

MÉMOIRE

présenté

devant le jury du concours interne

d'accès au corps des Ingénieurs des Mines

session 2013

par

BENJAMIN VIDAL

Sujet du mémoire :

*Le logiciel libre dans le secteur public :
Un état des lieux en juin 2013*

philosophie, enjeux et perspectives



Remerciements

Je remercie, par ordre de mes rencontres,

- ERIC GERMAN, administrateur des finances publiques adjoint
- FABIEN POPOFF, inspecteur principal des finances publiques
- RÉMI VUILLEMOT, attaché d'administration centrale
- LAURE PATAS D'ILLIERS, attachée principale d'administration centrale
- CHRISTOPHE CAZIN, architecte logiciel au ministère de l'Intérieur
- THIERRY AIMÉ, pilote du marché de support des logiciels libre du ministère des Finances
- XAVIER GUIMARD, colonel à la Gendarmerie nationale
- JEAN-SÉVERIN LAIR, [Directeur des Systèmes d'Information](#) du ministère de la Culture
- MICHEL BROUANT, directeur général adjoint technique du [DSI](#) de Pôle emploi

d'avoir bien voulu répondre à mes questions sur le rôle qu'ils ont joué pour promouvoir le logiciel libre dans leurs administrations respectives.

Je remercie mes collègues de l'[Agence France Trésor \(AFT\)](#) pour leur soutien et tout particulièrement GILLES SCHONE, qui m'a aidé dans le choix de ce sujet, m'a encouragé et conseillé tout au long de la rédaction de ce mémoire.

Table des matières

Préambule	1
Les acteurs qui concourent à délivrer un service informatique	1
Les acteurs internes	1
Les acteurs externes	2
Introduction	3
1 Philosophie & grands principes	5
1.1 Qu'est-ce que le logiciel libre ?	5
1.1.1 Un bref historique	5
1.1.1.1 Les années 1960 : l'avant licence	5
1.1.1.2 Les années 1970 : le logiciel comme une fin en soi	5
1.1.1.3 Les années 1980 : l'apparition du concept de logiciel libre	6
1.1.2 Le cadre juridique	6
1.1.2.1 La notion de copyleft	6
1.1.2.2 Les particularités des licences libres	7
1.1.2.3 Libre ou « Open-source » ?	8
1.1.3 L'organisation d'un projet libre	8
1.1.3.1 La naissance d'un projet	8
1.1.3.2 Le déroulement de la vie d'un projet	9
1.1.3.3 L'extinction d'un projet	9
1.1.4 Exemples de solutions libres populaires	10
1.1.4.1 Les logiciels du poste de travail	10
1.1.4.2 Les logiciels du terminal mobile	10
1.1.4.3 Les logiciels « serveur »	11
1.1.4.4 Les langages de programmation	11
1.2 Le secteur public face au monde du logiciel	12
1.2.1 La notion de service public	12
1.2.1.1 De façon générale	12
1.2.1.2 Pour le logiciel en particulier	12
1.2.2 La commande publique	13
1.2.2.1 De façon générale	13
1.2.2.2 Pour le logiciel en particulier	13
1.2.3 La valorisation du patrimoine immatériel de l'État	14
1.2.3.1 De façon générale	14
1.2.3.2 Pour le logiciel en particulier	15
2 Enjeux du logiciel libre dans le secteur public	17
2.1 Du point de vue de l'acheteur public	17
2.1.1 Aspects financiers	17
2.1.1.1 Coût d'acquisition	17
2.1.1.2 Mise en concurrence du support	18
2.1.1.3 Mutualisation des coûts de développement	19

2.1.1.4	Coûts cachés	20
2.1.2	Respect du code des marchés publics	21
2.1.2.1	Liberté d'accès à la commande publique	21
2.1.2.2	Égalité de traitement	21
2.1.2.3	Roulement des titulaires	22
2.1.3	Qualité	22
2.1.3.1	Facilité d'utilisation	22
2.1.3.2	Fiabilité	23
2.1.3.3	Pérénnité	23
2.2	D'un point de vue politique	24
2.2.1	Indépendance	25
2.2.1.1	Meilleure maîtrise en interne	25
2.2.1.2	Utilisation de normes plutôt que de standards	25
2.2.1.3	Souveraineté technologique de la France	26
2.2.2	Innovation	28
2.2.2.1	Dans le monde du logiciel propriétaire	28
2.2.2.2	Dans le monde du logiciel libre	29
2.2.2.3	Dans le monde internet	29
2.2.3	Macro-économie	29
2.2.3.1	Où va l'argent du logiciel ?	29
2.2.3.2	Externalités dues aux logiciels libres	30
2.2.3.3	Effet sur les entreprises	31
3	Perspectives à court et moyen terme	33
3.1	Le libre, une tendance qui monte	33
3.1.1	Le marché du logiciel libre en croissance constante depuis 10 ans	33
3.1.1.1	État des lieux en France	33
3.1.1.2	Un modèle économique encore perfectible	34
3.1.2	Le libre se propage au delà du logiciel	35
3.1.2.1	Les inspirations les plus populaires	35
3.1.2.2	L'OpenData en marche	36
3.2	Du logiciel au nuage, la grande mutation	37
3.2.1	Les niveaux de service offerts par le <i>cloud</i>	37
3.2.1.1	Stockage en tant que Service	38
3.2.1.2	Infrastructure en tant que Service	38
3.2.1.3	Plateforme en tant que Service	38
3.2.1.4	Logiciel en tant que Service	39
3.2.2	Les solutions logicielles pour le <i>cloud</i>	39
3.2.2.1	Solutions propriétaires	39
3.2.2.2	Solutions libres	40
3.2.3	L'idée de <i>cloud</i> « souverain »	40
3.2.3.1	Cloudwatt	41
3.2.3.2	Numergy	41
	Conclusion	43
	Glossaire des termes techniques	45
	Acronymes	47
	Bibliographie	51

Table des exemples

1	Exemple d'exemple	4
2	La licence CeCIL	7
3	Logiciel libre ou Open-source	8
4	Une solution d'inventaire de parc informatique et de télédiffusion	9
5	Prestations de veille stratégique dans le marché SLL	9
6	Le projet PLUME du CNRS	10
7	La gratuité dans les services publics	12
8	Marché de support des logiciels libres	14
9	Suite bureautique dans la Gendarmerie nationale	18
10	La modernisation du SI de l'ENA	18
11	Le marché de support des logiciels libres du ministère de l'Intérieur	18
12	Difficultés techniques pour l'exécution de marché de support	18
13	Développement d'une solution de calendrier partagé libre par la Gendarmerie	19
14	Environnement Numérique de Travail à la région Île-de-France	20
15	Le projet Lutèce de la ville de Paris	20
16	L'approche pragmatique de Pôle emploi en matière de logiciel	20
17	L'obsolescence de Windows XP pousse la DGFIP à s'interroger	20
18	Le passage à LibreOffice nécessite un effort de formation	21
19	La décision du Conseil d'État concernant les marchés de prestation de service	21
20	Le support du serveur d'application JBoss	22
21	L'impact de la notoriété des marques sur les utilisateurs	22
22	Déploiement de Linux Ubuntu à la Gendarmerie nationale	23
23	Les logiciels qui « portent » le web aujourd'hui	23
24	Le rachat de SUN microsystems par Oracle	24
25	Le « Freemium », un concept à la marge du logiciel libre	24
26	Exemple de projets internes/externes à l'AFT	25
27	Standard et norme dans le monde bureautique	26
28	Le sénateur JM Bockel à propos de la cyber-défense	27
29	Le contrat controversé du ministère de la Défense avec Microsoft	27
30	Découverte de versions modifiées d'Apache comportant une porte dérobée	27
31	CLIP, le système d'exploitation sécurisé de l'ANSSI	28
32	Trustedbird, la solution de messagerie électronique sécurisée de la DGA	28
33	La virtualisation, fleuron de l'innovation propriétaire	28
34	Besoins fonctionnels couverts uniquement par le logiciel propriétaire	28
35	Wiki, une idée nouvelle venue du libre	29
36	LemonLDAP, une solution de SSO conçue par l'administration	30
37	N'importe quelle société de service peut devenir « éditeur » de logiciel	34
38	Les éditeurs de logiciel éducatifs s'inquiètent de la montée en puissance du libre	34
39	The Document Foundation recherche 1M€ pour LibreOffice en ligne	34
40	Document sous licence « Creative Commons »	35
41	Le concours « Moov' in the city » organisé par la ville de Paris	36
42	L'offre <i>cloud</i> d'Apple (STaaS)	38
43	L'offre <i>cloud</i> d'Amazon (IaaS)	38

44 L'offre <i>cloud</i> de Microsoft (PaaS)	38
45 L'offre <i>cloud</i> de Google (SaaS)	39
46 Le modèle du <i>cloud</i> « souverain » déjà mis à mal	41

Préambule

Les acteurs qui concourent à délivrer un service informatique

Un service informatique en milieu professionnel a pour finalité de répondre à un besoin de traitement automatisé de l'information. Au départ, une phase projet permet de réfléchir sur le besoin puis de mettre en œuvre une solution permettant d'y répondre. Une fois le projet installé et utilisé, celui-ci passe en phase de maintenance. Les acteurs qui concourent à ce projet ont en général des points de vue et des intérêts différents. Voici un découpage que l'on retrouve dans beaucoup d'organisation faisant apparaître les intérêts par catégories d'acteurs¹ :

1 Les acteurs internes

1.1 Les utilisateurs

Rôle : Ce sont eux qui ont un besoin fonctionnel. En règle générale, ils sont assistés de spécialistes qui les aident à définir leur besoin ou à imaginer de nouvelles méthodes de travail plus automatisées en leur proposant des solutions qu'ils ne connaissent pas forcément.

Intérêts défendus : Ils veulent une solution simple à utiliser qui fait ce qu'on lui demande et qui le fait bien. La fiabilité et les performances sont donc des critères qui comptent pour eux. La pérennité de la solution est également au cœur de leur préoccupation car chaque changement de solution représente un investissement important (formation, risque d'erreurs accru pendant la période d'adaptation, rencontre de bug ou de défaut de configuration).

1.2 Les décideurs

Rôle : C'est à eux que revient de conduire le projet initial en faisant les choix les plus judicieux. Ces choix sont de plusieurs ordres : stratégiques, fonctionnels, techniques et financiers.

Intérêts défendus : Ce sont ceux de la structure qui lance le projet. Ils doivent s'assurer que la solution choisie répond bien aux besoins fonctionnels des utilisateurs, est pérenne, ne comporte pas de coûts cachés et est maîtrisée d'un point de vue technique.

1.3 Les collaborateurs en charge du système d'information

Rôle : Ce sont les personnels internes (exploitants, experts métier) qui vont s'occuper de faire vivre le système d'information au jour le jour (aussi bien d'un point de vue métier que technique). Ils doivent s'assurer que tout fonctionne bien au quotidien, gérer les demandes d'évolutions futures, les mettre en place lorsque cela est possible ou les relayer auprès de l'éditeur si besoin. Leur rôle est également de veiller au respect des règles de sécurité.

1. Ce découpage est volontairement différent de celui qu'on retrouve le plus souvent à savoir [Maîtrise d'ouvrage \(MOA\)](#), [Maîtrise d'œuvre \(MOE\)](#), Assistance à [MOA](#), [MOE](#) déléguée. Il est issu de mon expérience professionnelle à l'[AFT](#) en tant que responsable informatique enrichie de la vision que j'ai eu du programme Copernic de la [DGFIP](#). Il peut parfois être différent (notamment dans la répartition interne/externe) en fonction de la taille et de la nature du projet.

Intérêts défendus : Ils sont attachés à la fiabilité globale du système. En effet, un système peu fiable les exposera à des plaintes régulières des utilisateurs. Les aspects sécurité sont également importants afin d'éviter toute intrusion dans le système d'information qui porterait préjudice à la structure.

2 Les acteurs externes

2.1 Le propriétaire du logiciel

Rôle : C'est lui qui assure le développement du logiciel qui servira de base au service informatique à mettre en place. Il s'agit d'un éditeur dans le cas de logiciel propriétaire, et d'une communauté dans le cas du logiciel libre. Ce logiciel est parfois nommé **progiciel** pour exprimer le fait qu'il s'agit d'un produit « prêt-à-l'achat ».

Intérêts défendus : Dans le cas d'un éditeur de logiciel, celui-ci cherche à vendre des licences de son produit. Dans le cas d'une communauté, celle-ci a intérêt à accroître le nombre d'utilisateurs de son logiciel afin que la communauté soit la plus active possible (afin d'en assurer la pérennité notamment).

N.B. : Il arrive fréquemment que le besoin exprimé ne soit couvert par aucun logiciel du marché (propriétaire ou libre). Dans ce cas, le projet donnera lieu à un **développement spécifique** à ce besoin. La propriété du logiciel résultant de ce développement dépendra alors du contrat signé entre le client et la société choisie pour réaliser ce développement.

2.2 L'intégrateur

Rôle : Au cours du projet de mise en place de la solution, c'est lui qui a la charge de conseiller le client sur l'architecture nécessaire, d'installer et de paramétrer le logiciel choisi dans l'environnement définitif. L'intégrateur n'a pas vocation à modifier le produit pour l'adapter aux besoins du client. Dans le cas d'une solution propriétaire, il arrive que l'éditeur et l'intégrateur soient la même société. Il est également possible que ce rôle soit directement pris en charge par les collaborateurs internes en charge du **Système d'Information (SI)** lorsque la taille de la structure et les compétences internes le permettent.

Intérêts défendus : Celui-ci souhaite que le client soit satisfait de la mise en œuvre concrète de la solution en vendant de la prestation de service. Il travaille généralement de concert avec des éditeurs et/ou communautés partenaires et leurs intérêts convergent souvent (certification d'un intégrateur par un éditeur via des formations qualifiantes).

2.3 Le mainteneur (ou support)

Rôle : C'est lui qui a la charge d'assister et de fournir des solutions aux collaborateurs internes en charge du **SI** en cas de problème au cours de la vie du service. Le support a généralement un délai maximum de résolution des incidents qui le lie contractuellement avec le client. Dans le cas du logiciel propriétaire, seul l'éditeur du logiciel peut en assurer le support, dans le cas d'un logiciel libre, plusieurs sociétés peuvent proposer ce service en s'appuyant sur le **code source** librement diffusé.

Intérêts défendus : L'intérêt du mainteneur est d'arriver à répondre aux demandes de résolution des incidents dans les délais définis contractuellement avec le client. Il va de soit que plus une solution logicielle est fiable, moins le support aura besoin d'intervenir et plus le rendement d'un tel contrat sera élevé puisque son montant est fixé de manière forfaitaire.

Introduction

Le logiciel libre², malgré une apparence utopique, offre dans bien des domaines des solutions fonctionnelles et performantes unanimement reconnues. Parallèlement, les besoins du secteur public en logiciels, aussi bien en produits standards que spécifiques à ses activités, sont de plus en plus importants. Ils sont en effet nécessaires à l'amélioration du service rendu aux usagers et à l'augmentation de la productivité des agents. À cela s'ajoute la nécessité de réduire les coûts dans le contexte économique que nous connaissons. Le choix du logiciel libre pour répondre aux besoins du secteur public apparaît de plus en plus comme une solution pertinente.

Par ailleurs, des signaux politiques ont récemment été envoyés dans ce sens. Les plus importants sont la circulaire « Ayrault » de septembre 2012³ sur les « *orientations pour l'usage des logiciels libres dans l'administration* » (source : [5]), ainsi que la feuille de route du gouvernement présentée lors du « séminaire gouvernemental sur le numérique » en février 2013, mentionnant que « *l'utilisation de logiciels libres sera encouragée dans l'administration* ». Fort de ce constat, il m'a paru pertinent d'approfondir ce sujet au travers de ce mémoire découpé comme suit :

Dans un bref préambule, nous avons décrit l'organisation la plus courante d'un projet informatique en milieu professionnel afin de définir clairement les acteurs d'un projet et leur rôles.

Dans un premier chapitre, nous nous attacherons aux grands principes du logiciel libre et nous verrons quels sont leurs points communs et leurs différences avec ceux du secteur public. Nous définirons brièvement ce qu'est un logiciel libre et d'où provient ce concept, nous nous attacherons ensuite à expliquer comment fonctionne un projet libre et quel sont les logiciels les plus populaires. Nous verrons enfin en quoi ces concepts peuvent ou non être en adéquation avec la philosophie et les contraintes propres du secteur public.

Dans un second chapitre, nous verrons quels sont les enjeux de l'adoption du logiciel libre par le secteur public. D'abord du point de vue de l'acheteur public ayant des contraintes budgétaires, juridiques et fonctionnelles ; puis du point de vue politique avec des contraintes d'indépendance vis-à-vis d'entreprises et de puissances étrangères, d'incitation à l'innovation et de stimulation de l'économie française.

Dans un troisième chapitre, nous nous attarderons sur les perspectives à court et moyen terme autour du logiciel libre et du secteur public. Dans un premier temps nous regarderons le marché du logiciel libre en France, sa place dans le monde et les initiatives inspirées par le logiciel libre (privées et publiques) ; nous aborderons ensuite le phénomène de l'informatique en nuage

2. Il est communément admis qu'un logiciel est dit « libre » lorsque celui-ci est gratuitement utilisable, modifiable et redistribuable. La modification de ce type de logiciel est rendue possible grâce à la diffusion sans contrainte de son « code source », véritable recette de fabrication permettant de s'approprier un logiciel et de le modifier pour l'adapter à son besoin. Cette catégorie de logiciel s'oppose aux logiciels dits « propriétaires » dont l'utilisation est soumise à un contrat de licence (le plus souvent payants) et dont la distribution est réservée à son éditeur. Le code source est, dans ce cas, tenu secret ce qui empêche un utilisateur d'un logiciel d'en développer une variante.

3. Circulaire n°5608/SG du 19 septembre 2012

(ou *cloud*), dont certains disent qu'il remplacera demain le logiciel tel que nous le connaissons, et de la manière dont le logiciel libre a un rôle à jouer à l'avenir sur ce marché en pleine expansion.

Notes importantes à l'attention du lecteur

Bien que nous parlions, dans ce mémoire, du secteur public au sens large, les exemples et descriptions concernent majoritairement la fonction publique d'État (ministères) même si la fonction publique territoriale est brièvement évoquée.

L'ensemble de ce mémoire est parsemé de nombreux exemples (encarts gris comme celui-ci) permettant d'illustrer le propos de façon concrète. La majorité de ces exemples pour lesquels la source n'est pas précisée proviennent d'entretiens réalisés auprès des personnes figurant dans les remerciements. Certains exemples proviennent également de conférences du salon « Solutions libres & open source » ayant eu lieu les 28 et 29 mai 2013 au CNIT de Paris - la Défense. Tous ces exemples sont recensés dans une table des exemples au début du document.

Les mots en **bleu foncé** sont des liens hypertextes dans la version électronique du document. Ils vous permettent d'avoir accès, soit à la définition du terme ou de l'expression, à la signification de l'acronyme ou encore au document cité comme source dans la bibliographie (numéro en bleu entre crochets). Pour la version papier, le lecteur se reportera à la liste des acronymes, au glossaire des termes techniques ou à la bibliographie.

Chapitre 1

Philosophie & grands principes

Introduction

En matière de logiciel, le modèle historique est celui du logiciel propriétaire qui consiste pour un utilisateur à acheter le droit d'utiliser un logiciel puis à s'en remettre entièrement à son éditeur pour ce qui est de ses évolutions ou des corrections à y apporter. C'est une situation très confortable pour l'utilisateur mais qui présente un risque de dépendance envers l'éditeur. C'est un peu comme si l'acquisition d'une voiture ne nous donnait pas le droit d'en ouvrir le capot pour vérifier le niveau d'huile ou changer un filtre.

A contrario, le logiciel libre propose un modèle entièrement ouvert et gratuit. Bien que le produit soit gratuit, il nécessite généralement un coût d'entrée plus important avant de pouvoir être utilisé efficacement. En effet, il requiert souvent un plus grand niveau de qualification en informatique et un investissement en temps plus important. C'est un peu comme si votre voiture était cette fois livrée en pièces détachées et nécessitait des compétences en mécanique pour être assemblée et plus tard réparée.

Dans ce chapitre nous nous attacherons d'abord à expliciter la philosophie des logiciels libres, son origine et sa mise en pratique. Nous expliciterons ensuite quels sont ses points communs et ses divergences avec la philosophie du secteur public.

1.1 Qu'est-ce que le logiciel libre ?

1.1.1 Un bref historique

1.1.1.1 Les années 1960 : l'avant licence

Jusqu'à la fin des années 60, le marché du logiciel est intimement lié à celui du matériel. De fait, l'achat de matériel s'accompagne de façon systématique du logiciel permettant de l'exploiter. À cette époque, le matériel est donc indissociable du logiciel et ces deux composants sont vendus par un même fournisseur de façon liée. La notion de licence logicielle n'existe pas encore, le [code source](#) d'un programme est librement fourni à qui souhaite y avoir accès puisque les coûts liés au développement d'un logiciel font partie intégrante du prix de vente du lot ainsi constitué.

1.1.1.2 Les années 1970 : le logiciel comme une fin en soi

Dans les années 70, certaines grandes entreprises du domaine informatique, comme Xerox et IBM, commencent à « protéger » leurs logiciels en refusant de fournir le [code source](#) aux utilisateurs voulant l'étudier ou l'améliorer. Le concept de « licence logicielle » naîtra de cette volonté de protéger la propriété intellectuelle d'un développement logiciel.

En 1976, Bill Gates écrira même une lettre adressée au Homebrew Computer Club¹ pour expliquer que l'usage des logiciels développés par sa société (Microsoft) sans s'acquitter des droits de licences est considéré comme un vol et que ces comportements découragent les développeurs dans leur effort pour créer des logiciels de qualité.

La même année, au [Massachusetts Institute of Technology \(MIT\)](#), Steven Levy publie « L'éthique du hacker » dont la première règle peut se résumer par : « Le partage d'information est un bienfait puissant et positif. . . »². Cette philosophie sera reprise plus tard comme principe fondateur du logiciel libre.

1.1.1.3 Les années 1980 : l'apparition du concept de logiciel libre

En 1985, Richard Stallman, considéré par beaucoup comme le père du logiciel libre, crée la [Free Software Foundation \(FSF\)](#) dont le rôle est notamment de concevoir et de maintenir un cadre juridique protecteur pour ce type de logiciels et leurs auteurs. La [General Public Licence \(GPL\)](#) version 1 contenant la notion de copyleft³ voit alors le jour en 1989. L'objectif initial de cette licence était d'unifier les licences alors utilisées pour distribuer les logiciels du projet [GNU is Not Unix \(GNU\)](#)⁴. Elle a permis de mieux formaliser les règles d'utilisation, de divulgation du [code source](#), de modification et de distribution des logiciels libres. La [GPL](#), aujourd'hui disponible dans sa version 3 depuis 2007, n'est pas la seule licence de ce type mais elle reste la référence⁵.

Le projet le plus emblématique utilisant cette licence viendra dans les années 1990 d'un étudiant finlandais, Linus Torvalds, qui proposera le noyau d'un système d'exploitation, qui sera ensuite intégré au projet [GNU](#) : Linux.

1.1.2 Le cadre juridique

1.1.2.1 La notion de copyleft

La logique qui sous-tend le copyleft est de prendre le contre-pied de la notion de copyright tout en s'appuyant sur celle-ci. En effet, le copyright entend protéger l'auteur d'une œuvre en autorisant sa diffusion selon des règles établies par celui-ci alors que le copyleft entend donner des droits au public auquel l'œuvre s'adresse en faisant renoncer l'auteur à tout ou partie de ses droits⁶. Empilées ces deux notions, copyright puis copyleft, permettent alors de diffuser une œuvre de façon libre⁷ (ce que permet le copyleft) sans pour autant en abandonner la paternité⁸ (ce que permet le copyright). Ce principe se traduit dans la pratique par l'interdiction pour quiconque apporte des modifications à un logiciel distribué sous une licence copyleft d'en distribuer ensuite le résultat sous une autre licence. On parle dans ce cas de licence héréditaire.

1. Club célèbre d'informatique de la Silicon Valley entre 1975 et 1986

2. « *The belief that information-sharing is a powerful positive good, and that it is an ethical duty of hackers to share their expertise by writing open-source code and facilitating access to information and to computing resources wherever possible* »

3. Par opposition à copyright. Parfois traduit en français par « gauche d'auteur »

4. [GNU is Not Unix](#) est un acronyme récursif (en référence à la récursivité en programmation)

5. Plus de la moitié des logiciels libres disponibles sont distribués sous cette licence. A noter que les logiciels les plus populaires sont souvent distribués selon une licence qui leur est propre (exemple : la Mozilla Public Licence sous laquelle est distribué Firefox)

6. En France, le droit moral attaché à une œuvre est perpétuel, inaliénable et imprescriptible. L'auteur ne peut donc y renoncer d'après l'article L121-1 du code de la propriété intellectuelle.

7. De façon gratuite, reproductible, redistribuable et modifiable

8. Interdiction pour quiconque de s'appropriier l'œuvre et d'en faire commerce en son nom sans l'autorisation de l'auteur

Pour résumer ce montage, le copyright est la notion supérieure qui permet de protéger l'auteur (contre l'appropriation de son travail par d'autres) et le copyleft se place en dessous pour élargir le plus possible la diffusion de l'œuvre en donnant des droits quasi-illimités au public (à l'exception de la restriction d'appropriation). En vertu de la supériorité du copyright sur une œuvre, l'auteur est donc le seul à pouvoir choisir de la placer sous les règles du copyleft ou non en choisissant une licence d'exploitation appropriée.

Il arrive qu'un auteur décide de changer le mode de diffusion des futures versions de son œuvre. C'est la crainte de beaucoup d'utilisateurs de logiciels libres dont le copyright a été racheté par des sociétés éditant traditionnellement des logiciels propriétaires. Nous reviendrons sur les risques inhérents au choix du logiciel libre au 2.1.3.3 page 23.

Bien que la notion de copyleft soit à l'origine du principe de licence libre, les licences utilisées aujourd'hui pour distribuer des logiciels libres ne sont pas toutes copyleft.

1.1.2.2 Les particularités des licences libres

Il existe un très grand nombre de licences de logiciels dites « libres » (environ 90). Nous nous attacherons principalement à la licence GNU-General Public Licence plus connue sous l'abréviation de GPL (il existe une cinquantaine de déclinaisons dites compatibles avec la GPL, i.e. n'ayant pas de contradiction avec celle-ci).

La licence GPL intègre pleinement la notion de copyleft et autorise donc l'utilisateur à exécuter, copier, modifier et redistribuer librement un programme et son code source, y compris pour un usage commercial. En revanche, il existe une limite aux droits de l'utilisateur dans le cas où celui-ci souhaiterait redistribuer le logiciel (avec ou sans modification) sous une autre licence. En effet, la licence GPL est une licence dite « copyleft » ce qui signifie que tout logiciel intégrant une partie d'un logiciel se trouvant sous cette licence doit également respecter l'ensemble des règles de la GPL et ne peut être distribué que sous une licence compatible avec elle. Ceci a pour effet d'interdire à quiconque de s'appropriier tout ou partie d'un logiciel distribué sous licence GPL. Il existe cependant une version de la GPL qui assouplit cette restriction (la Lesser GPL (LGPL)) afin d'autoriser des logiciels propriétaires à s'appuyer sur des briques logicielles sous licence libre. De ce fait, la LGPL ne peut être considérée comme une licence copyleft.

Une licence française dérivée de la GPL a vu le jour en 2004 grâce à une collaboration de trois organismes de recherche publics, le Commissariat à l'Énergie Atomique et aux énergies alternatives (CEA), le Centre National de la Recherche Scientifique (CNRS) et l'Institut National de Recherche en Informatique et en Automatique (INRIA) afin de mieux définir en droit français les modalités d'usages des logiciels libres. Cette licence se nomme la CeCILL⁹ et est principalement utilisée par les universitaires français souhaitant s'échanger librement des logiciels scientifiques. À l'image de la LGPL, la licence CeCILL-C est une version assouplie de la CeCILL permettant l'utilisation d'un logiciel sous cette licence dans des produits non libre.

N.B. : Les logiciels disponibles sous licences libres ne doivent pas être confondus avec les logiciels gratuits. En effet, il existe des logiciels propriétaires distribués gratuitement, c'est le cas par exemple d'Internet Explorer de Microsoft ainsi que d'un grand nombre de petits utilitaires informatiques (logiciels de compression par exemple). Dans ce cas, le code source n'est pas librement accessible. Il n'est donc pas possible de disposer de celui-ci pour l'étudier ou encore y apporter des modifications. Dans la mesure où l'éditeur conserve le contrôle total de ces logiciels, appelés « *freeware* », ils ne peuvent en aucun cas être considérés comme des logiciels libres.

1.1.2.3 Libre ou « Open-source » ?

Alors que ces deux notions sont souvent considérées comme synonymes, il existe une nuance d'ordre philosophique. En effet, le logiciel libre fait de la gratuité un principe de base¹⁰ alors que l'open-source met plutôt l'accent sur le fait que le `code source` soit diffusé sans contrainte. Il est ainsi relu, corrigé et amélioré par un large public lui assurant, par là même, une plus grande qualité. La méthode de développement du logiciel est donc l'idée centrale de l'open-source, plutôt que la liberté de l'utilisateur et la notion de gratuité. Dans quelques rares cas, il existe des licences reconnues par l'[Open Source Initiative \(OSI\)](#)¹¹ mais pas par la [FSF](#). D'un point de vue opérationnel, il n'existe pas de différence entre un projet libre et un projet open-source.

À l'origine de cette distinction se cache une réticence du monde professionnel à travailler avec des logiciels dont le modèle économique leur paraît fantaisiste. Il est effectivement plus logique d'accorder sa confiance à un logiciel en expliquant que celui-ci a été élaboré par le biais de méthodes de développement collaboratives grâce à l'ouverture de son `code source`, plutôt que de faire référence à des développeurs philanthropes non rémunérés recherchant la reconnaissance de leur pairs.

Aujourd'hui, le logiciel libre est sorti de la marginalité dans laquelle il est né. En effet, une part importante des contributeurs à des projets libres sont rémunérés pour leur travail sur ces logiciels. Le concept de libre porte alors plus sur la méthode de mutualisation des efforts de développement, que sur la gratuité qui peut donner le sentiment d'un manque de professionnalisme.

Pour illustrer la différence de philosophie entre ces deux termes, nous pouvons par exemple considérer que Linux, Firefox et LibreOffice sont des logiciels libres alors que Android, JBoss et OpenOffice sont des logiciels open-source. Les logiciels libres sont portés par des communautés composées d'acteurs d'horizon divers (entreprises utilisatrices du produit, utilisateurs bénévoles) alors qu'un logiciel open-source est généralement développé par une seule entreprise (dont on peut s'attendre à ce qu'elle recherche à un moment donné un retour sur investissement d'une manière ou d'une autre).

1.1.3 L'organisation d'un projet libre

1.1.3.1 La naissance d'un projet

Le projet GNU, certainement le plus emblématique des projets libres, a prospéré sur la frustration de Richard Stallman qui cherchait à se raccrocher à la culture *hackers* des années 70. Ce projet donnera naissance à une multitude de briques logicielles libres qui servent aujourd'hui de socle à un très grand nombre de solutions informatiques. D'autres, comme la fondation Mozilla, ont donné entre autres naissance à Firefox, né de la mort de la société Netscape en récupérant son patrimoine devenu libre faute de modèle économique viable. D'autres encore, comme l'association VideoLAN Organization, sont partis d'un projet étudiant de l'[École Centrale de Paris \(ECP\)](#) et l'ont fait évoluer jusqu'à obtenir un produit très abouti utilisé dans le monde entier ([Video-LAN Client \(VLC\)](#)).

Dans tous les cas, le formidable potentiel de distribution que constitue internet permet à ce type de projets de se répandre, parfois à une très grande vitesse, avec un coût de distribution dérisoire pour l'auteur (l'hébergement de fichiers sur internet est souvent proposé gratuitement). Il est probable que sans internet, les communautés de logiciels libres n'aurait jamais pu exister.

10. Le terme « *free* » en anglais signifie à la fois libre et gratuit

11. Organisation à but non lucratif visant à promouvoir les logiciels open-source

1.1.3.2 Le déroulement de la vie d'un projet

Alors qu'un logiciel libre peut naître d'une multitude de façons, on constate généralement qu'un projet en vie depuis un certain temps s'organise toujours à peu près de la même manière : lorsqu'un projet a passé un cap de maturité et commence à être utilisé par un nombre suffisant d'acteurs, il se crée naturellement une communauté autour de ce produit. Ces acteurs, dont les intérêts convergent, souhaitent tous voir évoluer le produit. L'importance et la motivation de la communauté qui « pousse » un logiciel libre est fondamentale, à la fois pour sa longévité et pour la réactivité dans les corrections d'anomalies. De la même manière que la taille d'une entreprise et son chiffre d'affaires nous assure de sa solidité, plus la communauté autour d'un logiciel est importante, plus le logiciel a des chances d'être maintenu longtemps et d'intégrer de nouvelles fonctionnalités régulièrement.

Il est intéressant de voir que certains logiciels libres, issus de développements internes dans une administration, sont repris par d'autres afin d'être améliorés. C'est le cas d'[Open Computers and Software \(OCS\) Inventory](#), une solution d'inventaire de parc informatique. Au départ, le développement de ce logiciel était porté par la [Caisse Nationale d'Assurance Maladie \(CNAM\)](#) puis, faute de ressources en interne, celui-ci fut abandonné. La Gendarmerie nationale s'est alors emparé du projet afin de le moderniser et d'y ajouter des fonctions de télédistribution de logiciels. Aujourd'hui, [OCS Inventory](#) dispose d'une vraie communauté très active dont deux gendarmes font partie. Outre l'administration, cette solution a séduit en 2011 la Banque du Brésil ([Banco do Brasil](#)) qui a décidé de l'utiliser pour inventorier son parc de 96 000 ordinateurs et de 44 000 [Distributeurs Automatiques de Billets \(DAB\)](#) (source : [31]).

Au delà de cet aspect de longévité, il existe aussi des cas où certains membres d'une communauté peuvent faire scission en raison d'intérêts divergents autour d'un logiciel. Ces divergences peuvent être de différentes natures : un désaccord sur un changement de licence, sur les fonctionnalités ou sur les choix techniques pour faire évoluer le produit. Ces ramifications appelées « fork » sont relativement courantes dans le monde du libre puisque les licences permettent de reprendre tout projet sans l'accord de qui que ce soit. Il faut cependant se méfier de cette tendance qui peut affaiblir les ressources cristallisées autour d'un projet.

Lorsqu'un logiciel libre est très utilisé, il n'est pas rare de voir se créer une fondation comme « The Mozilla Foundation » qui porte entre autre le développement de Firefox et Thunderbird ou « Apache Software Foundation » qui porte le serveur [HTTP Apache](#). Ces fondations ont parfois des locaux et des salariés et sont dans ce cas financées via des partenariats commerciaux avec des entreprises, des dons ou encore des revenus d'actifs.

1.1.3.3 L'extinction d'un projet

L'extinction d'un logiciel libre se produit, le plus souvent, à cause d'une trop faible activité de la communauté qui supporte le projet. Il est donc primordial, lors du choix d'une solution libre, de bien étudier l'importance et le niveau d'activité de sa communauté (les mises à jour proposées régulièrement sont un bon indicateur). Ceci permettra d'éviter de se retrouver avec une solution logicielle non pérenne ou d'avoir à en assumer seul l'avenir.

Afin de faire de bons choix en matière de solution logicielle libre, le ministère des Finances s'est doté, dans le cadre de son marché de [Support Logiciel Libre \(SLL\)](#), de la possibilité de recourir à des prestations de veille stratégique sur un domaine particulier. L'[Institut National de la Statistique et des Études Économiques \(INSEE\)](#) a, par exemple, récemment utilisé ce type de prestation afin d'avoir un panorama des meilleurs outils de [Système d'Information Géographique \(SIG\)](#) en vue d'équiper les enquêteurs en charge du recensement de la population française.

1.1.4 Exemples de solutions libres populaires

Voici une synthèse des principaux logiciels libres dans une grande variété de domaines fonctionnels. Il existe bien sûr une très grande quantité d'autres projets libres répondant à ces mêmes besoins ou à d'autres mais nous avons essayé de nous concentrer ici sur les projets les plus utilisés et les plus populaires.

1.1.4.1 Les logiciels du poste de travail

Catégorie	Logiciel	Commentaire
Système d'exploitation	Ubuntu Linux	Distribution de GNU/Linux la plus simple à installer et à utiliser
Navigateur web	Google Chrome	Basé sur le projet open-source Chromium développé principalement par Google
Navigateur web	Mozilla Firefox	La fondation Mozilla a repris le produit Communicator devenu libre à la mort de la
Messagerie électronique	Mozilla Thunderbird	société Netscape en juillet 2003 ¹
Suite bureautique	OpenOffice	Produit open-source développé par Oracle (cédé en 2011 à la fondation Apache)
Suite bureautique	LibreOffice	Développé par « The Document Foundation » issu d'un <i>fork</i> d'OpenOffice
Retouche d'image	GIMP	Souvent présenté comme une alternative crédible à Adobe Photoshop
Lecture et diffusion de médias	VLC	Né d'un projet étudiant de l'ECP en 1996

1.1.4.2 Les logiciels du terminal mobile

Catégorie	Logiciel	Commentaire
Système d'exploitation	Android	Produit open-source développé par la société Google, principal concurrent d'iOS d'Apple
Navigateur web	Google Chrome	Fourni avec Android, celui-ci existe également sur iOS

Le projet Promouvoir les Logiciels Utiles Maîtrisés et Économiques (PLUME) mené par le CNRS a pour vocation de répertorier les logiciels libres utilisés dans l'enseignement supérieur et la recherche mais aussi dans l'administration française en général. Ce projet se présente sous la forme d'un site internet : <http://www.projet-plume.org/> qui est aujourd'hui considéré comme une référence par le secteur public.

1. La société Netscape a été frappée de plein fouet par la concurrence de Microsoft lors de la mise à disposition gratuite d'Internet Explorer par celui-ci et son intégration au système d'exploitation Windows. Par la suite, la société Microsoft sera inquiétée par la justice américaine et européenne à de multiples reprises pour vente liée d'Internet Explorer avec Windows

1.1.4.3 Les logiciels « serveur »

Catégorie	Logiciel	Commentaire
Système d'exploitation	GNU/Linux	Devenu un standard de marché dans le monde professionnel (grand nombre de distributions disponibles)
Serveur HTTP	Apache	Logiciel le plus utilisé dans le monde pour servir des pages web sur internet
Serveur DNS	BIND	Logiciel le plus utilisé pour résoudre les noms de domaines sur internet
Serveur de messagerie	SendMail	Logiciel historique permettant l'acheminement d'e-mails
Serveur de messagerie	Postfix	Concurrent de Sendmail plus récent, il est plus rapide, plus facile à administrer et plus sécurisé
SGBD	MySQL	Idéal pour des bases de données de petite taille, s'installe et se paramètre très facilement
SGBD	PostgreSQL	Plus complexe que MySQL à maîtriser mais plus adapté à des bases de grande taille
Annuaire LDAP ¹	OpenLDAP	Solution d'annuaire à la norme LDAP utilisée en milieu professionnel
Serveur d'application Java	Tomcat	Logiciel porté par la fondation Apache très populaire dans le monde
Serveur d'application Java	JBoss	Produit open-source développé par la société RedHat, très utilisé en milieu professionnel
CMS	WordPress	Un gestionnaire de blog devenu CMS au fil des modifications proposées par sa communauté
Solution cryptographique	GnuPG	Implémentation libre de la norme OpenPGP ²
Téléphonie sur IP	Asterisk	Solution complète de téléphonie sur IP pour le monde professionnel
Gestionnaire de Wiki	MediaWiki	Moteur utilisé sur internet pour l'encyclopédie collaborative Wikipédia ³
Supervision	Nagios	Solution de supervision de serveurs physiques et de services applicatifs modulaires (par ajout de greffons)

1.1.4.4 Les langages de programmation

Catégorie	Langage	Commentaire
Langage semi-interprété	Java	Langage orienté objet dont l'idée première est de faciliter le portage d'applications d'un environnement à un autre. Depuis, Java est devenu une référence pour les applications web complexes.
Langage interprété	PERL	Langage conçu pour traiter les fichiers textes de façon simple et efficace, il est devenu au fil du temps une référence et un très grand nombre de modules optionnels sont librement disponibles aujourd'hui.
Langage interprété pour le web	PHP	Proche du PERL, avec des facilités pour générer des pages web et interroger des bases de données, ce langage est utilisé par des millions de sites web (généralement d'envergure restreinte).
Compilateur	GCC	Suite de compilateurs faisant partie du projet GNU, GCC permet entre autre de compiler les langages C et C++ dans lesquels Linux et la majorité des projets libres historiques sont écrits.

1. Lightweight Directory Access Protocol (LDAP) est un protocole standard de requête d'annuaire défini dans la RFC4510

2. OpenPGP est un standard de la cryptographie dont le principe est décrit dans la RFC4880 parue en novembre 2007

3. Ce moteur peut être installé sur un serveur pour créer un Wiki propre à une organisation

1.2 Le secteur public face au monde du logiciel

1.2.1 La notion de service public

1.2.1.1 De façon générale

La notion de service public peut varier selon les pays, le modèle de société et le rôle que l'on souhaite donner à l'autorité publique. En France, un service public a pour but de satisfaire un besoin d'intérêt général tout en respectant, autant que faire se peut, un principe d'égalité de traitement des citoyens, de continuité de service et de laïcité. À cela s'ajoute souvent un principe de gratuité (ou de quasi-gratuité) pouvant être vu comme une conséquence directe du principe d'égalité d'accès au service par tous les citoyens.

Ainsi, nous pouvons citer l'école « gratuite et obligatoire » voulue par Jules Ferry, l'assurance maladie bâtie sur l'égalité d'accès à des soins de qualité ou encore la défense nationale qui vise à protéger l'ensemble de la population d'un pays sans distinction d'aucune sorte. Ces grands principes sont tous la conséquence de l'application du modèle républicain résumé par la devise : « Liberté, Égalité, Fraternité ».

Il existe pourtant en France des services publics qui ont choisi une autre voie que celle de la gratuité. Arguant du fait que les ressources financières fournies par l'État sont insuffisantes pour l'exercice de leur mission, ces services publics ont choisi de faire un usage commercial de certains de leurs travaux. Prenons comme exemple les prévisions météorologiques réalisées par MétéoFrance qui sont gratuites pour le grand public (travaux agrégés) alors que les données des relevés météo (données brutes) font l'objet d'une exploitation commerciale. Cette opinion n'est cependant pas partagée par son homologue britannique, The MetOffice, qui distribue l'ensemble des données de ses stations météo de façon totalement gratuite et en autorise leur exploitation commerciale.

Bien que la gratuité ne soit pas une règle absolue pour l'ensemble des services publics, il n'en reste pas moins que les entorses à cette règle sont rares et n'existent que pour combler des carences de financement. Par ailleurs, lorsque ces cas existent, ils n'entravent pas les missions premières de ces services.

1.2.1.2 Pour le logiciel en particulier

À la lueur de la philosophie du logiciel libre explicitée précédemment, nous pouvons trouver de grandes similitudes avec celle des services publics. En effet, l'ouverture du `code source` nous rappelle le principe d'égalité d'accès du citoyen/utilisateur, la distribution gratuite des logiciels libres s'inscrit dans le modèle de gratuité voulu par la majorité des services publics, enfin la satisfaction des besoins d'intérêt général plutôt que l'intérêt particulier du créateur/éditeur de logiciel est précisément ce qui oppose un logiciel libre à un logiciel propriétaire.

Il est également possible de considérer qu'un logiciel produit par une administration, même si l'objectif premier de cette administration n'est pas de concevoir des logiciels, est une production comme une autre d'un service public. À ce titre, cette production doit bénéficier à tous les citoyens de manière égale et si possible de manière gratuite. En suivant cette logique, il serait donc normal que les logiciels développés par le secteur public soit distribués sous des licences libres.

« Je puis expliquer la base philosophique du logiciel libre en trois mots : liberté, égalité, fraternité. Liberté, parce que les utilisateurs sont libres. Égalité, parce qu'ils disposent tous des

mêmes libertés. Fraternité, parce que nous encourageons chacun à coopérer dans la communauté »

RICHARD STALLMAN

interview dans le magazine Programmez!, n° 95

1.2.2 La commande publique

1.2.2.1 De façon générale

Le code des marchés publics est un ensemble de règles qui permettent à une administration de passer commande de biens ou de services à un prestataire qui sera retenu (on parle alors de titulaire du marché). Ce code s'appuie sur trois principes : La liberté d'accès pour les candidats à la commande publique, l'égalité de traitement des candidats ayant répondu à l'appel d'offre et la transparence tout au long du déroulement de la procédure. Ces critères permettent d'assurer une équité de traitement des candidats souhaitant devenir fournisseurs de l'administration, ainsi que la bonne gestion de l'argent public.

Au delà de ces principes généraux, il est possible d'observer des conséquences induites par ces règles. En effet, un marché public ayant une durée limitée (généralement entre une et quatre années), lorsque celui-ci arrive à son terme et que le besoin existe toujours, il est nécessaire de procéder à un nouvel appel d'offre ayant un objet similaire. De ce fait, lorsque plusieurs acteurs sont capables d'y répondre avec des offres comparables, il s'opère généralement un roulement entre les différents acteurs. Même si rien n'oblige à ce comportement, il est couramment admis qu'il permet de répartir sur l'ensemble de ces acteurs l'avantage que représente la commande publique. Cet usage permet également d'éviter à l'administration de se retrouver dépendante d'un seul fournisseur puisque, chaque renouvellement de marché conduit au choix d'un nouveau titulaire.

1.2.2.2 Pour le logiciel en particulier

propriétaire : Comme décrit dans le préambule, l'acquisition d'une solution logicielle en milieu professionnel se compose dans la majorité des cas de trois aspects distincts :

- L'acquisition des licences : Il s'agit de l'achat du droit d'utiliser un logiciel à proprement parler. Le plus souvent, une licence correspond à un utilisateur du produit. Il est donc nécessaire d'acheter autant de licences qu'il n'y a de futurs utilisateurs du produit.
- Le recours à un intégrateur : Il s'agit d'avoir recours à de la prestation de service pour paramétrer et installer le logiciel dans l'environnement du client.
- La souscription d'un contrat de maintenance (ou support) : Il s'agit d'un contrat permettant à l'acheteur du produit de se retourner vers son éditeur à chaque fois que cela lui est nécessaire (rencontre d'un bug, incompatibilité avec un autre produit, etc.). Généralement, ce type de contrat est souscrit par le client sous la forme d'un abonnement forfaitaire pour une durée fixe reconductible.

La prise en compte du premier et du troisième aspect dans un marché public relève souvent d'une gymnastique complexe. En effet, il est parfois difficile de mettre réellement en concurrence plusieurs acteurs lors de l'acquisition de licences car les produits répondant à un même besoin sont souvent incompatibles avec les produits concurrents pouvant avoir été acquis précédemment (c'est le cas pour le système d'exploitation Windows de Microsoft). Mais surtout, il est impossible de faire assurer la maintenance d'un produit par un autre acteur que l'éditeur du logiciel puisque celui-ci est le seul à pouvoir modifier le produit (en vertu de la non divulgation du [code source](#)

de son produit). Le recours à l'intégrateur pose quant à lui moins de problèmes car il n'est pas nécessaire d'avoir accès au [code source](#) du produit pour effectuer ce type de prestations.

libre : Si nous reprenons les trois aspects cités ci-dessus dans le contexte du logiciel libre, il s'avère que :

- Le premier point n'a plus de raison d'être puisque le logiciel libre peut être dupliqué et utilisé librement par un nombre potentiellement infini d'utilisateurs. En cas d'accroissement du besoin sur ce logiciel, il ne sera pas nécessaire de recourir à une nouvelle commande de licence, pouvant être complexe ou onéreuse si elle n'a pas été prévue dans le contrat initial.
- L'intégrateur pourra être choisi librement de la même manière qu'avec un logiciel propriétaire. Petit bémol cependant : la plupart des éditeurs propriétaires proposent des « certifications » sur leur gamme de produits, cela peut être une aide pour le choix de l'intégrateur. Ce type de certification existe plus rarement dans le monde du logiciel libre.
- La maintenance (ou support) : Elle pourra être assurée par une multitude d'acteurs puisque chacun d'eux aura accès librement au [code source](#) du produit. Il n'y aura donc pas de contrainte forte sur le choix de la société. Il en ira donc de même lors du renouvellement du marché qui pourra donner lieu à une nouvelle mise en concurrence entre différents acteurs disposant des compétences suffisantes pour reprendre le [code source](#).

Un exemple concret illustrant cette mise en concurrence sur les marchés de SLL est celui du marché des ministères Économiques et Financiers mis en place en 2005, attribué aux sociétés Cap Gémini et Linagora, puis, à l'occasion de son renouvellement en 2008, à la société Atos. En 2012, ce marché a de nouveau été attribué à la société Linagora épaulée par Bull et 25 PME et TPE du secteur. Ce marché couvre aujourd'hui plus de 250 logiciels libres et permet donc, en comparaison avec le modèle du logiciel propriétaire, de ne pas avoir à souscrire autant de contrats de maintenance avec des éditeurs. Dans la pratique, un petit nombre de logiciels (une vingtaine) concentre l'essentiel des demandes. Il est prévu que les corrections produites dans le cadre de ce marché soient reversées aux communautés des logiciels concernés (dans les faits, 70 à 80 % de ces corrections sont effectivement intégrées). Fort de ce modèle, un accord cadre interministériel (hors finances) a été passé entre le Service des Achats de l'État (SAE) et les sociétés Capgémini, AlterWay et Zenika afin d'obtenir du support sur plus de 350 composants logiciels libres. Pour en bénéficier, les ministères devront toutefois passer leur propre marché subséquent à cet accord cadre. Si plusieurs ministères demandent du support pour un même logiciel, le tarif sera alors dégressif en fonction du nombre de souscripteurs.

1.2.3 La valorisation du patrimoine immatériel de l'État

1.2.3.1 De façon générale

La notion de patrimoine immatériel de l'État a été mise en lumière en 2006, à l'occasion du rapport Lévy-Jouyet¹² sur l'économie de l'immatériel. Ce rapport souligne que le dynamisme de l'économie reposera à l'avenir plus sur les créations de l'esprit que sur de la production matérielle (source : [29]). Fort de ce constat, l'Agence du Patrimoine Immatériel de l'État (APIE) a été créée en 2007 afin que l'État, en tant qu'acteur de l'économie, ne soit pas en reste en matière de valorisation de son patrimoine immatériel. Pour l'année 2011, l'APIE a évalué à 10 Md€ la valeur nette totale des actifs immatériels de l'État.

12. Rapport de Maurice Levy et Jean-Pierre Jouyet remis en novembre 2006 au ministre des Finances de l'époque, Thierry Breton

1.2.3.2 Pour le logiciel en particulier

L'administration peut être tour à tour producteur ou consommateur de logiciel selon les cas. Il existe même une troisième possibilité, très fréquente, dans laquelle l'administration fait développer à une société tierce un logiciel en fonction de ses besoins puis l'acquiert en pleine propriété (elle possède le [code source](#)) et s'occupe ensuite d'en assurer la maintenance et les évolutions futures. En matière de patrimoine immatériel, ce qui intéresse l'APIE ce sont les logiciels acquis de cette manière ou développés entièrement en interne. Dans ce cas, l'administration est considérée comme éditeur de logiciel pour ses propres besoins. L'APIE défend alors l'idée que l'administration ne doit pas céder gratuitement ce qui lui a demandé des efforts financiers et/ou humains. Dans ce contexte, il est évident que si tous les logiciels dont l'administration possède le [code source](#) étaient mis sous licence libre, ce patrimoine aurait une valeur nulle...

Pour l'année 2011, l'APIE a évaluée à 1 Md€ la valeur nette totale des logiciels détenus par l'État. Il convient d'y ajouter les logiciels encore en cours de développement fin 2011, c'est le cas par exemple du projet Chorus¹³ pour près d'1 Md€ supplémentaire.

Conclusion

Pour conclure ce chapitre, nous pouvons dégager deux visions différentes de ce qu'est un logiciel :

- La première consiste à considérer un logiciel comme n'importe quelle création immatérielle pouvant être produite de façon industrielle ou non, il s'agit de la vision défendue par les éditeurs de logiciels propriétaires. Elle peut être comparée à la distribution de livres (dématérialisés ou pas) soumis à des droits d'auteurs.
- La deuxième consiste à considérer un logiciel (et en particulier son [code source](#)) comme une connaissance d'ordre général ayant pour but d'irriguer au maximum la société afin que chacun puisse en tirer parti. Elle peut être comparée à la libre mise à disposition de cours et de publications de recherche sur internet par les universitaires.

Cette seconde vision est parfaitement bien illustrée par l'extrait suivant :

« Si tu as une pomme, que j'ai une pomme, et que l'on échange nos pommes, nous aurons chacun une pomme. Mais si tu as une idée, que j'ai une idée et que l'on échange nos idées, nous aurons chacun deux idées. »

GEORGE BERNARD SHAW

écrivain irlandais (1856-1950), prix Nobel de littérature

Intuitivement nous pouvons penser que les valeurs qui animent les services publics sont plus proches de la seconde vision que de la première. De plus, le secteur public n'étant pas soumis aux lois du marché et en particulier à la concurrence, celui-ci n'a pas la même vision de ses intérêts que le secteur privé et n'a pas à craindre une perte de compétitivité du fait de l'appropriation de son travail par d'autres. Mais, bien que ces arguments puissent trouver des justifications philosophiques, ces dernières viennent s'entrechoquer avec une multitude d'autres paramètres dans la pratique, au premier rang desquels figure l'offre de logiciel libre disponible qui couvre des domaines fonctionnels bien moins vaste que l'offre de logiciels propriétaires.

13. Le projet Chorus est un projet de grande envergure de refonte de l'informatique comptable de l'État sur 5 ans. Il permet la gestion de la dépense et des recettes non fiscales selon les dispositions contenues dans la [Loi Organique relative aux Lois de Finances \(LOLF\)](#). Ce programme était à l'origine piloté par la [Direction Générale de la Comptabilité Publique \(DGCP\)](#), devenue depuis la [Direction Générale des Finances Publiques \(DGFIP\)](#)

Chapitre 2

Enjeux du logiciel libre dans le secteur public

Introduction

Le logiciel libre est aujourd'hui aux portes des administrations qui sont maintenant confrontées à des choix. Bien que le modèle dominant soit souvent celui du logiciel propriétaire, de nombreux arguments montrent chaque jour un peu plus que les logiciels libres sont une alternative crédible.

Dans ce chapitre, nous verrons quels peuvent être les enjeux du choix du logiciel libre dans le secteur public, d'abord en considérant l'administration comme un consommateur de logiciel ayant ses contraintes propres (marchés publics en particulier) ; puis en considérant que le secteur public, au travers de ses choix, a une mission politique (souveraineté, stimulation de l'innovation et de l'économie notamment), nous regarderons en quoi le choix du logiciel libre peut être différent du choix du logiciel propriétaire.

2.1 Du point de vue de l'acheteur public

2.1.1 Aspects financiers

2.1.1.1 Coût d'acquisition

Comme nous avons pu le voir, en matière de logiciel libre, le coût d'acquisition d'un produit est nul à partir du moment où le logiciel choisi correspond totalement à nos besoins. Dans le cas contraire, il sera nécessaire de faire une étude de marché pour trouver le produit libre le plus proche de nos besoins et évaluer le coût de son adaptation. Dans ce cas, le coût d'achat d'un logiciel propriétaire peut parfois se révéler plus avantageux que de supporter les coûts d'adaptation d'un logiciel libre. Attention cependant, ces deux solutions ne sont pas strictement équivalentes puisque, dans le cas de l'adaptation d'un logiciel libre, il peut être décidé de reverser ces modifications à la communauté et ainsi que tout un chacun puisse bénéficier de ces nouvelles fonctionnalités. Selon les cas, cela peut être un avantage (plusieurs administrations ayant les mêmes besoins) ou un inconvénient (entreprises concurrentes).

Il existe également un autre coût qu'il ne faut pas négliger lors de l'acquisition d'une nouvelle solution logicielle, c'est celui de la formation des personnels qui en seront, à terme, utilisateurs et/ou administrateurs. En effet, bien souvent les éditeurs de logiciels propriétaires font le maximum pour éviter que leur produits ne nécessitent de trop grandes compétences en matière de système d'information (en particulier pour les produits du poste de travail). Ce n'est pas toujours

le cas des logiciels libres dont les développeurs sont parfois plus soucieux de l'exploit technique que de la facilité d'utilisation du produit final. Cependant, beaucoup de projets libres rattrapent depuis quelques années leur retard sur ce terrain (ex : La distribution Ubuntu de Linux dont la devise est « *Linux for Human Beings*¹ »).

La Gendarmerie nationale a par exemple estimé avoir réalisé 20 M€ d'économie en licences à niveau de service équivalent en choisissant OpenOffice en lieu et place de Microsoft Word 2007 (source : [23]).

Parmi les exemples d'économies qu'ont permis de réaliser les logiciels libres, nous pouvons citer le projet de refonte du système d'information de l'École Nationale de l'Administration (ENA) opéré en 2010-2011 qui a fait le choix d'utiliser le progiciel de gestion OpenERP et de le faire adapter à ses besoins par une société, faisant ainsi économiser plusieurs centaines de milliers d'euros en coût d'acquisition selon Didier Georgieff, le Directeur des Systèmes d'Information (DSI) de l'ENA. Les développements effectués pour adapter ce produit aux besoins de l'ENA ont ensuite été reversés à la communauté (sources : [13][21]).

2.1.1.2 Mise en concurrence du support

Comme évoqué dans le premier chapitre, l'utilisation de logiciels libres permet à plusieurs acteurs du marché informatique de proposer du support sur ces logiciels. Cela a nécessairement pour conséquence de stimuler la concurrence entre les acteurs ayant les compétences pour offrir ce support. Cela évite également d'être tributaire des augmentations de coût du support décidées par l'éditeur pour un logiciel dont l'investissement initial a été important (financièrement et en temps) et qu'il est donc difficile de remplacer.

Dans le cadre des marchés publics, cette meilleure mise en concurrence apparaît comme plus conforme à la nécessité de bonne gestion des deniers publics.

Suite à l'accord cadre du SAE sur le support des logiciels libres, un marché subséquent a récemment été passé par le ministère de l'Intérieur. Ce marché permet aujourd'hui à ce ministère de bénéficier d'un support sur 70 briques logicielles libres pour un montant annuel de 428 000 €. Cette somme est considérée comme très faible en regard de ce que coûterait la souscription d'autant de contrats de maintenance auprès d'éditeurs différents. Par ailleurs, le ministère de la Culture se dit très satisfait de ce marché qui lui permet d'avoir un support de très bonne qualité sur le Content Management System (CMS) Liferay. Notons que si plusieurs ministères souscrivent au support d'un même produit (dans la limite de 3 versions différentes), une remise de prix est prévue par le prestataire.

Il est important de préciser que le changement de prestataire assurant le support de logiciels libres ne se fait pas sans difficultés pour certains logiciels structurants. En effet, CapGémini, actuellement retenu dans l'accord cadre du SAE pour assurer le support interministériel, a rencontré de grandes difficultés avant de réussir à compiler LibreOffice dans son propre environnement (ce qui l'a contraint à faire appel à la société Suse/Novell en attendant). Pareillement, dans le cadre du marché de support du ministère des Finances, Linagora a rencontré des difficultés importantes pour reprendre le support de la version de JBoss précédemment assuré par Atos. Dans les deux cas, ces problèmes ont été surmontés au bout de quelques mois entraînant un pic du délai de résolution des incidents pendant cette période.

La Direction Interministérielle des Systèmes d'Information et de Communication (DISIC), dont la mission est de rationaliser les différents systèmes d'information de l'État en recherchant

1. Linux pour tous

des pistes de mutualisation des besoins des structures publiques, a mis en place plusieurs groupes de travail interministériels. Parmi ceux-là se trouvent des groupes de travail sur le logiciel libre (système d'exploitation, suite bureautique, téléinventaire et télédéploiement, [Système de Gestion de Base de Données \(SGBD\)](#)) dont le rôle est de limiter l'éparpillement dans le choix des logiciels et de leurs versions dans les différentes administrations. Ainsi, lors du rattachement d'un ministère à un accord cadre interministériel de support comme celui du [SAE](#), le mainteneur n'aura à gérer qu'un nombre limité de versions, ce qui lui permettra de mutualiser ses efforts et donc de proposer des tarifs plus avantageux.

2.1.1.3 Mutualisation des coûts de développement

Dans le cas d'un logiciel propriétaire du marché fourni par un éditeur, une fois les coûts de développement initiaux amortis, celui-ci répercute, en théorie, les coûts d'évolution sur les différents clients du produit par l'intermédiaire, soit d'un contrat de maintenance lorsque celui-ci inclut les évolutions, soit par le biais de la facturation de nouvelles licences à l'occasion d'une montée de version.

Il n'est cependant pas rare de voir un éditeur de logiciel propriétaire vendre des licences d'un logiciel toujours au même prix alors que celui-ci a été maintes fois amorti et que les évolutions qu'il y apporte sont mineures à chaque nouvelle version. Comme pour un grand nombre de marchés aujourd'hui, la fixation des prix des logiciels se fait plus à partir de la demande des consommateurs (ce qu'ils sont prêts à payer) qu'à partir d'un calcul basé sur le coût de revient. À ce jeu là, plus un acteur est important, plus il sera favorisé par l'effet de masse de ses ventes.

Dans le cas d'un logiciel libre, les efforts de développement portent sur la communauté, dont chaque acteur qui souhaite voir le produit évoluer peut faire partie. Les coûts sont donc également répartis entre les utilisateurs mais sont à la charge de ceux qui ont des besoins d'évolutions en premier. Les autres utilisateurs bénéficient alors sans surcoût de ces évolutions plus tard en adoptant la version les intégrant.

Un exemple intéressant est celui de la Gendarmerie nationale qui souhaitait adopter le logiciel de messagerie libre Thunderbird, mais estimait qu'il lui manquait une fonction de calendrier partagé appropriée. Un marché pour un montant de 494 000 € a alors été lancé afin de combler cette lacune. La société Linagora a été retenue pour son exécution et le résultat de ces travaux a été intégré à la solution de travail collaboratif [Open Business Management \(OBM\)](#). Grâce à ce projet, l'ensemble des administrations, entreprises privées et particuliers peuvent maintenant bénéficier de ce logiciel (la [DGFIP](#) l'a par exemple déployé pour un grand nombre d'utilisateurs). À titre de comparaison, le logiciel Microsoft Outlook qui comporte déjà cette fonctionnalité a un coût estimé à 60 € par utilisateur, ce qui aurait représenté un coût total d'équipement de 4,8 M€ pour les 80 000 postes de travail de la Gendarmerie.

Afin d'aider le secteur public local à mettre en commun ses besoins métiers et d'éviter que l'argent public ne paye de multiples fois le développement de logiciels similaires, l'[Association des Développeurs et des Utilisateurs de Logiciels Libres pour les Administrations et les Collectivités Territoriales \(ADULLACT\)](#) a été créée en 2002. Elle vise à établir une gouvernance unique et à favoriser la mise en commun de logiciels métiers entre ces administrations via leur diffusion sous licences libres (ex : [OpenCimetière](#), [OpenMairie](#), [parapheur électronique](#)...). L'association propose également un service informatique concret permettant aux services publics de collaborer sur des projets libres : la [forge ADULLACT](#) (source : [18]).

Un exemple emblématique est celui du projet Lilie lancé par la région Île-de-France en 2009 qui vise à proposer un *Espace Numérique de Travail (ENT)* commun à l'ensemble des lycées de la région sous la forme d'un portail web de service de vie scolaire pour les élèves et les personnels. Ce projet a pour but d'uniformiser les pratiques dans les lycées de la région en matière de système d'information (un seul outil plutôt que des contrats et des pratiques disparates dans chaque établissement). Cette solution a été pensée dès le début du projet pour être open-source afin, entre autres, que les autres régions françaises puissent en bénéficier. Début 2012, trois régions avaient déployé ou envisageaient le déploiement de cette solution avec deux intégrateurs différents (Logica pour 400 établissements en *IDF*, Atos pour la région Picardie et 2 départements d'*IDF*).

La ville de Paris a été pionnière avec le projet Lutèce : Un socle web offrant des fonctions de portail internet, de *CMS* et de cadre de développement d'applications. Ce projet, qui était interne à la ville de Paris au départ, a été libéré en 2002 à la suite d'une décision du conseil de Paris impulsée par la publication d'un rapport favorable au logiciel libre de Thierry Carcenac adressé au premier ministre de l'époque [11]. Lutèce est aujourd'hui utilisé par paris.fr, marseille.fr, notaires.fr, MétéoFrance (portail public meteo.fr), Thalys TheCard (programme de fidélité client de Thalys) et la mairie de Bobigny.

2.1.1.4 Coûts cachés

Le logiciel propriétaire comporte des coûts cachés peu visibles lors du choix d'une solution informatique. En effet, les éditeurs de logiciels ont souvent tendance à faire évoluer leurs produits en y ajoutant de nouvelles fonctionnalités mais sans se soucier du surcroît de puissance nécessaire pour exécuter les nouvelles versions. S'en suit, plus ou moins rapidement, une obsolescence du matériel qui représente un coût non négligeable.

Une approche pragmatique dans ce domaine est celle de Pôle emploi qui a choisi d'utiliser Linux et Firefox plutôt que Windows et Internet Explorer pour ses bornes en libre-service permettant aux usagers d'accéder aux services en ligne du site www.pole-emploi.fr. Parmi les avantages de cette solution figure les économies sur les licences mais aussi la possibilité d'amortir un parc de *PC* sur une durée beaucoup plus longue (Linux étant moins consommateur en ressource matérielle que Windows). Ce choix a également été motivé par le fait que cette solution offre de meilleures possibilités de sécurisation des postes (verrouillage de l'accès aux seuls contenus autorisés).

L'obsolescence prochaine de Windows XP (absences de mises à jours de sécurité du produit au delà d'avril 2014) conduit certaines administrations à s'interroger sur l'avenir du système d'exploitation sur les postes de travail. En effet, le parc informatique de la *DGFiP* (plus de 100 000 machines) étant vieillissant, une grande partie de ce parc n'est pas compatible avec Windows 7. Ce constat pousse aujourd'hui la *DGFiP* à s'interroger sur un passage de ces postes sous Linux mais les difficultés semblent aujourd'hui très nombreuses.

Le logiciel libre comporte lui aussi des coûts cachés qui tiennent le plus souvent à l'effort nécessaire pour passer d'une solution propriétaire à une solution libre. En effet, lorsqu'une solution propriétaire est en place et que tout le monde sait l'utiliser, son coût récurrent est relativement faible en regard d'un projet de bascule vers une solution libre. De même, il est souvent moins coûteux de monter en version un logiciel propriétaire que de le basculer vers du libre, en raison de la rétro-compatibilité des versions entre elles et de la capitalisation sur les habitudes d'usages.

Beaucoup d'administrations ayant adopté LibreOffice ont dû mettre en place des formations internes sur ce produit. L'institut de formation du ministère des Finances a par exemple ajouté à son catalogue des formations à tous les modules de la suite (traitement de texte, tableur, présentation...) afin d'aider les directions ayant choisi de passer de Microsoft Office à LibreOffice.

2.1.2 Respect du code des marchés publics

2.1.2.1 Liberté d'accès à la commande publique

La première des exigences de la commande publique est celle de la liberté d'accès pour les entreprises à cette commande. En effet, lorsqu'une structure publique connaît un besoin qu'elle n'est pas capable de couvrir elle-même, celle-ci a recours à un appel d'offre décrivant sans ambiguïté ses besoins et exigences. En principe, les besoins doivent être décrits de manière suffisamment générale pour que cette commande n'exclue pas l'un des acteurs ou encore ne risque pas d'en favoriser un autre.

Cette exigence est bien complexe à remplir lorsqu'il s'agit d'acquérir une solution logicielle. En effet, les logiciels propriétaires étant rarement compatibles entre eux, il est très complexe d'imaginer, par exemple, un mélange de plusieurs systèmes d'exploitation sur les postes de travail des utilisateurs en fonction de la période où ils ont été acquis. Une gestion de ce type serait très inconfortable pour les agents en charge d'un parc de machines hétérogène et pour les utilisateurs qui seraient assurément très mécontents de devoir tout réapprendre à chaque changement d'ordinateur.

Là encore, le logiciel libre apporte une réponse élégante et en adéquation avec cette exigence puisque l'acquisition de ce type de logiciel ne nécessite aucune commande. En effet, un logiciel libre ne s'acquiert pas puisqu'il appartient, par essence, à tous et peut s'installer et s'utiliser librement. La commande publique n'existe tout simplement pas avec ce type de logiciel et seul le support peut donner lieu à la contraction d'un marché public avec une société. Dans ce cas, le `code source` étant librement accessible, plusieurs acteurs pourront librement répondre à ce besoin sans qu'il ne soit nécessaire pour eux d'être l'éditeur de ce logiciel.

2.1.2.2 Égalité de traitement

De la même façon que pour la liberté d'accès à la commande publique, l'égalité de traitement des candidats face à un besoin d'intégration et/ou de support sur un logiciel libre est assuré par un accès égal au `code source` du produit.

Au sujet de la mise en œuvre d'un logiciel libre précis, la région Picardie à été confrontée en 2011 à une attaque en justice par deux éditeurs de logiciels propriétaires s'estimant mis à l'écart d'un appel d'offre ayant pour objet le déploiement de la solution de l'ENT Lilie dans les lycées. Le tribunal administratif d'Amiens a, dans un premier temps, condamné la région au motif qu'il n'est pas possible de faire référence à une marque ou à un produit dans un appel d'offre dès lors que cela fausse les règles de concurrence. En dépit de ce jugement et de l'avis du rapporteur public sur ce dossier, le Conseil d'État a fait annuler cette décision en vertu de la distinction entre un marché de fourniture de logiciel et un marché de service. En effet, rien n'empêche ces deux éditeurs de logiciels à répondre à cet appel d'offre et à proposer du service autour du logiciel Lilie (source : [33]).

Un exemple intéressant dans ce domaine est celui du produit JBoss, un serveur d'application open-source (sous licence *LGPL*) très utilisé comme socle pour des applications web en Java. Ce produit a été développé par la société JBoss qui le distribue en open-source et propose des prestations autour de ce produit. Cette société est restée pendant longtemps la seule à proposer un support avancé sur ce produit. Le marché de support JBoss contracté par la *Direction Générale des Impôts (DGI)* en 2004 a alors été attribué à la société Atos avec comme partenaire pour les demandes les plus complexes, la société JBoss (source : [44]). Les années passant, la compétence des sociétés de services informatiques (et en particulier d'Atos) sur ce produit s'est améliorée et lui a permis d'offrir un niveau de support avancé. C'est alors qu'en 2008, lors du renouvellement du marché *SLL* du ministère des Finances, la société Atos s'est positionnée pour offrir du support sur un grand nombre de logiciels libres dont JBoss. Cette opération a alors permis d'éviter d'avoir un marché spécifique pour ce produit, permettant ainsi de réaliser une économie importante.

2.1.2.3 Roulement des titulaires

À la lueur de l'exemple ci-dessus, nous pouvons constater que, même si de solides compétences sont nécessaires pour qu'une société puisse répondre à un appel d'offre de support d'une solution logicielle libre, ces marchés sont accessibles à un nombre croissant de sociétés à mesure que les logiciels dont il est question sont répandus. Il n'existe donc pas de frein au roulement des titulaires de ce type de marchés à partir du moment où une société se donne la peine d'avoir les compétences nécessaires. Il est important de remarquer que plus la solution logicielle sera répandue, plus la concurrence entre les sociétés sera importante.

2.1.3 Qualité

Une idée répandue chez les acheteurs consiste à juger la qualité d'un produit en fonction de son prix d'acquisition. En suivant ce raisonnement, le logiciel libre n'aurait donc aucune valeur alors qu'un logiciel propriétaire onéreux serait forcément gage de qualité. Par ailleurs, la psychologie d'un consommateur le pousse à penser qu'une grande marque connue sera forcément meilleure qu'une autre moins visible commercialement. Après tout ce que nous venons de voir à ce sujet, il apparaît que cette idée ne s'applique pas au domaine du logiciel.

*La notoriété d'une marque a souvent un impact fort sur la perception de la qualité de ses produits par le grand public. Or, contrairement aux logiciels de grands éditeurs qui renvoient à une image de qualité, les logiciels libres ne font l'objet d'aucun marketing et sont assimilés par certains utilisateurs à des marques inconnues perçues comme « bas de gamme ». À tel point que certains utilisateurs au ministère des Finances se montrent plus tolérants lorsqu'ils rencontrent des difficultés avec un produit propriétaire que libre. Le refus de ces utilisateurs d'abandonner Microsoft Office pour LibreOffice (ou de passer de l'iPad à des tablettes équipées d'Android) fait craindre un effritement de la démarche de migration voulue par le *Secrétariat Général - Service de l'Environnement Professionnel (SG-SEP)*.*

2.1.3.1 Facilité d'utilisation

En matière de facilité d'utilisation, le logiciel libre est parfois moins axé sur les problématiques des utilisateurs que ne le sont les logiciels propriétaires. On constate cependant depuis une dizaine d'années une réelle volonté d'aller vers des préoccupations plus proches de l'utilisateur. Le logiciel libre comble ainsi peu à peu son retard sur ce point.

Le système d'exploitation Ubuntu, dont l'ambition est de mettre Linux à la portée de tous, a été utilisé par la Gendarmerie nationale pour équiper l'ensemble des postes de travail des agents. De l'aveu même de Xavier Guimard, superviseur du projet de basculement de Windows XP à Ubuntu, aucune conduite du changement n'a été nécessaire pour mener à bien cette opération. En effet, les utilisateurs étant déjà habitués à utiliser des logiciels libres sous Windows (OpenOffice, Firefox et Thunderbird), le changement de système d'exploitation s'est révélé n'être qu'une formalité. En revanche, ce propos doit être nuancé par le fait que la bascule de Microsoft Office à OpenOffice, opérée par la Gendarmerie quelques années plus tôt, a rencontré beaucoup plus de difficultés d'adaptation de la part des utilisateurs.

2.1.3.2 Fiabilité

Sur le terrain de la fiabilité, les logiciels libres sont aujourd'hui reconnus pour être, dans la grande majorité, supérieurs aux logiciels propriétaires. Ceci s'explique par le fait qu'une communauté est généralement constituée d'un grand nombre d'acteurs d'horizons et de cultures informatiques différentes. Ainsi, le `code source` de ces logiciels est relu par tous ces acteurs et bénéficie de la somme des compétences de tous ces profils. Au final, leur fiabilité est supérieure à celle d'un logiciel développé par des équipes plus homogènes et parfois plus guidées par le *business* que par la qualité du `code source`.

*Pour constater la fiabilité des logiciels libres, il suffit de regarder quels sont les logiciels les plus utilisés sur les serveurs web dans le monde. En effet, ce type de serveur nécessite une très grande fiabilité sous peine de voir le site internet de telle entreprise ou telle institution ne plus être disponible pour les internautes. Comme indiqué dans le tableau en 1.1.4.3 page 11, le logiciel phare qui permet de rendre accessible un site web se nomme Apache. Au 1^{er} janvier 2012, il avait une part de marché de près de 65%. Son principal concurrent, un logiciel propriétaire distribué par Microsoft nommé *Internet Information Server (IIS)*, se situe à moins de 15% (source : [15]). De même, parmi les serveurs d'application Java permettant aux sites web de proposer du contenu dynamique, les 4 premiers du classement sont des logiciels libres (Tomcat, Jetty, JBoss et GlassFish) totalisant plus de 82% du marché là où les solutions propriétaires (principalement WebSphere d'IBM et WebLogic d'Oracle) sont très en retrait. (source : [17])*

2.1.3.3 Pérennité

En ce qui concerne la pérennité d'un logiciel, les choses sont moins tranchées. En effet, un logiciel libre n'est pas toujours gage de pérennité et en particulier lorsque la communauté n'est plus active. En effet, comme nous l'avons évoqué au chapitre 1.1.3.3 page 9, un logiciel libre peut mourir faute de contributeurs. Dans ce cas, bien qu'il soit légalement possible de continuer à l'utiliser éternellement, les montées de version des composants utilisés par le produit finiront par le rendre incompatible avec un environnement récent. Par ailleurs, les failles de sécurité qui pourraient y être découvertes ne seront jamais comblées et obtenir du support deviendra de plus en plus difficile.

Un autre risque concernant un logiciel libre est de voir le copyright de celui-ci être racheté par une société n'ayant pas l'intention de maintenir la distribution sous licence libre. Dans ce cas, l'avenir du produit peut devenir incertain car, comme nous l'avons vu au chapitre 1.1.2.1 page 7, la personne qui détient le copyright sur le `code source` peut en changer les règles de distribution pour les versions futures.

La meilleure illustration de ce propos est le rachat de la société *Stanford University Network (SUN microsystems)* par Oracle intervenu en 2009. En effet, *SUN microsystems* était depuis longtemps un pilier dans le monde du logiciel libre par ses multiples projets distribués sous licence GPL (Java, MySQL, OpenOffice, OpenSolaris et VirtualBox). Après ce rachat, Oracle étant un éditeur de logiciels propriétaires, les communautés autour de MySQL et OpenOffice ont souhaité se détacher des ex-activités de *SUN microsystems* en créant des projets parallèles (ou « fork »). Ces projets se nomment MariaDB pour MySQL (dont la communauté est dirigée par son créateur) et LibreOffice pour OpenOffice (porté par « The Document Foundation »). En pratique aujourd'hui, cette stratégie n'a pas permis à Oracle de transformer les acteurs des communautés en clients captifs et seul le nom « commercial » des produits leur appartient réellement puisque les futurs efforts des communautés vont désormais se porter sur les produits issus des *fork* réellement libres, à savoir LibreOffice et MariaDB.

Il existe un autre risque de détournement de logiciel, plus gênant cette fois. Il consiste, pour un éditeur de logiciel, à proposer deux versions distinctes d'un même produit, l'une dite « premium » distribué sous licence propriétaire et l'autre dite « communautaire » sous licence libre. La société Alfresco a par exemple choisi ce système de distribution tout en affirmant être un éditeur de solution open-source. Le risque dans ce cas est de voir la version communautaire stagner (puisque'elle n'est portée par aucune communauté en réalité mais seulement par l'éditeur) alors que la version propriétaire continuera à évoluer en proposant régulièrement des nouveautés. Là encore, il s'agit d'une stratégie commerciale d'appropriation de l'image du logiciel open-source pour amener les clients vers une solution propriétaire. Lorsque cela est possible, il est préférable de choisir une solution réellement libre.

En ce qui concerne le logiciel propriétaire, l'éditeur peut également décider d'abandonner un produit en arrêtant de proposer du support sur celui-ci s'il estime que ce n'est plus suffisamment rentable. La solution dans ce cas pour les clients est d'accepter une augmentation de prix suffisante pour être attractive pour l'éditeur. Notons que s'il est toujours possible de reprendre en interne un logiciel libre orphelin, il est rare qu'un éditeur accepte de céder le code source de son produit, fût-il considéré comme « mort ».

Une autre problématique lorsqu'on parle de pérennité d'une solution logicielle libre est celle de la durée de vie d'une version majeure. En effet, certains logiciels libres ont un rythme de mise à jour très soutenu (nouvelle version tous les 2 mois par exemple) or, dans un environnement professionnel, il n'est pas toujours possible de mettre régulièrement à niveau un logiciel car la période de test nécessaire avant un changement de version peut être longue. Pour répondre à cette problématique, souvent peu prise en compte par les communautés de logiciels libres, la fondation Mozilla propose des versions *Extended Support Release (ESR)* de Firefox et de Thunderbird. Le principe est de créer un *fork* temporaire d'une version donnée afin de maintenir pendant une année complète cette souche du produit (en y apportant que les corrections de sécurité nécessaires pour ne pas qu'elle soit vulnérable). À l'issue de cette période, une version plus récente intégrant plusieurs versions successives prendra alors sa place pour une nouvelle année.

2.2 D'un point de vue politique

D'un point de vue politique, lorsque le secteur public se procure un logiciel, de nombreuses retombées peuvent se produire, à la fois pour l'administration qui en fait l'acquisition mais aussi pour l'écosystème économique qui gravite autour de ce logiciel. Nous allons voir ici les différentes retombées que nous pouvons attendre du choix du logiciel libre et les comparer avec le choix du logiciel propriétaire.

2.2.1 Indépendance

2.2.1.1 Meilleure maîtrise en interne

Il existe plusieurs façons d'envisager un projet informatique en milieu professionnel. On peut choisir d'externaliser la majeure partie du projet (développement, intégration et support) en le confiant à une ou plusieurs sociétés ou on peut choisir de s'appuyer sur les ressources internes pour la majeure partie du projet (au besoin en faisant appel à des formations adaptées).

Comme décrit dans le préambule page 1, la réalité est en général plus complexe et fait apparaître un mélange de ces deux solutions. En effet, un projet entièrement externalisé finira malgré tout par faire naître en interne des compétences propres à ce projet alors qu'un projet interne aura tendance à s'essouffler s'il ne trouve pas de débouchés vers l'extérieur à un moment donné.

Le fait de baser une solution logicielle sur des briques de base propriétaires ne permettra jamais de maîtriser totalement en interne ce que fait cette solution et comment elle fonctionne. Il restera toujours une dépendance forte vis à vis du ou des éditeur(s) de ces briques. Le fait de développer entièrement en interne une solution logicielle est très rarement envisagé en raison du coût que cela représente. Dans ce contexte, le logiciel libre représente une réelle opportunité de s'approprier des technologies sans pour autant avoir à en supporter les coûts de développement en partant de rien.

L'AFT, qui est en charge de la gestion de la dette et de la trésorerie de l'État, est cliente de multiples solutions logicielles dont la principale est un progiciel vendu par un éditeur. Ce modèle présente des contraintes en matière d'évolution du produit (seul l'éditeur est à même de le faire) ainsi qu'en terme de souplesse sur l'évolution des briques logicielles sous-jacente à cette solution (SGBD, système d'exploitation, librairies d'impression, etc. . .). Il est toutefois possible d'effectuer du paramétrage qui permet des modifications simples dans le produit (extraction automatique de données, personnalisation de certains traitements de type batch etc. . .). Plus récemment, des solutions logicielles libres basées sur la technologie Linux Apache MySQL PHP (LAMP) ont été déployées de façon autonome par l'équipe informatique de l'AFT. Parmi ces solutions, certaines ont été utilisées sans modification, d'autres ont nécessité des développements internes supplémentaires. Elles ont permis de compléter le progiciel existant (supervision du produit, interface d'accès aux données, alertes par messagerie) ou de couvrir de nouveaux besoins (Gestion Électronique de Documents (GED), Gestion de la Relation Client (GRC)). Il existe également une troisième catégorie de projet, le développement pour un besoin propre à l'AFT (la gestion des annonces de certains correspondants du Trésor) qui a été effectué par une société externe en se basant sur des briques logicielles libres. Le support et les évolutions fonctionnelles ont ensuite été pris en charge par l'équipe interne. Ce développement a récemment été présenté aux gestionnaires de trésorerie d'une autre institution publique qui a envisagé de l'acheter pour son propre besoin.

2.2.1.2 Utilisation de normes plutôt que de standards

De la même manière que dans le monde du logiciel, il existe différentes manières de voir les formats de données utilisés par les logiciels (format de fichier et protocole de transfert principalement). Ces formats peuvent être issus de normes, ce qui signifie que leur spécification est librement disponible, a été longuement débattue et a fait l'objet d'un consensus par les acteurs du secteur qu'elle concerne. Une norme est généralement éditée par un organisme de normalisation qui prend en charge sa mise au point et en assure la promotion. A contrario, un standard est une technique (dont les spécifications ne sont pas publiques) dont l'emploi s'est répandu par l'usage et qui est de ce fait devenu prépondérant. Un standard reste donc entre les mains d'un

acteur et son avenir dépend uniquement de lui.

Nous pouvons donc aisément faire le parallèle entre un logiciel propriétaire et un standard et entre un logiciel libre et une norme. Il est d'ailleurs intéressant de constater qu'un logiciel propriétaire ne respecte généralement une norme que lorsque celle-ci est incontournable (ex : le respect de la norme définissant le fonctionnement du protocole [HTTP](#) est incontournable pour un navigateur web propriétaire tel qu'Internet Explorer). A contrario, un logiciel libre s'attache généralement à utiliser les normes existantes.

Prenons pour exemple les suites bureautiques pour lesquelles le choix du format de fichier est un élément primordial pour leur bonne intégration dans un écosystème existant. Le standard du marché est aujourd'hui Microsoft Office et ses formats associés (doc, xls et ppt principalement) dont il n'existe pas de spécifications publiques détaillées. Les autres suites bureautiques du marché, qu'elles soient libres ou non, sont contraintes de proposer des convertisseurs permettant de lire ces formats étant donné qu'ils sont les plus répandus. Le problème auxquels sont alors confrontés les utilisateurs de ces suites « alternatives » est que ces convertisseurs sont imparfaits et obligent à refaire une partie de la mise en page des documents. Ces difficultés proviennent du fait que les formats des fichiers de Microsoft ne sont pas décrits de manière précise et qu'ils ne peuvent être utilisés comme format natif pour les autres suites. Pour pallier ce problème, l'Organization for the Advancement of Structured Information Standards (OASIS)² s'est donné pour mission de définir la norme Open Document Format for office applications (ODF) qui deviendra à l'avenir un format supporté nativement par l'ensemble des suites bureautiques. Microsoft a d'ailleurs fait l'annonce que la mouture 2013 de sa suite Office supporterait nativement la norme ODF. Son principal challenger, LibreOffice, supporte cette norme depuis de nombreuses années.

Tout ceci concourt à une meilleure interopérabilité des logiciels entre eux et à une plus grande pérennité des données pour les utilisateurs. Les logiciels libres cherchant par essence à se conformer à des normes, cela leur donne naturellement un avantage. Par exemple, en cas d'abandon d'un logiciel libre dont le format de fichier est basé sur une norme, d'autres logiciels sauront aisément exploiter ces fichiers. Si ce n'est pas le cas, les spécifications de ce format étant publiques, il sera alors possible de développer un nouveau logiciel prenant en charge ce format sans passer par des méthodes dites d'« ingénierie inversée » coûteuses et souvent illégales.

2.2.1.3 Souveraineté technologique de la France

Une autre manière de parler d'indépendance en matière de SI concerne l'intérêt de la Nation. En effet, dans un monde où la concurrence entre les pays est très présente, l'hypothèse de voir des données sensibles être dévoilées à certaines puissances étrangères pose question. Que se passerait-il si certains pays d'Asie avaient accès aux secrets de fabrication de certains grands industriels français ou encore aux investissements que le gouvernement prévoit de faire dans le domaine militaire ces 20 prochaines années ? Pire, l'hypothèse d'une attaque contre nos moyens de communication modernes (réseaux IP et donc les informations véhiculées³) paralyserait la société française dans son ensemble.

2. L'OASIS est un consortium mondial créé en 1993 qui regroupe un très grand nombre de sociétés informatique ou non (ex : Adobe, Hewlett-Packard, IBM, Microsoft, Oracle, Airbus, Boeing...)

3. Communications électroniques, téléphonie, flux vidéo, etc.

Une illustration frappante est la proposition faite par le sénateur Jean-Marie Bockel d'interdire à l'échelle européenne les équipements réseau « de cœur » d'origine chinoise (en particulier les routeurs). En effet, les équipementiers chinois vendent ce type matériel à perte (probablement grâce à des aides de leur gouvernement), ce qui ne peut que nous inciter à la prudence. Par ailleurs, toute demande de collaboration de la France avec la Chine en matière d'attaques informatiques provenant de leur territoire est systématiquement rejetée (source : [30]).

Si ces sujets ne tournent pas qu'autour du logiciel, la majorité des cyber-attaques proviennent de l'exploitation malveillante de failles dans les logiciels (qu'ils soient intégrés au matériel sous forme de *firmware* ou pas). Les équipements réseau tels que les routeurs intègrent du logiciel en leur sein et c'est souvent dans ce dernier que se trouvent des risques de fuite, d'altération ou de suppression de données. Une solution pour pallier ce problème pourrait être de demander au fabricant de livrer le logiciel et son compilateur pour son équipement sous forme de *code source* afin que l'utilisateur du produit puisse contrôler que cet ensemble logiciel ne contient pas de *porte dérobée* ou de faille connue. Cette solution doit tout de même être modérée par le fait que des risques peuvent exister au niveau du matériel et pas seulement logiciel. Dans ce cas, l'accès au *code source* offre une bien meilleure visibilité sur le comportement du système mais ne peut être considéré comme suffisante pour se prémunir totalement de comportements malveillants.

Loin de la polémique lancée par le Canard Enchaîné le 17 avril dernier sur la reconduction d'un contrat entre le ministère de la Défense et Microsoft pour 4 ans dans un article intitulé « L'armée capitule face à Microsoft » (source : [10]), il est tout de même important de souligner que l'utilisation massive des produits Microsoft dans des fonctions régaliennes telles que la défense nationale comporte des risques. En effet, d'après cet article qui cite un document interne du ministère de la Défense, la *National Security Agency (NSA)* introduirait systématiquement des *portes dérobées* dans les logiciels exportés, rendant ainsi l'armée française vulnérable à son égard. Dans un tel contexte, le choix du logiciel libre aurait permis au ministère d'avoir une meilleure maîtrise du comportement des logiciels qu'il utilise (source : [42]).

Là encore, le logiciel libre présente un avantage certain par rapport au logiciel propriétaire. Grâce à la libre diffusion du *code source*, tout un chacun peut s'assurer avant de l'utiliser qu'il ne comporte pas de *porte dérobée*. Par ailleurs, la relecture du *code source* par un grand nombre de personnes assure une plus grande couverture des failles courantes. En revanche, un piège serait de faire confiance à des versions déjà *compilées* disponibles sur internet et de ne pas s'assurer que ces versions correspondent bien au *code source* librement accessible. Un autre inconvénient du logiciel libre est que le *code source* permet parfois à des pirates de découvrir des failles qui n'auraient pas été visibles sans l'accès à celui-ci. Ces derniers cas restent cependant rares.

Pour illustrer le fait qu'un logiciel libre peut avoir été modifié si on ne prend pas garde à la source à partir de laquelle il a été *compilé*, une alerte a récemment été publiée par le *Centre d'Expertise gouvernemental de Réponse et de Traitement des Attaques informatiques (CERTA)* concernant une version modifiée d'Apache comportant une *porte dérobée*. En effet, une société spécialisée dans la sécurité a récemment découvert des versions *binaires* modifiées d'Apache ne provenant pas de la *compilation* de versions dont le *code source* est disponible librement. Il résulte de cette tromperie que ces versions *binaires* contiennent une *porte dérobée*. Heureusement, il existe des solutions simples permettant de vérifier l'authenticité des versions *binaires* téléchargées grâce à des algorithmes de vérification de l'intégrité des fichiers (ex : *Secure Hash Algorithm (SHA)*). Cet exemple montre néanmoins qu'il est illusoire de faire aveuglément confiance en un logiciel pré-compilé simplement parce que l'on a accès à son *code source* par ailleurs. Seule une *compilation* de ce *code source* en un *binaire* nous assure que le comportement du programme sera conforme à celui présent dans le *code source*. (source : [12])

Aujourd'hui, les projets sur le thème de la sécurité informatique utilisent prioritairement des produits libres pour ces raisons d'indépendance vis à vis d'éditeurs de logiciels étrangers et pour des raisons de meilleure maîtrise du comportement des produits.

En reprenant un projet universitaire nommé CLIP, l'Agence Nationale de Sécurité des Systèmes d'Information (ANSSI) a récemment développé un système d'exploitation sécurisé de haut niveau. Ce système basé sur Linux permet, sur un même poste de travail, d'avoir un accès à internet non bridé et de bénéficier d'un environnement sécurisé via un Virtual Private Network (VPN) d'entreprise par exemple. Ces deux environnements sont totalement étanches et s'exécutent simultanément sur une même machine. Un des objectifs de ce projet est de permettre l'accès à des informations sensibles sur des ordinateurs portables de l'administration en situation de nomadisme (source : [3]).

Le projet Trustedbird, lancé en 2008 par la Direction Générale de l'Armement (DGA) avec le concours de British Telecom (BT) a pour objectif de fournir une solution complète de messagerie sécurisée à l'administration française. Pour y parvenir, ce projet est conçu comme un fork temporaire du produit libre Thunderbird y apportant de nombreux éléments de sécurisation (sous la forme de modification du produit, de modules complémentaires et de scripts serveurs). Les équipes qui travaillent sur ce projet collaborent étroitement avec la fondation Mozilla afin que leur développements puissent être, à terme, intégrés dans Thunderbird. Cette collaboration est rendue possible par la mise à disposition sous licence libre de l'ensemble de ces travaux. (sources : [39][27])

2.2.2 Innovation

2.2.2.1 Dans le monde du logiciel propriétaire

Historiquement, le modèle économique du logiciel se base sur les revenus générés par la vente de licences. À une époque, celui-ci a permis de faire émerger des produits qui sont aujourd'hui devenus incontournables. Ceci dit, parmi les sociétés qui ont choisi ce modèle économique, peu nombreuses sont celles qui ont récemment proposé de réelles innovations. En effet, les grosses sociétés qui possèdent les produits incontournables continuent de les faire évoluer et leur apportent toujours plus de fonctionnalités et d'optimisation mais aujourd'hui, ce modèle s'essouffle. Parmi ces sociétés, nous pouvons citer Oracle, Microsoft, IBM...

La virtualisation, qui représente une avancée majeure en informatique, est un contre-exemple à cette règle. En effet, la société VMware propose les meilleures solutions du genre actuellement disponible pour les professionnels sous la forme de logiciels propriétaires. Il existe cependant des solutions de virtualisation libres mais moins abouties (VirtualBox par exemple).

Dans certains cas bien spécifiques, il n'existe pas de logiciel libre répondant aux besoins des utilisateurs. C'est par exemple le cas en finance de marché ou aucune communauté ne s'est formée pour offrir une solution répondant aux problématiques de ce secteur. Dans ce cas, notamment à l'AFT, deux solutions sont possibles : la plus sûre est d'acheter un progiciel existant et de l'adapter aux besoins précis de la salle de marché ; la deuxième, plus originale et ambitieuse en termes de ressources humaines, serait d'effectuer un développement spécifique en interne. Se poserait alors la question d'en libérer le code source afin de le partager avec des institutions similaires.

2.2.2.2 Dans le monde du logiciel libre

En matière d'innovation, il est difficile d'affirmer que le logiciel libre est aujourd'hui un moteur. Si certains projets ont pu être pionniers à une époque (notamment au tout début d'internet dans les années 80), aujourd'hui ils se contentent souvent de reprendre des fonctionnalités d'un produit propriétaire existant. Cependant, lorsqu'un logiciel libre devient très populaire, il arrive souvent qu'il dépasse son modèle initial en terme d'ergonomie, de fonctionnalité ou encore de performance mais force est de constater qu'il ne s'agit pas véritablement d'innovation.

Aujourd'hui, la majorité du volume d'affaire généré par le logiciel libre est issue des contrats de maintenance. Or ces contrats permettent d'améliorer la qualité des logiciels en corrigeant les anomalies découvertes dans ces derniers, mais ne permettent pas réellement de promouvoir l'innovation.

Les systèmes de Wiki, devenus populaires depuis le début des années 2000 avec le lancement de Wikipédia, sont un bon contre-exemple de cette affirmation. En effet, de nombreux produits libres (dont le plus populaire est MediaWiki) existent aujourd'hui et représentent une catégorie de logiciels à eux seuls. Ces systèmes permettent l'édition d'articles en ligne de manière collaborative en les classant par catégorie. Un des atouts est également l'intégration d'un puissant moteur de recherche permettant de retrouver les contenus très facilement.

2.2.2.3 Dans le monde internet

Depuis le milieu des années 90, début de l'ère internet « grand public », les innovations viennent le plus souvent de petites entreprises (nommés *start-up*) qui s'attachent à développer un concept nouveau de service en ligne et à lui trouver un modèle économique viable. Certaines sont aujourd'hui devenues de très grandes entreprises ayant un rôle majeur dans le secteur (Google, Yahoo!, Amazon, Facebook,...).

Depuis plus d'une dizaine d'années, nous assistons à des rachats de ces petites sociétés innovantes par de plus importantes. En effet, alors que le modèle d'innovation fonctionne pour une petite société, cela s'avère moins vrai pour une société de taille importante. C'est ainsi que la plupart des grandes entreprises du secteur proposent des solutions innovantes essentiellement grâce au rachat de *start-up*.

Alors que l'innovation s'est déplacée ces dernières années du monde du logiciel vers celui d'internet, nous verrons au 3^e chapitre comment ce modèle d'innovation issu de la mouvance internet rencontre aujourd'hui et rencontrera demain celui du logiciel.

2.2.3 Macro-économie

2.2.3.1 Où va l'argent du logiciel ?

Nous sommes en droit de nous poser cette question étant donné les sommes dépensées par l'administration qui se chiffrent en centaines de milliers, voire en dizaines de millions d'euros (et parfois plus pour des projets de très grande envergure comme Chorus⁴). Aujourd'hui, force est de constater que les éditeurs des logiciels les plus utilisés dans le milieu professionnel sont majoritairement américains (Microsoft, IBM, Oracle, SAP,...) (source : [43]). Cet état de fait cumulé avec le constat de forte rentabilité sur des produits qui évoluent peu (cf. paragraphe 2.1.1.3 page 19) amène à se demander si l'argent ainsi dépensé par le secteur public français

4. Le projet Chorus est un projet de grande envergure de refonte de l'informatique comptable de l'État sur 5 ans. Il permet la gestion de la dépense et des recettes non fiscales selon les dispositions contenues dans la LOLF. Ce programme était à l'origine piloté par la DGCP, devenue depuis la DGFIP

influe positivement sur l'économie du pays.

L'histoire du projet LemonLDAP n'est pas banale. En effet, en 2006, la DGI était cliente d'une solution d'authentification et d'habilitation en environnement web (Signle Sign-On (SSO)) proposée par la société Novell. En 2007, alors que le ministre du Budget de l'époque annonce la fusion de la DGI avec la DGCP, les informaticiens de la DGI découvrent la solution homologue utilisée par la DGCP qui est en fait un développement interne fonctionnant sous Linux mené depuis 2003 par Eric German (agent de la DGCP) qui sera distribué sous licence libre par la suite. À l'époque, cette solution répond à des besoins bien moins ambitieux que celle de Novell à la DGI. La DGFIP, issue de la fusion des deux directions, optera alors pour la solution interne en remplacement de celle fournie par Novell. Le résultat de cette opération sera une économie importante pour l'administration sur le contrat de maintenance, une meilleure prise en compte des besoins pour les développements futurs qui seront effectués en interne, et une meilleure fiabilité de la plateforme finale (pour des raisons techniques). Ce projet, auquel la Gendarmerie nationale et le ministère de la Justice participent depuis 2004, fera l'objet d'un fork en 2006. En effet, la Gendarmerie associée à la société Linagora rajoutera l'acronyme Nouvelle Génération (NG) à son nom, en fera un produit plus complet en terme de fonctionnalités et en élargira la communauté. Aujourd'hui, le plus gros contributeur à ce projet est la société Linagora qui vend des prestations d'intégration de cette solution « clé en main » à de nombreuses entreprises.

Parallèlement à cela, l'argent investi dans le logiciel libre, principalement au travers des contrats de maintenance, génère de l'activité localement. En effet, si le développement d'un logiciel est « délocalisable », les activités d'intégration et de support nécessitent d'avoir la majorité des interlocuteurs sur place.

Cependant, tout ne peut pas être considéré comme parfait dans le modèle du logiciel libre. En effet, les coûts de développement échappent à toute comptabilisation analytique et il n'est pas possible de savoir qui les supporte vraiment. Nous l'avons vu, dans certains cas l'administration les supporte en interne en dédiant des ressources humaines à ces travaux. Il est alors bien difficile de comparer financièrement le coût de cette solution à une autre. De même, certains acteurs de communautés travaillent à des projets sans être rémunérés pour cela, ce qui ne va pas dans le sens d'une juste rétribution des acteurs en fonction du travail fourni.

2.2.3.2 Externalités dues aux logiciels libres

Une externalité est une influence induite par l'action d'un agent économique (mais ce n'est pas son objectif premier) sur d'autres agents sans pour autant qu'ils n'aient donné leur accord ou en reçoivent une quelconque compensation. Une externalité peut être positive : Par exemple un apiculteur dont le but est de produire du miel induit une pollinisation des arbres alentour pouvant être considérée comme une externalité positive par un arboriculteur ; ou être négative : La pollution des gaz d'échappement des voitures qui est subie par la population et provoque des problèmes de santé.

Le logiciel libre est considéré par les économistes comme une externalité positive⁵. Cela signifie que les agents économiques qui les utilisent bénéficient de retombées indirectes de ces logiciels (meilleure productivité, accès à la connaissance. . .) sans pour autant avoir à en supporter directement les coûts. Les économistes nous apprennent que le rôle d'un État est de favoriser les actions ayant des externalités positives (par des incitations fiscales ou des subventions par exemple) et de décourager celles ayant des externalités négatives (en les taxant par exemple) afin de rééquilibrer les bénéfices ou les inconvénients produits par ces actions sur les citoyens. Dans ce cadre, la propagation des logiciels libres peut être vue comme une mission pour l'État.

5. d'après « *the success of open source* » Steven Weber, 2004 Harvard University Press.

2.2.3.3 Effet sur les entreprises

Comme nous l’avons vu au paragraphe 2.2.3.1, l’argent investi dans le logiciel libre l’est le plus souvent localement, et permet aux entreprises qui en bénéficient d’être compétentes sur des produits qu’elles maîtrisent totalement plutôt que d’être en situation de dépendance vis-à-vis de la politique commerciale de grands éditeurs de logiciels propriétaires. Jacques Attali, dans les travaux de la commission pour la libération de la croissance française qu’il a dirigés, nous livre une vision intéressante du logiciel libre en soulignant la nécessité de « *ne pas rater le train du numérique* » et de « *renforcer le secteur du logiciel* ». Il va même jusqu’à suggérer que « *les aides aux communautés de logiciels libres devraient être considérées fiscalement comme du “mécénat de compétence”* » (proposition 58 du document [4]).

Enfin, de façon plus macroscopique, si les logiciels libres sont un levier pour réduire la dépense publique, ils permettent alors mécaniquement d’améliorer la compétitivité des entreprises françaises en réduisant la pression fiscale qui s’exerce sur ces dernières. Cette affirmation a une portée bien limitée, soyons-en conscients, mais en réalité, une meilleure appropriation des logiciels libres par les sociétés de services informatiques françaises leur permettra demain de proposer des solutions professionnelles éprouvées au secteur privé ce qui aura pour effet, à terme, une réduction des coûts informatiques pour ces entreprises.

Conclusion

Tout au long de ce chapitre, nous avons pu constater que le secteur public français a une appétence croissante pour le logiciel libre. En effet, celui-ci répond au besoin légitime de réaliser des économies en permettant la mutualisation des efforts entre différents acteurs publics ayant des besoins similaires et de stimuler une saine concurrence entre les acteurs du secteur privé, conformément aux objectifs d’un marché public. Par ailleurs, ces logiciels démontrent de grandes qualités dans leur ensemble même si elles peuvent être différentes de celles des logiciels propriétaires. Enfin, le libre permet de garantir une meilleure indépendance vis-à-vis d’un fournisseur et des autorités du pays dont il dépend et concourt à un meilleur retour de l’argent injecté par l’administration sur des acteurs nationaux (ou européens) plutôt que sur des sociétés le plus souvent nord-américaines. Il s’agit donc d’une voie alternative dans laquelle la France a tout intérêt à persévérer comme le soulignent les travaux de la commission Attali de 2008.

Dans cette démarche, certains sont très avancés (comme la Gendarmerie qui a fait de ce sujet une priorité depuis 2002, avec succès), quand la plupart ont réussi cette mutation pour leurs infrastructures centrales mais peinent à la mettre en œuvre pour les postes de travail. D’autres encore, comme Pôle emploi, ont une approche très rationnelle de la chose et bascule les solutions au coup par coup, uniquement quand cela présente un gain financier important. Quel que soit l’avancement de ces travaux et la méthode utilisée, il semble que l’adoption des logiciels libres soit une démarche souhaitée par beaucoup et qui progresse un peu partout.

Pour terminer, nous pouvons citer la ministre déléguée en charge de l’Économie numérique, Fleur Pellerin, qui a déclaré, à l’occasion de sa visite dans les locaux de la fondation Mozilla à Paris le 13 juin dernier, que le logiciel libre est « *un véritable garant de la souveraineté numérique* », qui « *remet en cause les rentes de situation* » (source : [47]).

Chapitre 3

Perspectives à court et moyen terme

Introduction

Nous avons pu le constater, le logiciel libre est très présent dans le secteur public et son adoption se fait lentement mais sûrement au sein des administrations. Ce modèle atypique dont le secteur public est très demandeur a donné naissance à une offre de service portée par des entreprises. Nous verrons comment est structuré ce marché et ce que l'on peut en attendre à l'avenir. Ensuite nous nous attarderons sur les dérivés du logiciel libre. En effet, cette mouvance a donné naissance à un grand nombre d'initiatives de partage d'informations ou de ressources bien au delà du `code source` d'un logiciel. Parmi elles figure l'OpenData, une formidable promesse de transparence du secteur public à l'égard du citoyen.

Une tendance qui apparaît depuis quelques années dans le paysage est celle de l'informatique en nuage (*Cloud computing*). Certains prédisent même la mort du logiciel tel que nous le connaissons aujourd'hui au profit de cette masse vaporeuse (un joli rêve qui mérite qu'on s'y attarde). Nous verrons donc les différents types de *cloud*, les logiciels sur lesquels ils reposent ainsi que les perspectives ouvertes par le logiciel libre dans ce domaine. Nous terminerons enfin par les offres de *cloud* disponibles en France et nous nous attarderons en particulier sur les raisons qui ont poussé les politiques à faire intervenir l'État dans ce que l'on appelle le *cloud* « souverain ».

3.1 Le libre, une tendance qui monte

3.1.1 Le marché du logiciel libre en croissance constante depuis 10 ans

3.1.1.1 État des lieux en France

Depuis le début des années 2000, le secteur du logiciel libre représente une part croissante du marché des logiciels et des services informatiques en France. En 2012, cette part de marché s'établissait à 6 % pour un volume de 2,5Md€ avec une très forte perspective de croissance en 2013 (source : [37]). Ce secteur emploie 30 000 personnes réparties dans 300 PME et TPE avec près de 1 500 nouvelles embauches attendues en 2013. Cependant, les talents dans ce domaine se font rares et 74 % des entreprises spécialisées du secteur déclarent avoir un mal à recruter (source : [9]).

Il est intéressant de regarder la répartition sectorielle de ce marché. En 2007, les administrations représentaient à elles seules 44 % pour un volume total de 320M€. Ce positionnement de la France comme l'un des fleurons du logiciel libre n'est donc pas un hasard. L'engagement massif du secteur public dans cette voie a permis de faire émerger des Sociétés de Services en Logiciel Libre (SSLL) de taille moyenne ainsi que des compétences dans ce domaine dans les

grandes Sociétés de Services en Ingénierie Informatique (SSII) françaises (sources : [40][45]).

Un phénomène intéressant dû au logiciel libre est celui du passage du statut de consommateur à celui d'acteur (ou éditeur) pour une société de service. En effet, de nombreuses entreprises du secteur (comme Linagora dont nous avons parlé à propos des projets OBM et LemonLDAP) peuvent aussi être considérées comme « éditeurs » de logiciel alors que cette activité n'aurait pas été possible si elles s'étaient contentées d'intégrer des logiciels propriétaires. C'est ainsi que les missions et métiers de ces sociétés sont amenés à évoluer si elles souhaitent conquérir des parts de marché dans le secteur du libre.

Le récent projet de loi « d'orientation et de programmation pour la refondation de l'école de la République » prévoit que « [L'école] utilise en priorité des logiciels libres et des formats ouverts de documents ». Une affirmation qui fait bondir le Syntec numérique et l'Association Française des Industriels de l'Éducation et de la Formation (AFINEF) qui considèrent qu'il s'agit d'une « atteinte au principe de neutralité technologique ». En effet, dans le domaine du logiciel éducatif, plusieurs sociétés éditrices de logiciels ne voient pas l'arrivée du libre dans les écoles d'un bon œil. (source : [32])

Selon une étude réalisée par la société américaine RedHat¹ en 2010, la France serait même numéro 1 mondial de l'open-source. Cette étude se base sur un certains nombre de critères gouvernementaux (politique et usage des logiciels libres dans l'administration), industriels (logiciels dans les entreprises, nombre de développeurs libre actifs. . .) et communautaires (couverture par les médias, formations au libre. . .) (source : [36]). Au niveau international, le Canada et la Grande-Bretagne ont d'ailleurs pris début 2013 des mesures pour favoriser le logiciel libre dans le secteur public (sources : [28][26]).

3.1.1.2 Un modèle économique encore perfectible

Malgré cet engouement autour du logiciel libre, force est de constater que le modèle est parfois fragile et ne rétribue pas toujours ceux qui lui apportent le plus. Si l'activité économique se développe et crée des emplois autour des services (prestations d'intégration et de support), il n'en reste pas moins que le travail des développeurs se fait sans visibilité et sans rétribution directe. Dans cette situation, les acteurs « historiques » de ce marché (RedHat, SuSE, Linagora, . . .) ont en général bien compris que leur intérêt à long terme est de contribuer le plus possible à ces projets afin d'acquérir les compétences nécessaires aux prestations vendues sur ces produits. Par ailleurs, l'expertise acquise à l'occasion de ces contributions leur permet de proposer plus facilement à leurs clients l'ajout de fonctionnalités (et d'être en position de force pour les faire intégrer par les communautés). Les nouveaux acteurs de ce marché en revanche (grandes SSII françaises) ont plutôt une vision à court terme et sont en général plus réticents à investir dans le développement. Ils estiment, comme ils le feraient pour des prestations sur des logiciels propriétaires, que ce qui leur est demandé se limite strictement à du support. Dans ce cas, les contributions apportées par ce type de sociétés aux projets se limitent à des corrections de *bugs* mais ne les font pas progresser sur le plan fonctionnel.

Un exemple à forte visibilité qui illustre cette problématique du financement des développements du logiciel libre est celui de « The Document Foundation », la communauté qui supporte le développement de LibreOffice, qui est actuellement à la recherche de sponsors à hauteur de 1M€ pour financer le développement de LibreOffice en ligne. En effet, à l'heure où Microsoft lance à grand renfort de publicité Office 365, une déclinaison de sa suite bureautique orientée cloud, et que Google propose un service équivalent depuis 2006 (Google Documents), la fondation peine à faire de ce projet une priorité faute de moyens.

1. RedHat est le plus grand éditeur et fournisseur de services au monde dans le domaine de l'open-source. En 2012 il comptait environ 5 500 collaborateurs pour un chiffre d'affaires de 1Md\$.

En revanche, le modèle voulu par l'ADULLACT semble un compromis équilibré qui permet d'éviter les développements redondants (et donc les coûts associés) dans chaque structure publique locale. Lorsque une collectivité identifie un besoin métier qui nécessite d'acquérir un logiciel, elle peut s'adresser à l'ADULLACT qui réunira alors toutes les structures publiques ayant le même besoin afin de cofinancer le développement de ce logiciel. Le fait que le résultat de cette commande soit ensuite mis sous une licence libre permettra d'en faire bénéficier simplement et à moindre coût une nouvelle structure qui en aurait besoin. Cette collaboration est rendue d'autant plus facile qu'il n'existe pas de concurrence entre les structures publiques et qu'il paraît légitime à tout un chacun d'éviter le gaspillage d'argent public. Ceci explique sans doute pourquoi le secteur public français est le premier client de solutions logicielles libres et tire aujourd'hui ce marché vers le haut. Il existe cependant des exemples de sociétés privées (parfois concurrentes sur le terrain des prestations) qui collaborent au travers de projets libres.

3.1.2 Le libre se propage au delà du logiciel

3.1.2.1 Les inspirations les plus populaires

Le logiciel libre, par son modèle utopiste et généreux, a fait naître un très grand nombre d'initiatives dans son sillage. Parmi elles, beaucoup n'ont pas trouvé le moyen de s'imposer (musique libre, art libre, conception de matériel libre, modèle industriel libre...) mais certaines ont aujourd'hui largement passé le cap de maturité nécessaire à leur essor. Voici quelques exemples de réussites dans ce domaine :

Les licences Creative Commons permettent à l'auteur d'une œuvre de la diffuser librement à la manière du code source d'un logiciel libre. Pour cela, l'auteur doit choisir une combinaison de 3 critères simples qui détermineront les conditions de reproduction et d'utilisation de son œuvre. Aujourd'hui un très grand nombre de documents sont diffusés via ces licences (livres, manuels scolaires, articles en tout genre, contenus audio et vidéo sur internet...).

Par exemple, le document que vous êtes en train de lire est diffusé sous la licence Creative Commons avec les restrictions BY-NC-SA (cf. logo sur la couverture) ce qui signifie que toute personne qui souhaite utiliser tout ou partie de ce document doit en citer l'auteur (BY), que personne ne peut utiliser tout ou partie de ce document pour un usage commercial (ex : en insérer des extraits dans un livre rapportant des royalties) (NC) et enfin que toute personne qui souhaite utiliser tout ou partie de ce document dans un autre doit publier le résultat sous la même licence que le document initial (SA).

L'encyclopédie en ligne Wikipédia est incontestablement la réussite la plus visible du grand public et la plus populaire. Parti de l'ambition de construire une encyclopédie que chacun pourrait enrichir dans le domaine qu'il connaît le mieux, Wikipédia est aujourd'hui devenu une base de connaissance de plus de 4,2M d'articles en anglais et de près d'1,4M d'articles en français sur tous les sujets imaginables. Il est cependant nécessaire de rester méfiant à l'égard de son contenu et en particulier sur des sujets pointus pour lesquels le nombre de contributeurs est souvent faible. Wikipédia repose sur la fondation Wikimédia dont le but est de promouvoir des initiatives autour du partage de connaissance. L'ensemble des contenus publiés sur Wikipédia sont sous licence Creative Commons (BY-SA 3.0).

Le projet OpenStreetMap propose de construire une cartographie mise à jour continuellement par ses utilisateurs. Plutôt que de laisser des acteurs du marché construire de coûteuses cartographies numériques dont la fréquence des mises à jour en un point donné est très faible, l'idée est que chaque utilisateur puisse signaler des modifications (changement de sens d'une voie par exemple). Pour la France, le contributeur initial à ce projet est le plan cadastral informatisé

mis à disposition par la DGFIP. L'ensemble des données et la documentation du projet sont sous licence Creative Commons (BY-SA 2.0).

3.1.2.2 L'OpenData en marche

La mouvance OpenData est née aux États-Unis à la fois d'une défiance des citoyens vis-à-vis de l'action publique (volonté de contrôler ce qui est fait) et d'une volonté de disposer des données produites par le secteur public sans avoir à les (re)payer (i.e. autrement que par la fiscalité). En France, cette tendance est apparue en 2009 (les villes de Paris, Brest et Rennes ont été les premières) et a plutôt pour objectif de stimuler la croissance des entreprises qui utilisent ces données pour offrir des services innovants aux citoyens. Au niveau national, la mission Etalab a été créée en 2011 par François Fillon pour fédérer et faciliter les initiatives de mises à disposition de données du secteur public derrière un portail web unique data.gouv.fr.

L'idée première de cette démarche est d'offrir de la transparence au citoyen. Dans une démocratie moderne, la puissance publique se doit de rendre des comptes au citoyen-contribuable qui finance son action. L'OpenData est un moyen d'y parvenir en offrant en libre accès les données produites et/ou collectées par les administrations. Mais cette libération des données ne doit pas se faire n'importe comment et au mépris de certaines libertés individuelles. Des comités d'éthiques sont par exemple mis en place dans certaines collectivités locales afin de déterminer si tel ou tel type d'archives municipales peuvent être rendue publique ou pas.

En procédant ainsi, la puissance publique entend livrer un produit brut (des données) que les citoyens et les entreprises pourront s'approprier pour créer de nouveaux usages. Il s'agit donc d'une sorte de partenariat public-privé dans lequel la collectivité se contente de mettre à disposition des informations non traitées (avec un coût minime pour elle) en laissant au secteur privé le soin d'innover autour des usages qu'il voudra bien en faire. L'exploitation de ces données est libre de droits (y compris pour un usage commercial) et une licence concise de 3 pages a d'ailleurs spécialement vu le jour à cet effet : la Licence Ouverte ou *Open Licence* (document : [19]). Les pouvoirs publics attendent des retombées économiques de cette libération des données.

Parmi les exemples d'utilisation des données publiques pour la vie quotidienne, nous pouvons citer les horaires des transports publics (pouvant par exemple permettre d'imaginer une application pour *smartphone* croisant des informations de la RATP, la SNCF et de la Direction des Routes d'Île-de-France (DiRIF) afin d'offrir la combinaison multimodale optimale lors d'un déplacement), les résultats des élections par communes/arrondissements (permettant par exemple à la presse d'information locale de proposer un « zoom » sur la zone géographique souhaitée), les indicateurs de résultats des lycées produits par l'éducation nationale, les données statistiques produites par l'INSEE, etc.

Un exemple d'initiative autour de l'ouverture des données publiques est celui du concours « Moov' in the city » lancé par la ville de Paris proposant aux participants d'utiliser les données issues en temps réel des SI des opérateurs de transports de la région (RATP, SNCF et Vélib') pour imaginer de nouveaux services web ou applications mobiles innovants. L'un des objectifs de ce concours est de permettre aux usagers d'anticiper les obstacles dans leurs parcours. Les lauréats de ce concours se partageront 25 000 € de prix. Résultats début juillet 2013. (source : moovinthecity.fr)

Comme nous l'avons vu au paragraphe 1.2.1.1 page 12, le principe de gratuité est la norme pour les services publics. L'OpenData n'échappe pas à cette règle et cette volonté a été réaffirmée fin 2012 par Jérôme Filippini, Directeur Général du Secrétariat Général pour la Modernisation de l'Action Publique (SGMAP) lors d'une interview du Journal du Net (source : [14]).

3.2 Du logiciel au nuage, la grande mutation

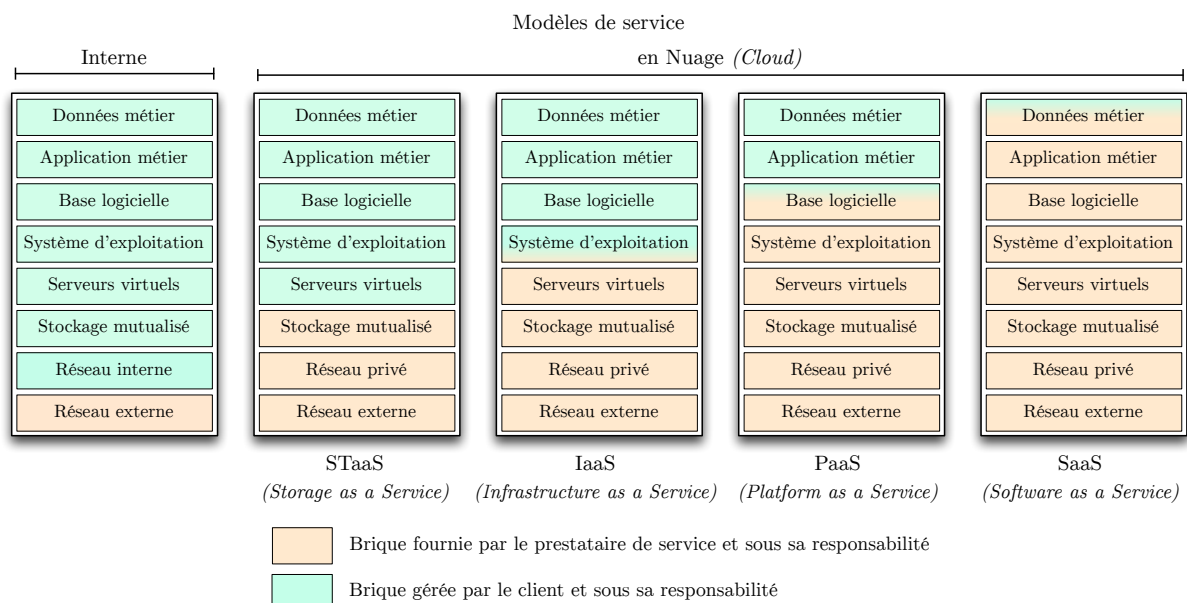
Avec la montée en puissance des réseaux haut débit, les usages de l'informatique changent et les logiciels se muent peu à peu en services en ligne accessible via internet. En effet, aujourd'hui il est de moins en moins question d'installer des logiciels sur un poste de travail mais plutôt d'y accéder avec un navigateur web. Cette nouvelle tendance amorcée depuis bien longtemps porte depuis peu un nom : l'informatique en nuage. Derrière ce terme imagé se cache en réalité quatre concepts bien plus vastes (voir schéma ci-dessous) dont l'idée est d'abstraire les ressources matérielles et/ou logicielles pour se concentrer sur le service rendu à l'utilisateur.

Le modèle économique le plus répandu pour ce type de services est celui de la gratuité avec pour contrepartie une exploitation commerciale des données de l'utilisateur (publicités ciblées, envoi de messages promotionnels...). Dans le monde professionnel, le modèle le plus courant est celui de l'abonnement facturé en fonction du nombre d'utilisateurs ou de la quantité de ressource consommée (volume de stockage utilisé, puissance consommée...). Le marché du *cloud* a connu une croissance de plus de 45 % en 2012 totalisant 2Md€ de chiffre d'affaires en France ce qui en fait l'un des enjeux majeurs pour l'avenir de l'informatique. Le monde professionnel peine cependant à adopter ce modèle pour des questions de sécurité des échanges, de localisation du stockage des données et les problèmes juridiques qui en découlent (droit applicable en fonction de la nationalité du prestataire hébergeant le service).

Il existe aujourd'hui deux types de *cloud*. Le plus répandu (dont nous avons parlé jusqu'ici) est le *cloud* public. Celui-ci se caractérise par le fait qu'il est accessible publiquement (via internet) et qu'il est opéré par une société spécialisée dans ce domaine. Le *cloud* privé se caractérise par une limitation de l'accès aux services par une seule structure. Celui-ci peut-être soit opéré par une société spécialisée (qui doit être considérée comme « de confiance »), soit directement par la structure utilisatrice. Ce dernier cas de figure apporte une réponse aux problématiques de sécurité et de localisation des données.

3.2.1 Les niveaux de service offerts par le *cloud*

Le schéma suivant regroupe le modèle de service informatique dit « interne » (le plus courant actuellement) et les quatre modèles de services dit « *cloud* » (l'avenir à en croire la tendance actuelle) à savoir Storage as a Service (STaaS), Infrastructure as a Service (IaaS), Platform as a Service (PaaS) et Software as a Service (SaaS) :



3.2.1.1 Stockage en tant que Service

Une première catégorie de *cloud* qui n'a en réalité que peu de points commun avec les trois autres, consiste à offrir du stockage de données en ligne utilisable depuis n'importe quel accès à internet. En milieu professionnel, ce modèle présente un intérêt essentiellement pour une utilisation collaborative et/ou nomade.

L'offre cloud d'Apple nommée "iCloud" propose aux utilisateurs de plusieurs équipements de la marque (Macintosh, iPhone, iPad, ...) de partager instantanément les mêmes données sur tous ces appareils. Par exemple, la prise d'une photo avec un iPhone donnera instantanément la possibilité de la consulter sur un Macintosh de bureau. La société DropBox propose également une offre similaire multiplateforme avec une version orientée entreprise.

3.2.1.2 Infrastructure en tant que Service

Le modèle *IaaS* se distingue du modèle interne (cf. schéma page 37) par le fait que les aspects gestion du réseau, gestion de la plateforme matérielle et gestion de la *virtualisation* des serveurs sont pris en charge par un fournisseur de service. Dans ce modèle, le client s'affranchit de tâches techniques généralement éloignées de son métier et nécessitant du personnel dédié qualifié (en particulier pour le réseau) lui permettant ainsi de se concentrer sur ses applications proches du métier. Le client garde cependant la maîtrise du système d'exploitation sur lequel tourne sa solution ce qui nécessite de conserver des compétences techniques dans ce domaine.

*Le modèle IaaS est aujourd'hui le plus répandu chez les fournisseurs de cloud. C'est notamment le modèle qui a été choisi pour l'offre commerciale actuellement leader sur le marché : "Amazon Web Services (AWS)" dont les prévisions de chiffre d'affaires dépassent les 2Md\$ pour 2013 (source : [16]). À noter qu'Amazon a installé un *datacenter* en Irlande afin de conquérir le marché européen.*

3.2.1.3 Plateforme en tant que Service

Le modèle *PaaS* se distingue du modèle *IaaS* par le fait que le fournisseur prend également en charge le système d'exploitation ainsi que le socle logiciel nécessaire au fonctionnement des applications du client (en général : serveur web, serveur d'application, *SGBD* et outils de développement). Cette facilité permet de s'affranchir de l'administration système et de la gestion des couches basses de son application en laissant le fournisseur s'occuper de gérer les problématiques de montée de version, de patch de sécurité ou de changement de socle technologique tout en garantissant au client que son service sera plus fiable que s'il s'occupait de tout cela lui-même. Pour y parvenir, le fournisseur compte sur l'expertise de ses équipes qui effectuent ce type d'opération pour un très grand nombre de clients et rencontrent donc fréquemment les mêmes difficultés. Ces opérations peuvent même être automatisées dans certains cas permettant encore de faire des économies d'échelle.

Le modèle PaaS est actuellement celui retenu par Microsoft pour son service de cloud "Windows Azure". Il est amusant de noter que, sous la pression des clients de son offre de cloud, Microsoft propose depuis peu d'utiliser Linux comme système d'exploitation et a ouvert une division open-source en charge de l'interopérabilité entre ses produits et les produits libres. Une belle reconnaissance de la suprématie des solutions libres dans les couches basses dites « serveur ».

3.2.1.4 Logiciel en tant que Service

Le modèle *SaaS* n'entre pas vraiment en compétition avec les deux précédents. C'est actuellement le modèle de *cloud* le plus courant, et pour cause, nous l'utilisons tous les jours quand nous accédons à des services en ligne sur internet via un simple navigateur (messagerie web, service de banque en ligne, télédéclaration d'impôts sur le revenu...). Ce modèle va plus loin en permettant de se passer totalement de compétence en informatique mais sa limite réside dans le fait que l'application dont le client a besoin doit être disponible « sur étagère » chez le fournisseur pour que celui-ci puisse le proposer en mode *SaaS*. Or, si cela est tout à fait possible pour des services « classiques » comme la messagerie électronique, ça ne l'est absolument pas pour des applications spécifiques au métier du client.

En suivant ce modèle, Google propose une offre nommée “Google Apps for Business” principalement à destination des PME/PMI n'ayant pas les moyens d'avoir leur propre infrastructure informatique et permettant de disposer d'une messagerie GMail au nom de l'entreprise, d'un agenda partagé, d'un espace de stockage en ligne, d'une suite bureautique en ligne et d'une plateforme de travail collaboratif (vidéoconférence et outils de formation) pour un tarif de base de 40 € par an et par utilisateur. À noter que Google propose également une offre purement STaaS nommée “Google Drive”, une offre PaaS nommée “Google App Engine” et une offre IaaS nommée “Google Compute Engine”.

Dans ce modèle, étant donné que le fournisseur de service gère entièrement les aspects logiciels, l'utilisateur se retrouve dans une situation de dépendance similaire à celle d'un logiciel propriétaire avec son éditeur. Il n'est à priori pas possible d'avoir accès au *code source* d'un logiciel en mode *SaaS* et encore moins d'en contrôler son exécution puisqu'elle se déroule directement chez le fournisseur. Dès 2008, Richard Stallman, le père du logiciel libre, nous alerte sur l'absence totale de contrôle de l'utilisateur sur les logiciels en tant que services (source : [22]).

Pour conclure, il existe une multitude de définitions du *cloud* et leur seul point commun est de proposer une abstraction en utilisant le réseau pour accéder à des ressources distantes « louées ». Si ce modèle paraît très séduisant au premier abord (économie d'échelle sur le matériel chez le fournisseur, réduction des tâches techniques à faible valeur ajoutée chez le client), il reste de nombreuses questions à résoudre comme la confidentialité des données échangées via le réseau puis stockées chez le fournisseur et la législation dont elles dépendent. Par exemple, le *US Patriot Act* accorde à l'État américain un droit de regard sur les données des entreprises américaines et leurs filiales, quelles que soient leur localisation. Autre problème, à l'image du logiciel propriétaire qui nous rend dépendant de son éditeur, le *cloud* nous rend dépendant de son fournisseur avec, en particulier pour le mode *SaaS*, un risque de ne pas pouvoir changer de fournisseur plus important qu'avec une solution logicielle classique (non-réversibilité).

3.2.2 Les solutions logicielles pour le *cloud*

Malgré les apparences, le *cloud* ne va pas faire disparaître le logiciel, bien au contraire. Pour pouvoir mettre en place un *cloud* (qu'il soit privé à l'intérieur d'une organisation ou public chez un opérateur), il est nécessaire de disposer de logiciels d'un genre nouveau permettant de faire abstraction du matériel.

3.2.2.1 Solutions propriétaires

Comme nous l'avons vu au paragraphe 2.2.2.1 page 28, le logiciel propriétaire est aujourd'hui en position de force dans le domaine de la *virtualisation* de système d'exploitation sur des serveurs. Cette brique logicielle est un composant indispensable à la mise en place d'un *cloud* (STaaS mis à part). Aujourd'hui, la société VMWare propose la solution la plus aboutie (nom-

mée “vSphere”) pour créer son *cloud* en mode [IaaS](#) et possède une position dominante sur ce marché (loin devant Microsoft et Citrix).

Parallèlement à cette offre propriétaire, VMWare développe une solution logicielle open-source permettant de monter son propre *cloud* en mode [PaaS](#) nommé “Cloud Foundry”. Pour fonctionner, ce logiciel doit utiliser un socle [IaaS](#) existant. VMWare peut ainsi proposer une offre de *cloud* [PaaS](#) clé en main en empilant “vSphere” et “Cloud Foundry”.

3.2.2.2 Solutions libres

Il existe aujourd’hui pas moins de quatre projets open-source proposant des solutions de *cloud* en mode [IaaS](#) :

OpenNebula est un projet qui a débuté en 2005 et dont la première version publique est sortie en 2008. Il a été choisi en 2010 par le projet 4CaaS de l’Union Européenne visant à construire une plateforme [PaaS](#) sur le sol européen. Aujourd’hui la société C12G assure un support avancé sur ce produit et semble le seul contributeur (pas forcément un gage de longévité donc...).

Eucalyptus est un projet issu des travaux de recherche de l’université de Californie (2008) devenu projet open-source en 2009 lors de la création de la société Eucalyptus Systems qui porte l’offre de service autour de ce projet. Parmi les organismes qui l’utilisent nous trouvons Nokia Siemens Networks, l’[European Space Agency \(ESA\)](#) et la [National Aeronautics and Space Administration \(NASA\)](#).

CloudStack est un produit à l’origine développé par la société *cloud.com* (première version du produit open-source sortie en 2010) puis racheté par Citrix en 2011. Depuis 2012, ce projet a rejoint la fondation Apache.

OpenStack est un projet initié conjointement par Rackspace (numéro 2 du *cloud* aux États Unis derrière Amazon) et la [NASA](#) en 2010. Depuis, plus de 150 grandes entreprises ont rejoint la fondation mise en place en 2012 pour coordonner le projet. Parmi ces entreprises figurent Intel, AMD, RedHat, Bull, Cisco, Dell, HP, IBM, NEC et même VMWare! Fort de ces soutiens, la fondation OpenStack concentre des efforts colossaux de [Recherche & Développement \(R&D\)](#) des plus grands de l’informatique au monde avec un seul objectif : faire d’OpenStack « le Linux du *cloud* ». Un des buts avoués de ce projet est clairement de rattraper le retard pris sur l’offre d’Amazon en fédérant toutes les forces pouvant apporter une contribution. Une jeune société française qui propose des solutions de *cloud*, eNovance, est d’ailleurs parmi les 10 plus gros contributeurs mondiaux à ce projet.

3.2.3 L’idée de *cloud* « souverain »

Pour faire face aux inquiétudes naissantes dans le modèle d’informatique en nuage et tenter de contrer l’hégémonie des fournisseurs américains dans ce secteur, le gouvernement français a estimé en 2009 qu’il était nécessaire de disposer d’un *cloud* « souverain » permettant de garantir que le matériel sur lequel se basera ce *cloud* sera exclusivement sur le sol français et que l’actionnariat national y sera majoritaire. Ceci afin de garantir que le droit français sera le seul à s’appliquer et que la puissance publique française sera la seule à avoir autorité sur son contenu. Ainsi, les administrations et les entreprises françaises manipulant des contenus sensibles pourront s’appuyer sur ce *cloud* en toute confiance sans avoir à craindre, par exemple, une prise de contrôle des autorités américaines sur leurs données.

De cette idée est né le projet Andromède en 2009. Le 18 janvier 2010, François Fillon déclarait : « *Mon souhait est que ce nouveau type d'infrastructure de service fasse l'objet d'un partenariat public-privé grâce aux fonds du programme pour les investissements d'avenir. Il faut absolument que nous soyons capables de développer une alternative française et européenne dans ce domaine, qui connaît un développement exponentiel [...] Force est de constater que les Nord-Américains dominent ce marché, qui constitue pourtant un enjeu absolument majeur pour la compétitivité de nos économies, pour le développement durable et même, j'ose le dire, pour la souveraineté de nos pays* ».

Deux consortiums cofinancés par le secteur privé et la Caisse des Dépôts et Consignations (CDC) vont alors émerger de ce projet :

3.2.3.1 Cloudwatt

Cloudwatt regroupe Orange (pour le réseau) et Thalès (pour la sécurité) dans cette entreprise. La particularité de cette future offre réside dans le fait qu'elle part d'une page blanche. En effet, le choix qui a été fait est celui de créer de toutes pièces de nouveaux *datacenter*, de choisir de nouvelles technologies logicielles, etc. et non de s'appuyer sur une offre existante et de l'aménager. Pour le socle logiciel, le choix de Cloudwatt s'est porté sur OpenStack. Un important partenariat a d'ailleurs été signé avec la société eNovance afin d'apporter à Cloudwatt toute l'expertise nécessaire sur cette brique logicielle. Coté commercial, Cloudwatt opte pour une vente directe des services au client. La mise en production de la totalité de l'offre est prévue pour septembre 2013.

3.2.3.2 Numergy

Numergy regroupe SFR (pour le réseau et l'infrastructure) et Bull (pour la sécurité) dans cette aventure. Cette fois, le choix a été fait de s'appuyer sur un *cloud* existant faisant partie de l'offre "SFR Business Team" afin d'ouvrir plus rapidement le service. Au niveau logiciel, cette infrastructure s'appuie sur VMWare mais un plan de migration vers OpenStack est déjà prévu afin de réduire les coûts. Numergy voit son développement plutôt à l'échelle européenne et espère créer un réseau d'opérateurs de *cloud* à cette échelle lui permettant de proposer une offre garantissant une qualité de service égale dans tous les pays. D'un point de vue commercial, l'offre ne sera pas proposée en direct par Numergy mais sera distribuée via un réseau de partenaires intégrateurs (sélection de SSII) qui accompagneront les clients souhaitant utiliser ce *cloud*.

Certains hébergeurs français (comme OVH et Ikoula) dont les offres de *cloud* étaient déjà opérationnelles avant ces initiatives voient d'un très mauvais œil la « concurrence déloyale » que leur font ces nouveaux venus aidés par l'argent public. C'est ainsi qu'un mouvement demandant le « dégroupage » de ces *clouds*, à l'instar de ce qui existe pour le réseau téléphonique, est né.

Mais ce modèle de cloud « souverain » connaît déjà ses premiers couacs. Fin 2012, à la suite d'un appel d'offre de la région Bretagne visant à souscrire à un service de cloud, la société qui a été retenue était un revendeur de l'offre Amazon AWS. En vertu de l'autonomie de gestion des collectivités locales et des organismes publics en général, il ne sera donc pas si simple de convaincre le public auquel ces cloud se destinent. (source : [24]).

Conclusion

Le logiciel libre a encore un fort potentiel de croissance pour les années à venir. Pressées par le contexte économique, il ne fait aucun doute que les migrations vers des solutions logicielles libres (dans l'administration et ailleurs) vont s'accélérer. La France, pays avancé dans ce

domaine, possède ainsi un atout essentiel au sein de ce secteur économique d'avenir. Il faudra sans doute trouver des solutions pour améliorer le financement de certains projets libres mais la tendance a peu de chance d'être remise en cause dans les années qui viennent. L'OpenData, en mettant les données publiques en libre accès, ouvrira la voie à de nouveaux usages pour le citoyen connecté et permettra à l'économie numérique de s'emparer de nouveaux marchés.

L'informatique en nuage ouvre également de belles perspectives pour l'avenir. Cette fois, la France a plutôt du retard mais est engagée dans la bonne voie. Une fois réellement en place, l'offre de *cloud* « souverain » devrait trouver son public et la participation de la France à des projets open-source majeur pour le *cloud* tel qu'OpenStack nous montre que nous sommes bien dans la course. Il est important que ces initiatives soient encouragées pour nous éviter de devenir de simples spectateurs de cette révolution numérique comme cela a pu être le cas par le passé.

Conclusion

Au terme de ce mémoire, nous pouvons dire que le logiciel libre est un mouvement qui a gagné en maturité depuis les années 1980 pour devenir aujourd'hui un mode de développement collaboratif pragmatique permettant d'unir des forces d'horizons divers derrière un but commun : produire un logiciel de qualité. Ce mode de fonctionnement atteint aujourd'hui, sur certains projets, un niveau de coopération internationale exemplaire (grâce à la création de fondations réunissant les acteurs d'une communauté). Tout cela est rendu possible par la généralisation des réseaux à haut débit qui permettent, à moindre coût, de diffuser et de partager son travail instantanément avec quiconque sur la planète.

Le secteur public français, dans cet univers, s'est positionné très tôt comme un acteur volontaire. En adoptant les logiciels libres massivement dès les années 2000 pour ses plateformes techniques, de nombreuses administrations ont su prendre un virage déterminant pour l'avenir. Aujourd'hui plusieurs analyses, indépendamment de leur coloration politique, constatent que le logiciel libre est une opportunité pour la France d'occuper une place importante dans le secteur du logiciel, sur lequel elle a souvent été distancée par les États-Unis. C'est ainsi que plusieurs rapports préconisent l'emploi de logiciels libres dans le secteur public et recommandent la mise en place de mesures incitatives en faveur de son développement.

Les contraintes financières, dans le contexte de crise que nous traversons, prennent également une dimension nouvelle avec la nécessité de réduire la dépense publique de manière drastique pour les années à venir. Les logiciels libres, comme nous l'avons vu, peuvent aider les administrations à réaliser d'importantes économies sur la mise en place, mais aussi sur les coûts récurrents relatifs à leurs systèmes d'information. Ils permettent également une meilleure mutualisation des logiciels déjà financés par l'argent public entre les administrations ayant des besoins similaires.

Dans un futur proche, les logiciels libres joueront un rôle déterminant dans la généralisation de l'informatique en nuage. La démarche volontariste dans laquelle se sont engagés les pouvoirs publics pour le déploiement d'un *cloud* « souverain » montre que la France a pris la mesure de l'importance de cette mutation. Là encore, l'open-source va permettre à notre pays de jouer pleinement son rôle dans cette transformation et d'en bénéficier sans en être un simple client captif avec le risque de perdre la maîtrise de ses propres données.

Au final, bien que le logiciel libre ne couvre pas tous les domaines fonctionnels, il propose des solutions attractives par rapport au logiciel propriétaire. Le secteur public, déjà bien engagé dans une démarche d'adoption du logiciel libre, doit persévérer et proposer des solutions pour améliorer le modèle économique qui reste aujourd'hui son principal point faible.

Glossaire des termes techniques

binaire

Qualifie, par abus de langage, un programme sous sa forme compilée, exécutable par une machine. Il s'oppose au programme sous sa forme de [code source](#). Le passage du [code source](#) au binaire se fait par une opération de transformation appelée [compilation](#). 27

code source

Le code source d'un programme est un ensemble de fichiers texte contenant des instructions écrites dans un langage de programmation par un ou des être(s) humain(s). Dans le cas d'un langage compilé, ce code source devra être transformé en langage machine au moyen d'un programme appelé compilateur pour pouvoir être exécuté par une machine. Dans le cas d'un langage interprété, ce code source pourra être exécuté directement au moyen d'un programme appelé interpréteur. 2, 3, 5–8, 12–15, 21, 23–25, 27, 28, 33, 39

compilation

La compilation est une opération de transformation du [code source](#) d'un programme (rédigé dans un langage de programmation par des humains) en fichiers [binaires](#) (exécutables par une machine). L'opération de compilation se fait au moyen d'un programme particulier appelé compilateur. 11, 18, 27

datacenter

Il s'agit d'un local technique sécurisé où sont physiquement installés un grand nombre de serveurs permettant d'offrir des services informatiques. 38, 41

développement spécifique

Un développement spécifique est un logiciel développé spécifiquement pour répondre à un besoin précis. Cette notion s'oppose à celle de [progiciel](#). 2, 28

firmware

Il s'agit d'un logiciel propre à un matériel intégré directement dans celui-ci (généralement sur une puce du circuit). Ce logiciel joue le rôle d'interface entre le matériel, figé à sa fabrication en usine, et le logiciel utilisateur, modifiable aussi souvent que nécessaire. Dans un équipement n'ayant pas de système d'exploitation (certains équipements réseaux, appareils photos numériques, copieurs multi-fonctions), le *firmware* joue à proprement parler le rôle de système d'exploitation. 27

forge

Dans le domaine du logiciel, une forge désigne un système de gestion de développement collaboratif de logiciel. Elle se présente généralement sous la forme d'un site web permettant aux développeurs de collaborer sur des projets mais aussi à des non-informaticiens d'apporter leur contribution (traduction, graphisme, forum d'entraide...). 19

fork

Parfois traduit en français par *fourche*, il s'agit d'une opération de séparation d'un projet libre en plusieurs projets. Les *fork* sont très courants dans le monde du libre car n'importe qui peut, en reprenant le [code source](#) d'un projet, créer son propre *fork* d'un logiciel. 9, 10, 24, 28, 30

greffon

Traduction de l'anglais *plug-in*, il porte aussi parfois le nom de module d'extension. Il s'agit d'un programme complémentaire qui peut-être ajouté à un autre logiciel pour en étendre les fonctionnalités. Les greffons répondent à des besoins spécifiques d'extension d'un logiciel ayant un caractère plus général. Un greffon est développé indépendamment du logiciel auquel il se greffe et peut-être conçu sans avoir accès au [code source](#) de celui-ci.

11

ingénierie inversée

Traduction de l'anglais « *reverse engineering* », dans le contexte du logiciel, il s'agit d'une technique qui consiste à reconstituer la manière dont un logiciel a été conçu (parce qu'on ne dispose pas du [code source](#)) à partir de son comportement. Cette activité peut être illégale dans la mesure où les licences de logiciels propriétaires peuvent contenir des clauses interdisant ce type d'ingénierie (quand cela ne va pas à l'encontre des lois du pays). 26

inventaire de parc informatique

Un logiciel d'inventaire de parc informatique (parfois appelé téléinventaire) permet de réaliser un inventaire de la configuration matérielle et logicielle des machines connectées au réseau d'une organisation. Ces informations sont ensuite stockées dans une base de données afin de pouvoir être exploitées à des fins de visualisation ou de statistiques. Ce type de logiciel facilite grandement la vie des gestionnaires de parc, en particulier lorsque ce parc est hétérogène (machines de différentes générations, logiciels variés...). 9

porte dérobée

Une porte dérobée (de l'anglais *backdoor*) dans un logiciel est une fonctionnalité inconnue de l'utilisateur qui donne l'accès à un tiers à ce logiciel. La plupart du temps, elles permettent de collecter des données sur l'utilisation du logiciel à des fins statistiques ou commerciales. Plus rarement, elles peuvent permettre de prendre le contrôle du logiciel à distance ou d'espionner ce qui est fait avec celui-ci. 27

progiciel

Le terme progiciel est un mot-valise, contraction de produit et de logiciel. Il désigne un logiciel qui répond à un besoin professionnel courant mais qui nécessite néanmoins d'être paramétré avant d'être utilisable. La notion de progiciel s'oppose à celle d'un logiciel [développé spécifiquement](#) pour répondre à un besoin. 2, 18, 25, 28

souche logicielle

Il s'agit d'un logiciel précis dans une version majeure donnée. Dans le contexte du logiciel libre, on parle de souches logicielles différentes lors d'un changement majeur de numéro de version ou lorsque s'opère un *fork* donnant naissance à plusieurs projets (ex : FireFox 10.0.1 et FireFox 10.0.16 sont issus de la même souche alors que FireFox 11 est considéré comme une souche différente ; OpenOffice et LibreOffice sont des souches différentes d'un même produit). 24

virtualisation

La virtualisation permet de faire fonctionner plusieurs environnements « invités » sur un environnement « hôte » installé sur une machine physique. L'intérêt de cette technique est de permettre des économies d'échelle en utilisant un même matériel pour installer plusieurs environnements, chacun étant généralement dédié à une tâche. L'autre avantage est d'éviter une adhérence entre un matériel et les environnements « invités ». En effet, un environnement « invité » pourra être migré d'un système d'exploitation « hôte » à un autre par simple copie de fichier entre les deux machines. 28, 38, 40

Acronymes

- ADULLACT** Association des Développeurs et des Utilisateurs de Logiciels Libres pour les Administrations et les Collectivités Territoriales. 19, 35
- AFINEF** Association Française des Industriels de l'Éducation et de la Formation. 34
- AFT** Agence France Trésor. 1, 3, 25, 28, 39
- ANSSI** Agence Nationale de Sécurité des Systèmes d'Information. 28
- APIE** Agence du Patrimoine Immatériel de l'État. 14, 15
- AWS** Amazon Web Services. 38, 41
- BIND** Berkeley Internet Name Daemon. 11
- BT** British Telecom. 28
- CDC** Caisse des Dépôts et Consignations. 41
- CEA** Commissariat à l'Énergie Atomique et aux énergies alternatives. 7
- CeCILL** Cea Cnrs Inria Logiciel Libre. 7
- CERTA** Centre d'Expertise gouvernemental de Réponse et de Traitement des Attaques informatiques. 27
- CMS** Content Management System. 11, 18, 20
- CNAM** Caisse Nationale d'Assurance Maladie. 9
- CNRS** Centre National de la Recherche Scientifique. 6, 7, 10
- DAB** Distributeurs Automatiques de Billets. 9
- DGA** Direction Générale de l'Armement. 28
- DGCP** Direction Générale de la Comptabilité Publique. 15, 29, 30
- DGFIP** Direction Générale des Finances Publiques. 1, 15, 19, 20, 29, 30, 36
- DGI** Direction Générale des Impôts. 22, 30
- DiRIF** Direction des Routes d'Île-de-France. 36
- DISIC** Direction Interministérielle des Systèmes d'Information et de Communication. 18
- DNS** Domain Name Server. 11
- DSI** Directeur des Systèmes d'Information. 3, 18
- ECP** École Centrale de Paris. 8, 10
- ENA** École Nationale de l'Administration. 18
- ENT** Espace Numérique de Travail. 20, 21
- ERP** Enterprise Resource Planning. 18
- ESA** European Space Agency. 40
- ESR** Extended Support Release. 24

- Eucalyptus** Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems. 40
- FSF** Free Software Foundation. 6, 8
- GCC** GNU Compiler Collection. 11
- GED** Gestion Électronique de Documents. 25
- GIMP** GNU Image Manipulation Program. 10
- GNU** GNU is Not Unix. 6–8, 10, 11
- GPL** General Public Licence. 6, 7, 24
- GRC** Gestion de la Relation Client. 25
- HTTP** HyperText Transfer Protocol. 9, 11, 26
- IaaS** Infrastructure as a Service. 37–40
- IBM** International Business Machines. 5, 26, 28, 29
- IDF** Région Ile-De-France. 20
- IIS** Internet Information Server. 23
- INRIA** Institut National de Recherche en Informatique et en Automatique. 7
- INSEE** Institut National de la Statistique et des Études Économiques. 9, 36
- IP** Internet Protocol. 11, 26
- LAMP** Linux Apache MySQL PHP. 25
- LDAP** Lightweight Directory Access Protocol. 11, 30
- LGPL** Lesser GPL. 7, 22
- LOLF** Loi Organique relative aux Lois de Finances. 15, 29
- MIT** Massachusetts Institute of Technology. 6
- MOA** Maîtrise d'ouvrage. 1
- MOE** Maîtrise d'œuvre. 1
- NASA** National Aeronautics and Space Administration. 40
- NG** Nouvelle Génération. 30
- NSA** National Security Agency. 27
- OASIS** Organization for the Advancement of Structured Information Standards. 26
- OBM** Open Business Management. 19, 34
- OCS** Open Computers and Software. 9
- ODF** Open Document Format for office applications. 26
- OSI** Open Source Initiative. 8
- PaaS** Platform as a Service. 37–40
- PC** Personal Computer. 20
- PERL** Practical Extraction and Report Language. 11
- PGP** Pretty Good Privacy. 11
- PHP** Parser Hypertext Preprocessor. 11
- PLUME** Promouvoir les Logiciels Utiles Maîtrisés et Économiques. 6, 10

PME Petites et Moyennes Entreprises. 14, 33, 39

PMI Petites et Moyennes Industries. 39

R&D Recherche & Développement. 40

RATP Régie Autonome des Transports Parisiens. 36

RFC Request For Comments. 11

SaaS Software as a Service. 37, 39

SAE Service des Achats de l'État. 14, 18, 19

SAP Systems, Applications, and Products for data processing. 29

SFR Société Française de Radiotéléphonie. 41

SG-SEP Secrétariat Général - Service de l'Environnement Professionnel. 22

SGBD Système de Gestion de Base de Données. 11, 19, 25, 38

SGMAP Secrétariat Général pour la Modernisation de l'Action Publique. 36

SHA Secure Hash Algorithm. 27

SI Système d'Information. 2, 26, 36, 39

SIG Système d'Information Géographique. 9

SLL Support Logiciel Libre. 9, 14, 22

SNCF Société Nationale des Chemins de Fer français. 36

SQL Structured Query Language. 11

SSII Sociétés de Services en Ingénierie Informatique. 34, 41

SSL Sociétés de Services en Logiciel Libre. 33

SSO Single Sign-On. 30

STaaS Storage as a Service. 37, 39, 40

SUN microsystems Stanford University Network. 6, 24

TPE Très Petites Entreprises. 14, 33

VLC Video-LAN Client. 8, 10

VPN Virtual Private Network. 28

Bibliographie

- [1] THIERRY AIMÉ, *Comment, pour une administration, assurer la mutualisation d'une application informatique ?*, (2007), p. 8. [voir le document](#).
- [2] THIERRY AIMÉ, *Guide pratique d'usage des logiciels libres dans les administrations*, (2008), p. 19. [voir le document](#).
- [3] ANSSI, *Système d'exploitation sécurisé clip*. [voir l'article](#).
- [4] JACQUES ATTALI, *Rapport de la Commission pour la libération de la croissance française*, (2008). [voir le document](#).
- [5] JEAN-MARC AYRAULT, *Orientation pour l'usage des logiciels libres dans l'administration*, (2012), p. 19. [voir le document](#).
- [6] JEAN-BAPTISTE BALLEYGUIER, *Utilisation et développement de logiciels libres au sein de l'administration publique*, (2006), p. 102. [voir le document](#).
- [7] PIERRE BAUDRACCO, *OBM : Découvrez la solution de messagerie collaborative libre!*, (2010), p. 41. [voir le document](#).
- [8] PATRICE BERTRAND, *10 propositions pour une politique du logiciel libre*, (2012), p. 12. [voir le document](#).
- [9] PATRICE BERTRAND, *Panorama de l'open source en France*, (2013), p. 33. [voir le document](#).
- [10] JÉRÔME CANARD, *L'armée capitule face à Microsoft*, Le Canard Enchaîné, (2013), p. 1.
- [11] THIERRY CARCENAC, *Pour une administration électronique citoyenne - Méthodes et moyens*, (2001). [voir le document](#).
- [12] ANSSI CERTA, *Bulletin d'actualité du CERTA*, CERTA-2013-ACT-019, (2013), p. 3. [voir le document](#).
- [13] JOURNAL DU NET, « *Nous avons économisé des centaines de milliers d'euros grâce à OpenERP* », 2011. [voir l'article](#).
- [14] JOURNAL DU NET, « *La gratuité de l'accès aux données publiques restera la règle* », 2012. [voir l'article](#).
- [15] JOURNAL DU NET, *Part de marché des serveurs web en 2012*, 2012. [voir l'article](#).
- [16] JOURNAL DU NET, *Amazon : plus de 2 milliards de dollars tirés du cloud en 2013 ?*, 2013. [voir l'article](#).
- [17] JOURNAL DU NET, *Tomcat : 54 % des serveurs d'applications java*, 2013. [voir l'article](#).
- [18] FRANÇOIS ELIE, *Les modèles économiques du logiciel libre*, (2012), p. 81. [voir le document](#).
- [19] ETALAB, *Licence Ouverte / Open Licence*, (2011), p. 3. [voir le document](#).
- [20] FRANCEINFO.FR, *La France veut son « cloud souverain »*, 2011. [voir l'article](#).
- [21] DIDIER GEORGIEFF, *L'ENA libère son système d'information*, (2010), p. 11. [voir le document](#).
- [22] THE GUARDIAN, *Cloud computing is a trap, warns GNU founder Richard Stallman*, 2008. [voir l'article](#).

- [23] XAVIER GUIMARD, *La politique des logiciels dans la Gendarmerie Nationale*, (2011), p. 25. [voir le document](#).
- [24] ITESPRESSO.FR, *La Bretagne choisit Amazon Web Services, coup de griffe au cloud souverain ?*, 2012. [ouvrir le document](#).
- [25] JEAN-MARIE LAPEYRE, *Le logiciel libre au sein du SI fiscal*, (2005), p. 28. [voir le document](#).
- [26] LAPRESSE.CA, *Le conseil du trésor veut forcer l'utilisation du logiciel libre*, 2013. [voir l'article](#).
- [27] CAILLEUX LAURENT, *TrustedBird, un client de messagerie de confiance*, (2009), p. 11. [voir le document](#).
- [28] LEMAGIT, *Le gouvernement britannique invite officiellement à privilégier les logiciels open source*, 2013. [voir l'article](#).
- [29] MAURICE LÉVY AND JEAN-PIERRE JOUYET, *L'économie de l'immatériel - La croissance de demain*, (2006). [voir le document](#).
- [30] L'EXPRESS.FR, « *L'implantation en France des chinois Huawei et ZTE pose une question de sécurité nationale* », 2012. [voir l'article](#).
- [31] LINUXFR.ORG, *La gendarmerie inventorie son parc et reverse ses contributions !*, 2005. [voir l'article](#).
- [32] GUY MAMOU-MANI, *Numérique éducatif et pédagogique : les professionnels du numérique interpellent le gouvernement sur le respect de la neutralité technologique*, (2013), p. 3. [voir le document](#).
- [33] MARCHÉ-PUBLIC.FR, *Conseil d'État, 30 septembre 2011, n° 350431, région Picardie*, 2011. [voir l'article](#).
- [34] MICHEL-MARIE MAUDET, *La migration vers les logiciels libres et open source : utopie ou réalité ?*, (2008), p. 49. [voir le document](#).
- [35] MICHEL-MARIE MAUDET, *Innovations et futur de l'open source, retour d'expérience et prospective*, (2012), p. 28. [voir le document](#).
- [36] NUMÉRAMA, *La France championne du monde de l'open-source ?*, 2010. [voir l'article](#).
- [37] NUMÉRAMA, *Le logiciel libre représente 6 % du marché en France*, 2012. [voir l'article](#).
- [38] PCINPACT, *Pôle Emploi : le logiciel libre, un choix massif et pragmatique*, 2010. [voir l'article](#).
- [39] PHILIPPE.SCOFFONI.NET, *Trustedbird, vos emails en toute sécurité et intégrité grâce aux logiciels libres*, 2010. [voir l'article](#).
- [40] MATHIEU POUJOL AND CHARLES DE SAINT-PIERRE, *Le logiciel libre en France étude qualitative*, (2008), p. 48. [voir le document](#).
- [41] TOOLINUX, « *La liberté ne s'impose pas, elle se choisit* », 2004. [voir l'article](#).
- [42] FRANCE 5 LE VINVINTEUR, *L'armée « accro » à Microsoft ?*, 2013. [voir l'article](#).
- [43] PRICE WATERHOUSE COOPERS (PWC), *Classement mondial des éditeurs de logiciels : le secteur connaîtra une profonde mutation à long terme*, 2011. [voir l'article](#).
- [44] ZDNET, *La DGI passe à l'open source pour la télédéclaration de revenus*, 2004. [voir l'article](#).
- [45] ZDNET, *La France est devenue « un pays phare pour le logiciel libre »*, 2008. [voir l'article](#).
- [46] ZDNET, *La gendarmerie passera ses 70 000 PC sous Ubuntu d'ici à 5 ans*, 2008. [voir l'article](#).
- [47] ZDNET, *Fleur Pellerin chez Mozilla : le logiciel libre, « garant de la souveraineté numérique »*, 2013. [voir l'article](#).