] La participation d'un fournisseur de services dans le cadre de l'acquisition d'un logiciel libre et ouvert constitue un avantage non négligeable en raison de son imputabilité. Alors que les membres de la communauté de développeurs du logiciel libre et ouvert peuvent difficilement être tenus responsables des défauts de celui-ci, le fournisseur sélectionné est responsable de remplir les obligations de son contrat. Certains fournisseurs de logiciels libres et ouverts proposent même des garanties de qualité additionnelles. Les risques s'en trouvent donc réduits d'autant pour l'Administration.

*Relations avec la communauté* [185] Peu importe le mode d'approvisionnement ou le fournisseur choisi, il est primordial pour l'Administration d'assurer le maintien des relations avec la communauté en charge du développement du logiciel libre et ouvert sélectionné. Ceci est particulièrement vrai lorsqu'un logiciel libre et ouvert est adapté ou personnalisé par l'Administration. Si ces adaptations ne sont pas intégrées par la communauté des développeurs, des incompatibilités pourraient survenir entre la version adaptée et la version officielle du logiciel. Éventuellement, il pourrait être extrêmement difficile pour l'Administration de

profiter des mises à jour apportées à la version officielle.

*Évaluation des  
fournisseurs*

[186] Cette interaction nécessaire avec la communauté peut tout aussi bien être assurée par l'Administration elle-même que par son fournisseur de services. Dans ce dernier cas, l'engagement de celui-ci auprès de la communauté en question constitue une considération supplémentaire à prendre en compte lors du processus d'évaluation des fournisseurs. Pour le reste, les critères traditionnels d'évaluation des fournisseurs de logiciels propriétaires s'appliquent aux fournisseurs de logiciels libres et ouverts. Ainsi, l'évaluation de leurs capacités financières, de leur stabilité en affaires et de leurs ressources humaines demeure tout aussi pertinente.

*Étude d'opportunité*

[187] Lorsque vient le temps de sélectionner un fournisseur spécifique, une étude d'opportunité visant à évaluer les possibilités que représentent les logiciels libres et ouverts doit être réalisée pour tout remplacement, acquisition, renouvellement ou mise à niveau de logiciels pour un montant supérieur à 25 000 \$. Ainsi, les choix alternatifs disponibles sous licences libres et ouvertes doivent être considérés chaque fois qu'une dépense logicielle substantielle doit être effectuée. Afin d'être en mesure de se prévaloir de son droit de passer un contrat à commande avec un fournisseur unique, cette étude d'opportunité devra satisfaire le critère de la recherche sérieuse et documentée établie par le Règlement sur les contrats d'approvisionnement des organismes publics.

*Appels d'offres  
neutres*

[188] Par ailleurs, afin de promouvoir la concurrence et le traitement équitable des fournisseurs potentiels, l'Administration se doit d'accorder ses contrats de la façon la plus neutre et objective possible. À cet effet, le recours à un processus d'appel d'offres, public ou sur invitation, peut être requis. Dans l'affaire *Savoir-faire Linux inc. c. Régie des rentes du Québec*, la Cour supérieure du Québec est venue rappeler que tout processus d'approvisionnement requérant un appel d'offres devrait être rédigé de façon à permettre la participation de l'ensemble des fournisseurs en mesure de répondre aux besoins opérationnels définis, y compris les fournisseurs ayant recours à des logiciels libres. L'Administration doit éviter de formuler ses besoins en se limitant à des produits ciblés spécifiquement. Ainsi, les références à des systèmes d'exploitation spécifiques, à un modèle de développement ou à une licence particulière devraient être évitées. Il n'est donc pas plus adéquat d'exiger une solution libre et ouverte qu'il l'est d'exiger une solution propriétaire donnée.

[189] Évidemment, l'environnement et l'architecture en place contribueront souvent à déterminer le type de solutions proposées. Il n'en demeure pas moins que certains éditeurs de logiciels propriétaires proposent des versions adaptées à des environnements libres, et vice-versa.

*Appels d'offres  
transparentes*

[190] La nécessité d'effectuer des appels d'offres neutres n'implique pas de renoncer à évaluer les facteurs qui peuvent motiver le recours aux logiciels libres et ouverts. L'interopérabilité, l'importance accordée à l'indépendance face au fournisseur ainsi que la nécessité de permettre à l'Administration de poursuivre le développement à l'interne sont autant de facteurs qui peuvent être pris en compte. La transparence du processus d'appel d'offres réclame cependant que ces facteurs soient explicitement spécifiés à l'ensemble des fournisseurs. De cette façon, ceux-ci ont la possibilité de s'adapter et de rédiger leur offre en conséquence.

*Politiques internes*

[191] Finalement, l'acquisition de logiciels libres et ouverts ne dispense pas l'Administration du respect des diverses politiques applicables en matière d'approvisionnement. À titre d'exemple, l'obligation d'acquérir des produits permettant l'utilisation intégrale du français demeure valable. Parfois, des politiques sectorielles devront aussi être prises en compte.

#### Tableau 5 - Étapes du cycle d'approvisionnement

1. L'identification des besoins opérationnels et du cadre d'application;
2. L'évaluation des différentes solutions envisageables;
3. La sélection de la meilleure solution;
4. La sélection d'un ou de plusieurs fournisseurs (optionnel);
5. L'établissement d'un projet pilote (optionnel);
6. L'établissement d'un plan de mise en œuvre;
7. La mise en œuvre.

#### Références

AUSTRALIAN GOVERNMENT INFORMATION MANAGEMENT OFFICE, "A Guide to open Source Software for Australian Government Agencies", Department of Finance and Deregulation, 2005, Web : <<http://www.finance.gov.au/publications/guide-to-open-source-software/index.html>>

IDABC EUROPEAN EGOVERNMENT SERVICES, "Guideline for Public administrations on Procurement and Open Source Software", OSOR.eu, 2010, Web : <<http://www.osor.eu/studies/expert-guidance/guideline-for-public-administrations-on-procurement-and-open-source-software-2010>>

KBST, "Migration Guide – A guide to migrating the basic software components on server and workstation computers", ePractice.eu, 2005, Web : <<http://www.epractice.eu/en/library/281501>>

Règlement sur les contrats d'approvisionnement des organismes publics, R.R.Q., c. C-65.1, r. 2, Web : <<http://www.canlii.org/fr/qc/legis/regl/rrq-c-c-65.1-r-2/derniere/rrq-c-c-65.1-r-2.html>>

Savoir-faire Linux inc. c. Régie des rentes du Québec, 2010 QCCS 2375, Web : <<http://www.canlii.org/fr/qc/qccs/doc/2010/2010qccs2375/2010qccs2375.html>>

SECRETARIAT À LA POLITIQUE LINGUISTIQUE, « Politique d'utilisation du français dans les technologies de l'information et des communications », 2006, Web : <<http://www.spl.gouv.qc.ca/documentation/loisreglementspolitiques/puftic/>>

Secrétariat du Conseil du Trésor, « Politique de gestion contractuelle concernant la conclusion des contrats d'approvisionnement, de services et de travaux de construction des organismes publics », 2011, Web : <[http://www.tresor.gouv.qc.ca/fileadmin/PDF/faire\\_affaire\\_avec\\_etat/marches\\_publics/POLITIQUE\\_D\\_E\\_GESTION\\_CONTRACTUELLE\\_concernant\\_conclusion.pdf](http://www.tresor.gouv.qc.ca/fileadmin/PDF/faire_affaire_avec_etat/marches_publics/POLITIQUE_D_E_GESTION_CONTRACTUELLE_concernant_conclusion.pdf)>

#### 4.4. Contrats de services

#### Problématique

[192] Après avoir trouvé la solution adéquate et sélectionné un ou plusieurs fournisseurs

pour la mettre en œuvre, il faut établir une relation d'affaires. Cela implique la rédaction de contrats de services. La gestion de la propriété intellectuelle des logiciels libres et ouverts faisant l'objet de particularités, la rédaction d'un contrat de services dans ce contexte requiert certains ajustements. À ce sujet, les normes fournies par le ministère de la Justice en matière de contrats de services professionnels fournissent la flexibilité requise.

*Propriété du code*

[193] La principale question soulevée lors de la rédaction d'un tel contrat concerne la propriété intellectuelle résultant des services professionnels acquis par l'Administration. À la différence des éditeurs de logiciels propriétaires, les fournisseurs de solutions basées sur l'utilisation de logiciels libres et ouverts ne sont pas nécessairement propriétaires du code source qu'ils redistribuent. Ils en acquièrent eux-mêmes une grande partie sous les termes de licences libres et ouvertes comportant plus ou moins de restrictions. Cependant, cela ne limite en rien leur propriété du code source résultant de leurs propres contributions et adaptations aux logiciels concernés. En conséquence, les contrats de services de l'Administration devraient recourir au mécanisme approprié dans chaque cas d'espèce, soit la cession de droits, soit la concession (licence). Ces deux mécanismes ne sont pas exclusifs et peuvent être appliqués conjointement à différents éléments de la solution acquise.

*Cession*

[194] La cession de droit implique un transfert de certains droits du fournisseur, ou de la totalité d'entre eux, vers l'Administration. Évidemment, le fournisseur ne peut céder ce qui ne lui appartient pas, ce qui limite le recours à la cession dès lors que la propriété d'un logiciel libre et ouvert est partagée par une communauté d'utilisateurs. Au surplus, bien des fournisseurs hésiteront à céder leurs droits, s'ils ont un intérêt économique à les licencier à d'autres clients. Toutefois, le mécanisme de la cession peut s'avérer adéquat lorsqu'il est question de services d'adaptation et d'intégration personnalisés. Par ce mécanisme, la propriété du code source, qui est propre à l'Administration, lui est assurée, garantissant ainsi un accès au code source et un maximum de flexibilité quant à sa gestion subséquente.

*Licence*

[195] Le mécanisme de la licence permet au fournisseur d'exercer son droit d'auteur de façon à autoriser l'Administration à poser des gestes qui autrement constituerait des infractions. L'inclusion d'une licence dans les contrats de services professionnels s'avère être la seule solution possible, dès que le mécanisme de la cession s'avère inapproprié. Afin de faciliter la gestion des droits, celle-ci devrait généralement reproduire les termes de la licence libre et ouverte utilisée pour la distribution du logiciel en question, bien qu'il soit toujours possible d'avoir recours à une autre licence. Les sections subséquentes de ce guide devraient être consultées afin de déterminer si la licence convient aux opérations envisagées.

*Droit d'auteur de la Couronne*

[196] En cas d'omission ou d'incertitude quant au mode de transmission des droits, le droit canadien prévoit que le droit d'auteur sur les œuvres préparées ou publiées sous la direction de l'Administration lui appartient. Évidemment, un logiciel libre et ouvert développé par une communauté préalablement à son acquisition par l'Administration est exclu du champ d'application de cette disposition. Au contraire, le produit des travaux d'intégration de ce même logiciel, réalisés par un fournisseur de l'Administration, y sont probablement soumis. Étant donné l'incertitude qui subsiste à ce sujet, le recours à un contrat de cession ou de licence s'avère beaucoup plus prudent.

*Droit de paternité*

[197] Aux clauses de cession ou de licence s'ajoutent celles relatives aux droits moraux qui s'attachent à la personne de l'auteur. Il n'est pas possible de céder ses droits, mais l'auteur peut toujours choisir d'y renoncer contractuellement. L'auteur d'un logiciel libre et ouvert, au même titre que tout autre auteur, jouit d'abord d'un droit de paternité sur son code. Il peut s'agir de l'employé d'un fournisseur ou de toute autre personne physique. Cette mesure implique le droit de voir son nom mentionné avec celui des autres auteurs, s'il est oublié. Au contraire, elle ne devrait pas permettre de forcer la mention de l'auteur, si la pratique fait en sorte que ce n'est pas le cas. En effet, les problèmes liés à la clause publicitaire de la licence Berkeley Software Distribution (BSD) originale ont, par le passé,

démontré le caractère déraisonnable de l'attribution excessive de crédits aux auteurs de logiciels libres et ouverts. Par ailleurs, certaines licences libres et ouvertes, comme la licence Apache 2.0, contiennent des clauses abordant la question de la paternité. Cette stipulation équivaut alors à une protection contractuelle de ce droit qui n'est pas reconnue partout dans le monde. De toute façon, puisque l'exercice de ce droit est peu susceptible de nuire à l'Administration, aucune renonciation ne devrait être réclamée à ce sujet.

*Droit à l'intégrité*

[198] Il en va autrement du second droit moral dont bénéficie l'auteur d'un logiciel libre et ouvert, soit le droit à l'intégrité. Ce droit permet à l'auteur de faire cesser toute modification préjudiciable à son honneur ou à sa réputation. Ainsi, tous les contributeurs à un logiciel libre et ouvert, peu importe qu'ils aient ou non cédé leurs droits patrimoniaux à une institution chargée de la gestion du projet, sont en mesure de forcer une rectification si leur honneur ou leur réputation est atteinte par des développeurs subséquents. Ce pourrait être le cas, par exemple, d'un logiciel modifié par un licencié pour faciliter la création d'organismes génétiquement modifiés (OGM). De plus, la Cour suprême du Canada semble pousser encore plus loin l'interprétation du droit à l'intégrité. Ainsi, dans l'arrêt *Galerie d'art du Petit Champlain inc. c. Théberge*, celle-ci a établi que le droit à l'intégrité protégerait également les modifications pouvant être effectuées à la structure de l'œuvre. Selon cette interprétation, l'auteur d'un logiciel libre et ouvert aurait toujours un droit de regard sur les modifications effectuées par des licenciés subséquents, malgré les termes permissifs de la licence. Par exemple, il lui serait probablement permis de s'opposer à certaines bifurcations du code qui impliquent le plus souvent une réorganisation importante du logiciel. Or, l'acquisition d'un logiciel libre et ouvert facilite justement le remodelage ou l'adaptation de celui-ci en fonction des objectifs bien précis de l'Administration. En conséquence, l'exercice de ce droit pourrait limiter considérablement les bénéfices anticipés. Pour cette raison, il s'avère avisé d'exiger une renonciation au droit à l'intégrité de la part des fournisseurs de logiciels libres et ouverts, lorsque cela s'avère possible.

*Références*

Galerie d'art du Petit Champlain inc. c. Théberge, 2002 CSC 34, Web : <http://www.canlii.org/fr/ca/csc/doc/2002/2002csc34/2002csc34.html>

GOUDREAU (M.), « Le droit moral de l'auteur au Canada », (1994) 25 R.G.D. 428, Web : [http://www.revuegeneralededroit.ca/index2.php?option=com\\_sobi2&sobi2Task=dd\\_download&fid=484&Itemid=0](http://www.revuegeneralededroit.ca/index2.php?option=com_sobi2&sobi2Task=dd_download&fid=484&Itemid=0)

MINISTÈRE DE LA JUSTICE, *Guide de rédaction des contrats de services professionnels*, 2005

VETTER (G.R.), "The Collaborative Integrity of Open-Source Software", (2004) Utah L. Rev. 563

## 5. Utilisation

[199] Ce chapitre aborde les principales questions découlant de la simple utilisation d'un logiciel libre et ouvert par l'Administration, c'est-à-dire son installation et son exécution par des utilisateurs. Il traite des différentes licences privilégiées pour une utilisation dans l'Administration et des conditions et des restrictions auxquelles l'Administration doit faire face.

### 5.1. Licences privilégiées

#### *Problématique*

[200] Il existe un nombre considérable de licences libres et ouvertes. Cette situation résulte de l'origine diversifiée des développeurs de logiciels libres et ouverts et du fait que chacun d'entre eux a des besoins plus ou moins grands en ce qui a trait à la protection de sa propriété intellectuelle. Il n'en demeure pas moins que la plupart des licences libres et ouvertes trouvent leur source dans l'une des quelques licences « classiques ».

[201] Par définition, les conditions d'utilisation énoncées dans les licences libres et ouvertes doivent permettre à tous les utilisateurs d'utiliser, de modifier et de distribuer les logiciels auxquels elles s'appliquent. Généralement, les conditions attachées à la simple utilisation du logiciel sont minimales. Cette caractéristique permet à l'Administration d'utiliser un vaste choix de logiciels libres, sans craindre de conséquences juridiques importantes.

#### *Certification*

[202] En raison de l'accroissement constant du nombre de licences libres et ouvertes, certaines organisations, comme la FSF et l'OSI, ont établi des critères permettant d'évaluer la qualité et la validité de ces licences. L'OSI s'est ainsi donné comme mission d'encadrer les licences libres et ouvertes en instaurant une procédure d'approbation. De façon générale, les licences reconnues par ces organisations sont caractérisées par une certitude juridique suffisante pour satisfaire aux besoins de l'Administration.

#### *Critères*

[203] Les principaux critères à surveiller lors du choix d'un logiciel libre et ouvert destiné à être utilisé par l'Administration sont les suivants :

1. L'accès et l'utilisation du code source sont assurés;
2. Aucuns frais ne sont imposés en cas de redistribution;
3. Les modifications sont permises et peuvent être redistribuées;
4. Toute discrimination envers des personnes ou des groupes est exclue;
5. Aucune condition ou restriction n'est imposée concernant un domaine d'activité ou une cause;
6. La licence est le seul document à définir la gestion des droits sur le logiciel;
7. La licence n'est pas exclusive à un logiciel spécifique;
8. Aucune condition ou restriction n'est imposée sur d'autres logiciels que celui couvert par la licence;
9. La licence est neutre technologiquement.

Tableau 8 – Les principales licences libres et ouvertes à privilégier et leurs caractéristiques

Caractéristique	CLLAQ	GPL 2.0	GPL 3.0	Apache 2.0	BSD 2.0	MIT
<i>Peut faire l'objet d'une compilation avec d'autres types de licences</i>	✓	✓	✓	✓	✓	✓
<i>Peut être exécutée avec d'autres types de licences</i>	✓	✓	✓	✓	✓	✓
<i>Le code source peut être intégré avec d'autres types de licences</i>	✓			✓	✓	✓
<i>Le code source peut être redistribué</i>	✓	✓	✓	✓	✓	✓
<i>Garantit les mêmes droits à tous les utilisateurs des logiciels dérivés</i>		✓	✓			
<i>Protection de la paternité</i>	✓					
<i>Protection des marques de commerce</i>	✓			✓		
<i>Concession de brevets</i>			✓	✓		
<i>Interdiction des mesures anticontournement</i>			✓			
<i>Licence applicable aux contributions</i>	✓			✓		
<i>Garanties</i>						

*Licences à risque*

[204] Il est important de se rappeler que le tableau précédent ne fait référence qu'aux licences « classiques ». En pratique, une infinité d'autres licences coexistent et certaines d'entre elles, même si elles respectent les critères établis plus haut, peuvent représenter un risque en cas d'utilisation par l'Administration. Parmi les licences à risque figure la licence Artistic. Bien qu'elle soit reconnue par l'OSI, cette licence fait l'objet d'un vif débat au sein de la communauté du libre. Sa particularité est de chercher la protection des droits de paternité et d'intégrité de l'auteur du logiciel. Elle suscite une polémique, car elle comporte des termes considérés ambigus, contradictoires et difficiles à interpréter. Bien que sa validité soit maintenant officiellement reconnue aux États-Unis, par suite de l'affaire Jacobsen v. Katzer, il n'est pas certain que le même résultat prévaudra dans d'autres juridictions. En conséquence, ce facteur de risque doit être considéré lors de la sélection d'un logiciel libre et ouvert soumis à la licence Artistic, comme Perl. Une utilisation ne devrait être envisagée que lorsque le logiciel apporte des bénéfices supérieurs aux incertitudes juridiques liées à sa licence.

Analyse  
contextuelle

[205] Cet exemple démontre que toutes les licences ne sont pas équivalentes et que certaines peuvent comporter des termes inacceptables pour l'Administration. Dès que l'Administration envisage l'utilisation d'un logiciel soumis à une licence ne faisant pas partie du tableau précédent, une lecture approfondie du texte de la licence s'impose.

[206] Évidemment, il peut survenir des situations où les besoins techniques de l'Administration ont préséance sur les enjeux juridiques suscités par une licence. Aussi, les critères et les directives énoncés précédemment doivent servir de guides, mais ne constituent pas des principes absolus.

Références

APACHE SOFTWARE FOUNDATION, "Apache License, Version 2.0", 2004, Web : <<http://www.apache.org/licenses/LICENSE-2.0>>

FREE SOFTWARE FOUNDATION, "GNU General Public licence", 2011, Web : <<http://www.gnu.org/copyleft/gpl.html>>

Jacobsen v. Katzer, (2008) 535 F. 3d 1373, Web : <[http://scholar.google.ca/scholar\\_case?case=17776182574171214893&hl=en&as\\_sdt=25&as\\_vis=1](http://scholar.google.ca/scholar_case?case=17776182574171214893&hl=en&as_sdt=25&as_vis=1)>

OPEN SOURCE INITIATIVE, "The BSD License", 2011, Web : <<http://www.opensource.org/licenses/bsd-license>>

OPEN SOURCE INITIATIVE, "The Open Source Definition (Annotated)", 2011, Web : <<http://www.opensource.org/docs/definition.php>>

OPEN SOURCE INITIATIVE, "The MIT License : Licensing", 2011, Web : <<http://www.opensource.org/licenses/mit-license>><<http://www.opensource.org/licenses/bsd-license>>

PERL, "The Artistic licence", 2011, Web : <<http://dev.perl.org/licenses/artistic.html>>

## 5.2. Conditions et restrictions

Principe général

[207] L'utilisation de logiciels libres et ouverts n'est généralement soumise qu'à trois conditions. Celles-ci établissent trois obligations pour le licencié :

1. Reconnaître que le donneur de licence n'offre aucune garantie;
2. Reconnaître que le donneur de licence n'assume aucune responsabilité;
3. Ne pas retirer les étiquettes faisant état des droits du donneur de licence sur le logiciel.

[208] Il est vrai que toute infraction à ces obligations pourra difficilement être constatée par le donneur de licence si le logiciel n'est jamais redistribué. Elles trouvent tout de même application à l'Administration dès qu'elle acquiert un logiciel libre et ouvert.

Garanties

[209] Toutes les licences de logiciels libres et ouverts ont en commun une clause d'exclusion de garantie. Cette clause vise d'abord à spécifier qu'aucune garantie expresse n'est fournie par le donneur de licence. Ceci revient à dire que le logiciel est fourni tel quel. Elle vise ensuite à exclure l'application de toute garantie, implicite ou légale, qui pourrait être imposée aux parties par la loi. Comme il a été souligné dans une section précédente de ce guide, cette exclusion ne sera effective que si la loi autorise les parties à renoncer à

l'avance à la garantie concernée.

*Engagements du fournisseur*

[210] Malgré tout, les clauses d'exclusion incluses dans les licences libres et ouvertes n'empêchent pas les fournisseurs d'offrir des services à valeur ajoutée. À titre d'exemple, plusieurs entreprises importantes avaient mis sur pied des fonds d'indemnisation en réaction à la poursuite de SCO envers les utilisateurs de Linux. Ces engagements particuliers peuvent contribuer à réduire certains risques associés à l'utilisation des logiciels libres et ouverts. Aussi, l'Administration ne devrait pas hésiter à y avoir recours, lorsqu'elle le juge utile.

*Divulgateion des droits*

[211] Finalement, un grand nombre de licences libres et ouvertes imposent aux licenciés de ne pas retirer les étiquettes faisant état des droits du donneur de licence sur le logiciel. Ces étiquettes peuvent se trouver dans le code source du logiciel, dans sa documentation, dans ses scripts d'installation et dans certaines de ses interfaces (particulièrement sous l'onglet « *about* ». Cette obligation n'impose aucune charge à l'Administration et devrait être aisée à respecter. Pour cette raison, l'Administration devrait agir de la sorte en toute situation, peu importe les termes de la licence concernée.

*Conditions supplémentaires*

[212] Évidemment, les conditions et les restrictions énumérées ci-dessus tiennent uniquement compte des licences libres et ouvertes dites « classiques ». Compte tenu de la diversité qui existe en matière de licences libres et ouvertes, une lecture attentive des termes de chaque licence s'impose. Les licences peu utilisées ou peu connues sont susceptibles d'inclure des conditions supplémentaires. L'objectif est donc d'éviter d'engager la responsabilité de l'Administration pour le non-respect des conditions qui y sont prévues, mais également d'éviter la publicité négative liée à un tel événement.

*Registre*

[213] Afin de réduire ce risque au strict minimum, il est recommandé de tenir un registre des logiciels libres utilisés dans chaque ministère et organisme. En plus de fournir une liste des logiciels libres auxquels l'Administration a recours, ce catalogue devrait inclure de l'information sur le type d'utilisation qui en est faite et la licence à laquelle chacun d'eux est soumis. Le gouvernement du Québec est présentement engagé dans la production d'un registre équivalent pour l'ensemble de l'Administration.

*Références*

LEMYRE (P.-P.), « Mémoire - Les logiciels libres sous l'angle de la responsabilité civile », Juriscom, 2003, Web : <<http://www.juriscom.net/uni/visu.php?ID=106>>

## 6. Adaptation et intégration

[214] Ce chapitre aborde les questions découlant de l'adaptation et de l'intégration d'un logiciel libre et ouvert par l'Administration, soit la modification de son code source ou sa fusion avec celui d'autres projets dans l'espoir qu'il réponde mieux à des besoins spécifiques. L'exercice de ce droit inhérent aux logiciels libres et ouverts requiert la considération de problématiques supplémentaires, dont la compatibilité entre les licences des logiciels concernés et l'importance d'identifier adéquatement le code appartenant à l'Administration.

### 6.1. Conditions et restrictions

#### *Principe général*

[215] Les mêmes critères de sélection devraient être appliqués aux logiciels libres et ouverts voués à faire l'objet d'adaptation et d'intégration qu'à ceux devant simplement être installés et utilisés tels quels. En conséquence, les mêmes licences sont à considérer et les conditions et les restrictions détaillées plus haut demeurent pertinentes. Cependant, dès qu'il est question de modifier le code source d'un logiciel libre et ouvert, ou de le fusionner avec un code provenant d'une autre origine, des conditions et des restrictions supplémentaires doivent être prises en compte.

#### *Mention des modifications*

[216] La principale condition commune à un grand nombre de licences libres et ouvertes consiste à exiger du licencié qu'il signale clairement les modifications qu'il a effectuées au logiciel. Cette condition vise à éviter que le code source original du logiciel ne soit confondu avec le code source modifié ultérieurement. Elle facilite également le partage des droits sur le code source en résultant. Ainsi, l'Administration, lorsqu'elle développe une solution technologique adaptée sur la base de code libre, se doit d'identifier le fruit de ses travaux adéquatement. Plusieurs licences précisent que cette obligation doit être remplie en ajoutant une mention à cet effet dans chaque fichier modifié. Les techniques à privilégier à ce sujet sont étudiées en détail un peu plus loin.

#### *Compatibilité des licences*

[217] L'intégration de code source provenant de plus d'un logiciel distinct dans un seul et même projet pose également la question de la compatibilité de leurs licences respectives. Chacune étant susceptible de préciser des conditions et des restrictions contradictoires, la fusion de code source dans ces conditions pourrait mener l'Administration à devoir enfreindre les termes de l'une ou de l'autre. Cette situation doit évidemment être évitée. Il est vrai que la plupart des licences libres et ouvertes n'empêchent pas l'Administration d'agir de la sorte pour répondre à un besoin interne, mais restreignent plutôt la distribution de ce code source. Cependant, afin de réduire les risques d'infractions qui résulteraient de sa propagation, il est recommandé de ne jamais fusionner de code provenant de logiciels dont les conditions des licences, qu'elles soient libres ou propriétaires, sont incompatibles. Les éléments à considérer pour déterminer la compatibilité de deux licences sont abordés un peu plus loin.

#### *Politique d'intégration*

[218] Afin de réduire les risques de tels conflits, une politique d'intégration devrait être définie au niveau de chaque ministère ou organisme engagé dans le développement de logiciels s'appuyant sur des bases libres et ouvertes. Cette politique vise à guider le personnel concerné lors de l'intégration de codes sources provenant de différentes sources. Elle peut réitérer l'importance de respecter les termes des licences et signaler les licences et les critères de sélection à considérer, compte tenu des infrastructures en place, ainsi que les techniques à privilégier pour le stockage du code source.

Importance de la sensibilisation

[219] Il est important que toutes les personnes susceptibles d'interagir avec le code source des logiciels de l'Administration soient informées et comprennent les implications de l'intégration de logiciels soumis à des licences incompatibles. Lorsqu'une politique d'intégration est établie, celle-ci devrait leur être rappelée, de façon régulière.

Types d'intégration

[220] De cette question découle également la nécessité de déterminer adéquatement ce qui constitue une intégration de codes sources distincts. Évidemment, la fusion de codes à l'étape de la programmation laisse peu de place au doute. Par contre, l'utilisation de bibliothèques de programmes dont les licences sont incompatibles pose des questions supplémentaires. La prudence requiert donc que les règles de compatibilité des licences soient appliquées à cette situation. Finalement, la création d'une compilation, comme une distribution assemblant divers logiciels libres et ouverts sur le même cédérom, constitue la situation la moins problématique. La compilation ne pouvant être considérée comme une version modifiée des logiciels originaux, la compatibilité de leurs licences n'est pas pertinente. Il n'en demeure pas moins que les compilations elles-mêmes peuvent être soumises à des licences et que ces dernières doivent être considérées.

Distribution ultérieure

[221] La compatibilité des licences est liée de près à la distribution du code intégré, car c'est à ce moment qu'une infraction peut être constatée. Comme il est précisé plus loin, si c'est l'Administration québécoise qui est titulaire des droits, la distribution de logiciels devrait être effectuée sous les termes de sa propre licence libre et ouverte. Dans la mesure où des logiciels libres et ouverts préexistants y sont intégrés, leur compatibilité avec cette licence doit alors être évaluée. Il en résulte que les logiciels voués à être modifiés puis redistribués doivent faire l'objet d'une attention toute particulière.

Conditions supplémentaires

[222] Encore une fois, il est possible que certaines licences libres et ouvertes contiennent des clauses particulières imposant des conditions et des restrictions supplémentaires à l'adaptation et à l'intégration des logiciels auxquels elles s'appliquent. Pour cette raison, il demeure important d'examiner à chaque occasion les termes spécifiques des licences logicielles impliquées.

Références

MEEKER (H. J.), *The Open Source Alternative : Understanding Risks and Leveraging Opportunities*, Hoboken, Wiley, 2008

SECRÉTARIAT DU CONSEIL DU TRÉSOR, « Politique-cadre sur la gouvernance et la gestion des ressources informationnelles des organismes publics », 2010, Web : <[http://www.msg.gouv.qc.ca/documents/ministere/politique\\_cadre.pdf](http://www.msg.gouv.qc.ca/documents/ministere/politique_cadre.pdf)>

## 6.2. Identification du code source

Problématique

[223] Un grand nombre de licences libres et ouvertes précisent qu'en cas de modification du code source couvert, le licencié se doit d'inclure une mention claire à cet effet dans chaque fichier modifié. Dans certains cas, l'inclusion de la date de la modification est également requise. Au-delà de la contrainte posée par ces licences, une identification adéquate du code source est à l'avantage de l'Administration qui peut ainsi départager aisément le code sur lequel elle détient des droits de celui qui appartient à des tiers. Ceci s'avère particulièrement crucial au moment d'établir une stratégie de distribution. Dans ces circonstances, l'application des techniques qui suivent, dès le stade de la programmation, simplifie énormément la gestion subséquente des logiciels libres et ouverts.

Étiquette

[224] Lors de toute modification au code source d'un logiciel libre et ouvert, une étiquette devrait être ajoutée dans le haut de chacun des fichiers modifiés. Cette étiquette peut contenir plusieurs éléments dont :

- ✦ Le nom du projet concerné;
- ✦ Le nom du fichier;

- ✦ Le nom du ministère ou de l'organisme responsable;
- ✦ Le nom du programmeur ayant effectué les travaux;
- ✦ La date de la modification;
- ✦ La licence applicable.

[225] Compte tenu des environnements de développement disponibles aujourd'hui, l'insertion de cette étiquette peut aisément être automatisée.

<i>Modularité</i>	[226] En plus de l'annotation des fichiers concernés par les modifications, le recours à des techniques de développement modulaires peut également contribuer à faciliter l'identification du code source d'une solution logicielle. Ceci est réalisé en détachant les fonctions développées pour les fins spécifiques de l'Administration de celles faisant partie du logiciel original.
<i>Fourches</i>	[227] En plus de permettre de formuler clairement les droits grevant le code, cette modularité permet d'éviter l'établissement de fourches et facilite les mises à jour subséquentes du logiciel. Une fourche survient lorsque le code source modifié devient techniquement incompatible avec le code original, créant ainsi deux projets distincts. Il devient alors difficile, voire impossible, pour les utilisateurs du nouveau projet de bénéficier des mises à jour du projet original, qu'il s'agisse de nouvelles fonctionnalités, de débogages ou de corrections de failles de sécurité. Cette situation devrait être évitée dans la mesure du possible.
<i>Contrats de services</i>	[228] Évidemment, le recours à ces techniques par les seuls programmeurs de l'Administration ne suffit pas si des programmeurs externes sont associés au processus de développement; elles devraient par ailleurs être requises des fournisseurs dans le cadre de leurs contrats de services.
<i>Références</i>	WIKIPEDIA, "Fork (Software Development)", 2011, Web : < <a href="http://en.wikipedia.org/wiki/Fork_(software)">http://en.wikipedia.org/wiki/Fork_(software)</a> >

### 6.3. Compatibilité des licences

<i>Problématique</i>	[229] L'approche choisie pour aborder la propriété intellectuelle par les licences libres et ouvertes cause un certain nombre de difficultés lorsque vient le temps d'intégrer du code provenant de projets distincts. À ce propos, plutôt que de réinventer la roue chaque fois qu'un problème connu ressurgit, le modèle de développement collaboratif des logiciels libres et ouverts encourage l'Administration à réutiliser, autant que possible, le code existant. Pourtant, les clauses contradictoires que contiennent certaines licences posent de sérieuses limites à ces combinaisons. Une analyse s'impose donc.
<i>Domaine public</i>	[230] L'intégration d'un logiciel libre et ouvert avec un logiciel faisant partie du domaine public et dont l'auteur a renoncé à ses droits moraux constitue la situation la plus simple. Aucune condition n'étant rattachée à ce dernier, il peut donc être combiné avec n'importe quel logiciel, peu importe la licence sous laquelle celui-ci est distribué. Le produit final de cette opération sera généralement soumis à la licence libre et ouverte grevant l'autre partie du code initial.
<i>Logiciels propriétaires</i>	[231] La situation est un peu plus complexe, lorsque l'un des logiciels visés est soumis à une licence propriétaire. Dans ce cas, les conditions restrictives de la licence propriétaire interdiront le plus souvent toute modification du code, et donc toute intégration avec un logiciel libre et ouvert. Dans l'hypothèse où cela serait permis, lorsque l'intégration est autorisée par le titulaire des droits, par exemple, les licences libres et ouvertes posent des contraintes différentes, selon qu'elles utilisent le « copyleft » ou non. Tout d'abord, les

licences libres et ouvertes sans « copyleft » étant extrêmement permissives, rien ne s'oppose à ce que le code leur étant soumis soit intégré à l'intérieur d'un logiciel propriétaire, et cela tant que leurs conditions sont respectées. Le logiciel Acrobat Reader d'Adobe est un excellent exemple : son fichier d'informations juridiques reproduit une multitude de licences libres et ouvertes, comme ces dernières le requièrent. À l'opposé, les licences libres et ouvertes avec « copyleft » sont toujours incompatibles avec les licences propriétaires, puisque le logiciel qui résulterait d'une telle combinaison serait nécessairement soumis à des contraintes supplémentaires.

*Licences libres et  
ouvertes sans  
« copyleft »*

[232] Les licences libres et ouvertes sans « copyleft », pour leur part, sont généralement compatibles les unes avec les autres, mais également avec tous les autres types de licences. En effet, les plus simples (BSD 2.0, MIT) ne contiennent aucune clause pouvant créer des interactions avec une autre licence. Cependant, les licences libres et ouvertes sans « copyleft » plus élaborées peuvent parfois entrer en conflit avec certaines licences ayant recours au « copyleft ». Ainsi, la licence Apache 2.0 est incompatible avec la GPL 2.0 à cause des conditions additionnelles qu'elle impose, particulièrement en ce qui a trait à la suspension des brevets logiciels. Elle est cependant compatible avec la GPL 3.0, puisque cette dernière intègre des conditions équivalentes.

*Licences libres et  
ouvertes avec  
« copyleft »*

[233] En ce qui concerne les licences libres et ouvertes avec « copyleft », elles ont justement comme particularité d'anticiper leur application à tout logiciel dérivé du code leur étant initialement soumis. Il en résulte que les licences libres et ouvertes avec « copyleft » sont totalement incompatibles avec toute licence plus restrictive, mais également avec toutes les autres licences avec « copyleft ». Cela est vrai même lorsque deux licences avec « copyleft » accordent exactement les mêmes droits, puisque chacune d'entre elles requiert que le logiciel résultant soit soumis à ses propres clauses. Il est donc uniquement possible d'intégrer du code avec « copyleft » avec du code provenant du domaine public ou soumis à une licence libre et ouverte sans « copyleft ». La seule exception notable est la possibilité de migrer du code soumis à la GPL 2.0 vers la GPL 3.0 lorsque le donneur de licence a précisé dans son étiquette que le licencié est en mesure de redistribuer sous les termes de toute version subséquente de la licence.

*Exceptions*

[234] Quelques licences libres et ouvertes avec « copyleft » contiennent cependant des exceptions à ce principe. C'est le cas de la Mozilla Public License (MPL) qui différencie le code initial du logiciel du code provenant de contributions subséquentes, permettant l'attribution d'une autre licence aux modifications apportées aux contributions. Dans un tel contexte, il s'avère donc possible de respecter les termes d'une autre licence libre et ouverte avec « copyleft ». Toutefois, l'identification adéquate du code assujéti à cette exception demande beaucoup de prudence.

Tableau 9 – Compatibilité entre les principales licences

De \ Vers	<b>Domaine public</b>	<b>CLLAQ</b>	<b>GPL 2.0</b>	<b>GPL 3.0</b>	<b>Apache 2.0</b>	<b>BSD 2.0</b>	<b>MIT</b>	<b>Propriétaire</b>
<b>Domaine public</b>	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui
<b>CLLAQ</b>	Non	Oui	Oui	Oui	Oui	Oui	Oui	Oui
<b>GPL 2.0</b>	Non	Non	Oui	Oui	Non	Non	Non	Non
<b>GPL 3.0</b>	Non	Non	Non	Oui	Non	Non	Non	Non
<b>Apache 2.0</b>	Non	Oui	Non	Oui	Oui	Oui	Oui	Oui
<b>BSD 2.0</b>	Non	Oui	Oui	Oui	Oui	Oui	Oui	Oui
<b>MIT</b>	Non	Oui	Oui	Oui	Oui	Oui	Oui	Oui
<b>Propriétaire</b>	Non	Non	Non	Non	Non	Non	Non	Non

*Application mécanique*

[235] Il n'en demeure pas moins que les principes décrits ci-dessus ne devraient pas être appliqués mécaniquement. En effet, les licences qui y sont spécifiées sont souvent utilisées comme des modèles afin d'en élaborer de nouvelles, dont les termes s'éloignent plus ou moins de l'original. Dans ces conditions, la prise en compte des termes exacts de chacune des licences visées est requise.

*Doubles concessions*

[236] Cependant, à cause des contraintes qu'entraînent ces conflits entre les licences, certains logiciels libres et ouverts sont offerts sous plus d'une licence. Dans une telle situation, il s'avère possible de choisir la licence la mieux adaptée aux besoins d'intégration du moment. Cette dualité de licence est rendue possible par la nature même des licences de logiciels, concédant uniquement un droit non exclusif de poser certains gestes qui autrement seraient interdits. De plus, il est toujours envisageable d'obtenir du titulaire de droits la permission de déroger aux termes de sa licence libre et ouverte. En pratique, ces

requêtes spéciales sont souvent acceptées, les développeurs de logiciels libres et ouverts ayant tout intérêt à ce que leurs logiciels soient utilisés.

*Compilations*

[237] Les compilations de logiciels libres et ouverts posent des difficultés supplémentaires lorsqu'il s'agit d'évaluer la compatibilité des termes de leurs licences. Tout d'abord, compte tenu de la structure des licences libres et ouvertes, les logiciels de ce type assemblés au sein d'une même compilation n'ont pas à respecter les termes de licences compatibles. En effet, tant que les différents logiciels de la compilation demeurent indépendants, c'est-à-dire qu'ils ne sont pas intégrés les uns aux autres, leur distribution peut aisément s'effectuer dans la mesure où leurs licences respectives les accompagnent. Toutefois, à ces licences s'ajoute maintenant une licence sur la compilation en elle-même. De plus, l'intégration de multiples compilations est sujette à l'ensemble des conflits de licences exposés ci-dessus. Par ailleurs, il demeure toujours possible de sortir un ou plusieurs éléments d'une compilation, voire de les intégrer à ceux d'autres compilations. Dans un tel cas, l'originalité de cette nouvelle compilation ne requiert pas le respect des licences des compilations antérieures.

*Bibliothèques de programmes*

[238] Enfin, les bibliothèques de programmes font également l'objet de particularités, lorsqu'il est question d'intégration de logiciels libres et ouverts. Découlant de la possibilité qu'une bibliothèque de programmes soit soumise à une licence différente de celle du logiciel principal, ceci est particulièrement problématique dans le cas des licences libres et ouvertes avec « copyleft ». Si la GPL 3.0 clarifie désormais que toutes les bibliothèques pour les systèmes n'ont pas à être soumises à la même licence, ce n'est pas le cas de la GPL 2.0. Aussi, la solution la plus avantageuse pour de telles bibliothèques est certainement de s'en remettre à des licences sans « copyleft » ou à la LGPL, version de la GPL qui ne possède pas son caractère viral.

FREE SOFTWARE FOUNDATION, "Various Licenses and Comments about Them", 2011, Web : <<http://www.gnu.org/licenses/license-list.html>>

OPEN SOCIETY INITIATIVE, "Open Source Licenses by Category", 2011, Web : <<http://www.opensource.org/licenses/category>>

ROSEN (L.), *Open Source Licensing : Software Freedom and Intellectual Property Law*, Prentice Hall PTR, Upper Saddle River, 2004

ST. LAURENT (A. M.), *Understanding Open Source & Free Software Licensing*, O'Reilly, Sebastopol, 2004

WHEELER (D. A.), "Make Your Open Source Software GPL-Compatible. Or Else", 2010, Web : <<http://www.dwheeler.com/essays/gpl-compatible.html>>

## 7. Distribution

[239] Ce chapitre aborde les questions découlant de la distribution d'un logiciel libre et ouvert par l'Administration, c'est-à-dire sa mise à disposition, publique ou privée, par quelque moyen que ce soit. Cette distribution concerne autant les logiciels libres et ouverts préexistants ayant été adaptés aux besoins spécifiques de l'Administration que les logiciels dont le code source lui appartient intégralement. Dans les deux cas, il est suggéré d'avoir recours à une licence uniformisée intitulée Convention de licence libre de l'Administration québécoise (CLLAQ).

### 7.1. Distributions internes

#### *Problématique*

[240] La distribution d'un logiciel libre et ouvert par l'Administration peut d'abord prendre la forme d'une distribution entre les ministères et les organismes du gouvernement du Québec. Dans ce scénario, l'un d'eux a acquis une solution technologique basée sur des logiciels libres et ouverts, l'a améliorée pour qu'elle réponde mieux à ses besoins particuliers, et souhaite la partager avec d'autres entités de l'Administration. En théorie, de tels transferts technologiques ne devraient nécessiter aucune licence, les multiples branches de l'Administration étant considérées comme une seule et même personne. En pratique, les limites de la personnalité juridique de l'Administration sont parfois difficiles à déterminer.

#### *Transfert entre entités administratives*

[241] Afin de simplifier et d'unifier la procédure, il est donc recommandé d'avoir recours à la Convention de licence libre de l'Administration québécoise (CLLAQ), même lorsque la distribution implique un transfert de technologie entre deux entités administratives. En agissant de la sorte, le respect des conditions et des restrictions imposées par les licences antérieures est plus certainement assuré. De plus, le recours à la CLLAQ assure le respect du Cadre de gestion et de valorisation de la propriété intellectuelle selon lequel les ministères et les organismes devraient favoriser entre eux l'accès aux actifs de propriété intellectuelle dont ils sont titulaires, et cela sans contrepartie.

#### *Transfert par le fournisseur*

[242] Si le transfert de technologie entre deux entités administratives distinctes pose des difficultés, il est toujours possible pour la seconde entité de contracter elle-même avec le fournisseur de la solution améliorée. En pareil cas, le fournisseur concerné devrait proposer des conditions avantageuses, étant donné que les coûts de développement ont déjà été supportés par la première entité. La licence applicable peut alors varier en fonction du fournisseur et des dispositions prévues dans le cadre de son contrat de services. Toutefois, ici encore, l'application de la CLLAQ devrait être privilégiée.

[243] Dans le même ordre d'idées, des ententes cadres peuvent être négociées avec le fournisseur afin de prévoir la réutilisation de la solution en question par plusieurs branches de l'Administration.

#### *Référence*

DÉVELOPPEMENT ÉCONOMIQUE, INNOVATION ET EXPORTATION QUÉBEC, « Cadre de gestion et de valorisation de la propriété intellectuelle résultant des activités scientifiques et techniques des ministères et les organismes publics du gouvernement du Québec », 2010, Web : <<http://www.mdeie.gouv.qc.ca/index.php?id=6156>>

## 7.2. Distributions externes

- Problématique* [244] Lorsque l'Administration cherche à rendre un logiciel lui appartenant, ou qu'elle a grandement modifié, librement accessible au monde extérieur, la forme de distribution est tout autre. On pourra parler d'une distribution privée impliquant la remise d'une seule copie du code source sur un support physique. Il peut également s'agir de sa simple mise à disposition au public sur un serveur Web. Finalement, l'Administration peut elle-même chercher à créer une communauté susceptible de l'aider à maintenir et à faire évoluer le logiciel dans le futur. Cette dernière possibilité requiert la mise en place d'une infrastructure adéquate, la disponibilité d'un minimum de financement destiné à cet effet ainsi que la prise en charge de l'encadrement des volontaires. Dans tous les cas, le recours à la CLLAQ devrait être privilégié.
- Propriété intellectuelle* [245] Avant de procéder à toute distribution externe, l'Administration devrait mener une réflexion approfondie sur sa stratégie en matière de gestion de la propriété intellectuelle. Cette réflexion devrait entre autres porter sur les paramètres de la convention à intervenir et sur l'opportunité d'exiger une contrepartie financière. Une telle mesure de protection vise à éviter la distribution en mode libre de logiciels, qui aurait autrement permis à l'Administration de percevoir des redevances liées à sa propriété intellectuelle sur ceux-ci. Toutefois, dans bien des cas, la distribution sous les termes de CLLAQ sera la seule solution valable.
- Garantie et responsabilité* [246] Une fois la décision prise et le logiciel distribué sous les termes de la CLLAQ, une relation contractuelle s'établit entre l'Administration, à titre de donneur de licence, et le licencié. Cette relation n'apporte pas pour autant une grande charge à l'Administration. Tout comme, en principe, elle ne bénéficie d'aucune garantie à titre d'utilisatrice d'un logiciel libre et ouvert, elle n'est pas tenue elle-même de garantir le logiciel qu'elle distribue. De la même façon, les licenciés ne devraient pas être en mesure d'impliquer sa responsabilité civile pour les dommages subis à cause du logiciel.
- Assistance technique* [247] Il en va de même en ce qui concerne l'assistance technique attendue de la part de l'Administration. Le logiciel peut très bien être distribué sans pour autant que celle-ci s'engage à poursuivre son développement, à résoudre les bogues et les failles de sécurité ou à répondre aux questions de ses usagers éventuels. Cependant, si le logiciel joue un rôle important à l'interne, l'Administration peut avoir intérêt à investir des ressources pour gérer les interactions avec ces derniers. Un logiciel qui suscite de l'intérêt est susceptible d'entraîner la naissance d'une communauté dont l'apport pourrait rapidement surpasser les efforts que l'Administration est elle-même en mesure de générer. Dans une telle situation, les contributions reçues de la communauté ont une plus grande valeur que les ressources requises pour fournir l'assistance technique.
- Documentation* [248] Plusieurs logiciels libres et ouverts sont fournis avec peu de documentation. Ceci s'explique d'abord par le manque d'intérêt commercial des éditeurs de logiciels libres et ouverts à investir dans sa rédaction. Ensuite, les utilisateurs s'attendent le plus souvent à trouver des réponses à leurs questions auprès de la communauté. Aussi, l'absence de documentation ne devrait pas justifier de ne pas distribuer un logiciel. Si le logiciel suscite de l'intérêt et qu'une communauté d'utilisateurs se développe, il est probable que certains d'entre eux s'investiront dans la rédaction de documentation. Celle-ci, loin d'être un élément essentiel au moment de la phase initiale de développement d'un projet libre, fait généralement son apparition lors de sa phase d'organisation.
- Non-respect* [249] Finalement, la distribution d'un logiciel sous les termes de la CLLAQ soulève également la question de son respect par les licenciés. Bien que ses termes soient extrêmement permissifs, il est toujours possible qu'un utilisateur en abuse. Cet abus prendra généralement la forme d'une redistribution du code couvert sans accompagnement du texte de la licence. En pareil cas, les droits d'auteur de l'Administration, ainsi que les clauses d'exclusion de garantie et de responsabilité ne seraient pas divulgués aux

utilisateurs subséquents. L'Administration devrait alors voir à contacter l'utilisateur fautif. Dans bien des cas, le non-respect résultera d'un oubli ou d'une incompréhension. Toutefois, si la situation persiste, la direction des affaires juridiques du ministère ou de l'organisme concerné devrait être avisée. Elle verra alors à prendre les mesures appropriées, selon les circonstances, afin d'assurer l'exécution des dispositions de la licence.

*Matériel* [250] Il est possible que du logiciel soumis à un titre de propriété intellectuelle de l'Administration soit intégré à un produit matériel lui-même destiné à être vendu. Bien qu'une copie du logiciel soit alors remise à chaque acheteur, il existe une certaine controverse à savoir si cette opération constitue une distribution. CISCO a prétendu que non dans un litige qui s'est finalement réglé hors cour aux États-Unis. En France, Iliad, la maison mère de Free.fr, est impliquée dans un conflit similaire concernant la FreeBox, le modem qu'elle distribue à sa clientèle. En pareille circonstance, l'Administration ne devrait pas hésiter à faire valoir ses droits comme pour tout autre type de distribution illicite.

*Références* BIREBENT (V.), « Free ouvre une partie du code source de sa Box », ZDNET.fr, 2008, Web : <<http://www.zdnet.fr/actualites/free-ouvre-une-partie-du-code-source-de-sa-box-39384326.htm>>

BROW (E.), "Cisco settles with FSF on GPL violations", LinuxDevices.com, 2009, Web : <<http://www.linuxfordevices.com/c/a/News/Cisco-settles-with-FSF-on-GPL-violations/>>

FOGEL (K.), *Produire du logiciel libre*, Traduction Framalang, In Libro Veritas, 2010, Web : <[http://www.framabook.org/Produire\\_du\\_logiciel\\_libre.html](http://www.framabook.org/Produire_du_logiciel_libre.html)>

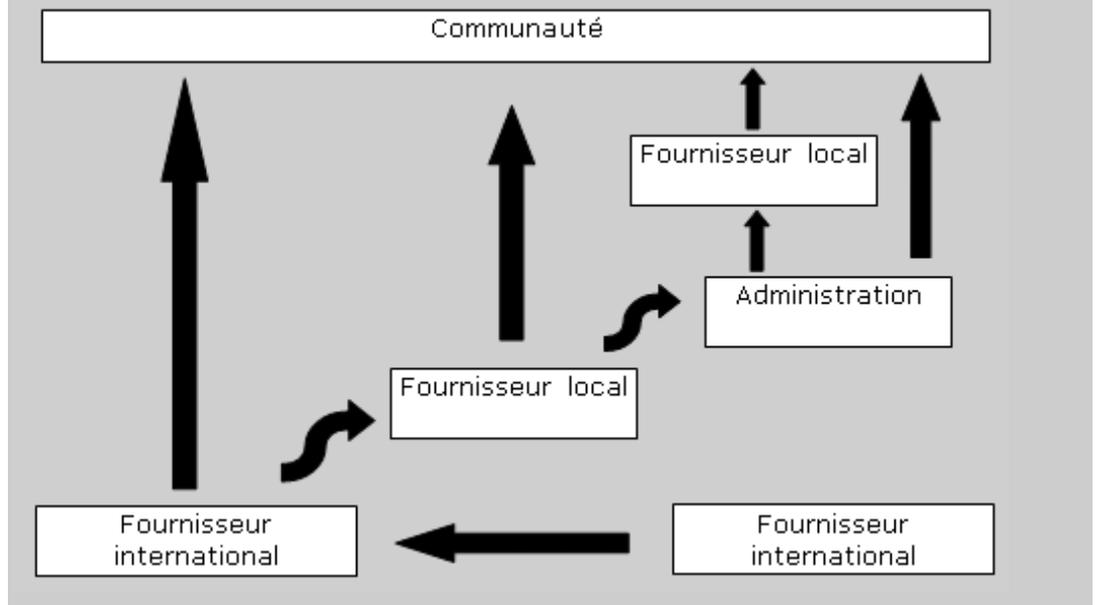
PUBLICATIONS DU QUÉBEC, « Normes en matière d'acquisition, d'utilisation et de gestion de droits d'auteurs des documents détenus par le gouvernement, les ministères et les organismes publics désignés par le gouvernement », Gazette officielle du Québec, 25 octobre 2000, 132<sup>e</sup> année, n<sup>o</sup> 43, p. 6753-6757 et Décret 12-2001, Gazette officielle du Québec, 31 janvier 2001, 133<sup>e</sup> année, n<sup>o</sup> 5, p. 1172-1173, Web : <<http://www.droitauteur.gouv.qc.ca/images/normesauteur.pdf>>

SCOTT (B.), "Why the State Should Release its Software as Open Source", Open Source Law, 2004, Web : <<http://www.opensourcelaw.biz/papers/BScottGovAccessRegimes040519.pdf>>

### 7.3. Contributions aux communautés

*Problématique* [251] En plus de la distribution de ses propres logiciels, l'Administration est en mesure de contribuer aux communautés des logiciels libres et ouverts. Ces contributions prennent généralement la forme de rustines, petits programmes qui servent à corriger des erreurs, permettant d'améliorer la sécurité ou le fonctionnement du logiciel. Parfois, elles peuvent être plus substantielles et consister dans l'apport d'une nouvelle fonctionnalité ou d'un nouveau module. Évidemment, il n'y a aucune garantie que ces contributions de l'Administration seront acceptées par la communauté et intégrées au code source du logiciel concerné. Cependant, en agissant de la sorte, l'Administration assure sa participation et son droit de regard à l'égard de l'évolution du logiciel.

Tableau 10 – Schéma des contributions aux communautés du libre



*Avantages*

[252] Plusieurs avantages découlent de l'établissement d'un partenariat étroit avec la communauté chargée de superviser l'évolution d'un logiciel. Tout d'abord, ceci permet d'assurer une uniformité entre la version utilisée par l'Administration et la version officielle du logiciel. De cette façon, les fourches sont évitées et l'Administration peut continuer de bénéficier des mises à jour fournies par le reste de la communauté. En soumettant ses apports, l'Administration s'assure également que ceux-ci seront révisés par des pairs et testés sous différents environnements; son code pourra alors atteindre un niveau de maturité et de sécurité adéquat plus rapidement. Finalement, l'appui ainsi fourni à la communauté contribue à renforcer l'image de l'Administration dans ce milieu, élément non négligeable étant donné l'appui réciproque attendu.

*Rustines et débogage*

[253] Dans la vaste majorité des cas, les contributions visées ne sont constituées que de quelques lignes de code dont l'introduction dans le code source du logiciel contribue à améliorer son fonctionnement. Étant donné que ces quelques lignes, prises isolément, n'ont aucune valeur intrinsèque, la gestion des droits de l'Administration sur ce code pose peu de difficulté. En fait, la licence logicielle applicable et le régime juridique qu'elle impose aux contributions importent peu. L'Administration peut alors transmettre sa contribution aussitôt qu'elle le juge opportun, sans se soucier des termes imposés par la communauté concernée.

*Contributions majeures*

[254] Toutefois, la situation est légèrement plus complexe lorsque la contribution anticipée est particulièrement importante. Ce serait le cas, par exemple, d'une contribution permettant de migrer le logiciel sous une nouvelle plate-forme, ou de celle ajoutant une fonctionnalité longuement attendue. En pareil cas, la contribution elle-même peut avoir une grande valeur, et l'Administration devrait alors se questionner sur sa stratégie de distribution. Plusieurs options peuvent être envisagées, dont la double concession de la contribution (sous les termes de la licence originale et sous les termes de la CLLAQ), la création d'un projet libre indépendant ou sa distribution contre rémunération sous les termes d'une licence propriétaire.

*Cession de droits*

[255] Par ailleurs, certaines communautés exigent une cession complète de droits de l'ensemble de leurs contributeurs. Ainsi, la FSF agit de la sorte pour les logiciels faisant

partie du projet GNU. Selon ce modèle, l'Administration se doit de renoncer entièrement à ses droits d'auteur sur le code qu'elle a elle-même développé, si elle désire le voir incorporer dans la version officielle du logiciel. Encore une fois, cette opération soulève peu de difficulté en ce qui concerne les contributions de rustines. Cependant, avant de renoncer à sa propriété intellectuelle sur une contribution substantielle, l'Administration devrait mener une réflexion approfondie quant à la stratégie qu'elle désire mettre de l'avant à ce sujet.

*Références*

FIRST MONDAY, "FM Interview with Linus Torvald : What Motivates Free Software Developers", 1998, Web : <http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/583/504>>

KEGEL (D.), "Contributing to Open Source Projects HOWTO", 2004, Web : <http://www.kegel.com/academy/opensource.html>>

METCALFE (R.), "Can you contribute code to an open source project?", *OSS Watch*, 2010, Web : <http://www.oss-watch.ac.uk/resources/contributing.xml>>

## 8. Conclusion

[256] Les logiciels libres et ouverts sont aujourd'hui omniprésents dans les architectures informatiques modernes. Au cours des dernières années, l'apparition de nouvelles approches hybrides en matière de gestion de la propriété intellectuelle et la progression fulgurante du modèle logiciel-services ont entraîné l'intégration de logiciels libres et ouverts dans une très vaste gamme de solutions. L'Administration québécoise, comme tous les grands consommateurs de logiciels, en fait déjà largement usage et doit apprendre à s'adapter aux caractéristiques de ce nouvel environnement.

[257] Dans le but d'apporter des réponses aux principales questions auxquelles sont confrontés les représentants de l'Administration lorsqu'ils interagissent avec des logiciels libres et ouverts, ce guide a abordé en détail les incompréhensions courantes sur la nature même des logiciels libres et ouverts. Il y fut ensuite démontré que malgré les particularités qui s'attachent à leur processus d'acquisition, rien ne limite l'utilisation, l'adaptation et l'intégration, ni la distribution de logiciels libres et ouverts par l'Administration. Finalement, on y propose un cadre juridique essentiel au bon déroulement des opérations menées dans ce domaine.

[258] Si les objectifs de ce guide sont atteints, l'intégration de logiciels libres et ouverts à l'infrastructure technologique de l'Administration québécoise se trouvera facilitée par une meilleure compréhension des enjeux et des problématiques qui s'y rattachent. À moyen et long terme, il est à espérer que cette démarche contribuera à la réduction des dépenses technologiques, tout en assurant l'interopérabilité des différentes branches de l'Administration et en accroissant son indépendance technologique.

## 9. Contacts

### *Logiciels libres et ouverts*

Pour tout renseignement concernant le recours aux logiciels libres et ouverts, veuillez communiquer avec :

Secrétariat du Conseil du trésor  
Sous-ministériat à l'encadrement des ressources informationnelles  
875, Grande Allée Est, 4<sup>e</sup> étage, secteur 500  
Québec (Québec) G1R 5R8  
Téléphone : 418 643-8383  
Télécopieur : 418 528-6153  
Web : [http://www.msg.gouv.qc.ca/gel/logiciel\\_libre/gouv\\_quebec.html](http://www.msg.gouv.qc.ca/gel/logiciel_libre/gouv_quebec.html)

### *Acquisition*

Pour tout renseignement concernant les processus d'acquisition de logiciels libres et ouverts, veuillez communiquer avec :

Centre de services partagés du Québec  
Direction générale des acquisitions  
875, Grande Allée Est, 4<sup>e</sup> étage, secteur 4.550  
Québec (Québec) G1R 5W5  
Téléphone : 418 643-6080  
Télécopieur : 418 528-2733  
Web : <http://www.cspq.gouv.qc.ca>

### *Propriété intellectuelle*

Pour tout avis sur la gestion de la propriété intellectuelle de l'Administration québécoise, veuillez communiquer avec :

Ministère de la Justice du Québec  
Groupe-conseil en propriété intellectuelle  
Édifice Louis-Philippe-Pigeon  
1200, route de l'Église  
Québec (Québec) G1V 4M1  
Téléphone : 418 643-5140  
Télécopieur : 418 646-4449  
Web : <http://www.justice.gouv.qc.ca>

