

L'OPEN SOURCE, UNE ALTERNATIVE AUX GRANDS FOURNISSEURS

Franchir le pas de l'open source



L'open source, une alternative aux grands fournisseurs

Franchir le pas de l'open source

Décembre 2018

Le Cigref est un réseau de grandes entreprises et d'administrations publiques qui a pour mission de développer la capacité de ses membres à intégrer et maîtriser le numérique. Par la qualité de sa réflexion et la représentativité de ses membres, il est un élément fédérateur et acteur important de la société numérique.

Association loi 1901 créée en 1970, le Cigref n'exerce aucune activité lucrative. Il regroupe à ce jour près de 150 grandes entreprises et administrations publiques françaises dans tous les secteurs d'activité. Sa gouvernance est assurée par 15 administrateurs, élus en Assemblée générale. Son activité est animée par une équipe de 10 permanents.

 Droit de propriété intellectuelle

Toutes les publications du Cigref sont mises gratuitement à la disposition du plus grand nombre mais restent protégées par les lois en vigueur sur la propriété intellectuelle. Est autorisée la copie du titre et d'extraits de 500 caractères, suivis chacun de la mention « Source : » assortie de l'url de la publication Cigref. Toute autre reprise doit faire l'objet d'une autorisation préalable auprès du Cigref : cigref@cigref.fr

SYNTHÈSE

Avec la transformation numérique, le périmètre d'intervention des systèmes d'information (SI) des entreprises et organisations publiques ne cesse de s'élargir. Toutes partagent le même objectif de maîtrise de leurs budgets informatiques. Pour dégager des marges de manœuvre dans ces budgets, les entreprises se tournent vers leurs fournisseurs. Dans un contexte où l'innovation métier repose très largement sur les outils et services numériques, la relation des entreprises avec leurs fournisseurs apparaît cruciale au regard des enjeux de compétitivité, d'agilité et de performance.

Aujourd'hui, le marché mondial des licences logicielles et des services numériques se concentre dans les mains de quelques grands acteurs internationaux, essentiellement américains. Ces derniers ont des objectifs de croissance de leur chiffre d'affaires insoutenables pour les entreprises et organisations publiques. De plus, celles-ci acceptent de moins en moins de supporter les coûts de solutions devenues des commodités, ne disposant que d'une valeur d'innovation très limitée, et qui les rendent dépendantes à un fournisseur au comportement hégémonique.

Les entreprises et organisations publiques cherchent donc des alternatives pour desserrer l'étau financier des fournisseurs et pour disposer de marges de manœuvre dans leurs négociations avec les éditeurs de logiciels. Le logiciel libre, dit « *open source* », en est une. En outre, il confère aux entreprises des avantages stratégiques ainsi que des leviers d'action sur leur compétitivité.

Le choix de l'*open source* engendre bien évidemment des coûts. En effet, avant d'ajouter ces solutions au champ des alternatives possibles, les entreprises doivent acculturer les collaborateurs, leur présenter les spécificités de cet écosystème et mettre en place une gouvernance dans le choix des logiciels. Cela nécessite également, en interne ou par externalisation, une montée en compétence pour appréhender les jeux d'acteurs, ainsi que des ressources techniques pointues si l'entreprise souhaite développer certaines spécificités pour répondre à un besoin précis.

L'*open source* questionne les modes de la gestion des risques : les grands éditeurs sont des marques cautions, une assurance, un gage de sécurité pour le Comex. Cependant, des solutions existent dans le logiciel libre pour chaque risque identifié.

De plus, les solutions *open source* répondent aux enjeux actuels : le *time to market*, l'agilité à l'échelle de l'entreprise et l'*open innovation*. Elles facilitent la collaboration et la mutualisation des développements logiciels. Les logiciels libres apportent également la pérennité, la sécurité, la souplesse avec des formats ouverts, une plus grande couverture de tests dans de multiples cas d'usage au sein des larges communautés et parfois une influence possible sur la *roadmap* (plan de développement). L'*open source* offre enfin un levier d'accroissement des compétences et d'attractivité des talents. L'*open source* est même parfois la condition pour attirer les jeunes générations.

L'open source, une alternative aux grands fournisseurs

Franchir le pas de l'open source

Ce sont autant de raisons pour lesquelles certaines entreprises, fortes du soutien de leur Comex, appliquent une stratégie qui étudie toutes les solutions, et en particulier celles de l'open source. Certaines ont même adopté la stratégie « *open source first* ».

Principales recommandations pour franchir le pas de l'open source :

- **Acculturer** les collaborateurs à l'écosystème *open source* (caractéristiques clés de cet écosystème, inconvénients et avantages des logiciels *open source*, gestion des risques)
- **Définir des critères d'analyse des logiciels *open source*, communs** à toute l'entreprise et qui prennent en compte la prise de risque à long terme ; par exemple, la propriété intellectuelle, le type de licence, les communautés d'utilisateurs et de développeurs, etc.
- **Établir une gouvernance** au sein de l'entreprise dans le choix des logiciels ;
- **Mettre en place un processus de choix des logiciels**. Autrement dit, choisir dans toutes les solutions logicielles possibles le meilleur compromis entre :
 - le **besoin métier**. Les entreprises préconisent de définir le besoin fonctionnel auquel doit répondre le logiciel ;
 - le **prix** ;
 - le **support possible**.

Ce choix prend en compte le degré de dépendance à des solutions logicielles propriétaires des grands fournisseurs et la possibilité de mutualiser des développements.
- De la même façon, mettre en place un **processus de choix du support du ou des logiciels**

REMERCIEMENTS

Nos remerciements vont à Stéphane Rousseau, CIO de Eiffage, qui a piloté cette réflexion ainsi qu'à toutes les personnes qui ont participé et contribué à ce groupe de travail Cigref :

Oussama ABOU - ALLIANZ
Aimery ASSIRE - EDF
Latifa AYEY - EDF
Noëlle BAILLON BACHOC - ORANGE
Nicolas BAILLY - SAINT-GOBAIN
Alain BARBAY - MINISTERE DE L'INTERIEUR
Cyril BARTOLO - LAGARDÈRE
Rony BASSET - VALLOUREC
Jean-Charles BASTOUL - MINISTERE DE L'INTERIEUR
Vincent BELROSE - LVMH
Jérôme BOIN - MAIF
Claire BONNETAIN - EIFFAGE
Didier BOUFFARD - AXA GROUP
Joël BOURHIS - GRDF
Loïc BOURNON - SAFRAN
Marc BOUSSET - SAFRAN
Priscilla BOUVET - TOTAL
Valérie BREUT - DASSAULT AVIATION
Yvan BRUNEL - COVEA
Marie-Flore CABANE - MINISTERE DES ARMEES
Noël CAVALIÈRE - GROUPE PSA
Philippe CHASSAING - VEOLIA
Simon CLAVIER - SNCF
David COURTOIS - IDEMIA
Yves DANIEL - SAFRAN
Sophie DARRACQ - BANQUE DE FRANCE
David DARWISH - IDEMIA
Olivier DE BERNARDI - SOCIÉTÉ GÉNÉRALE
Philippe DEBRAIZE - MINISTERE DES ARMEES
Laurent DELISLE - THALES
Philippe DOUBLET - GROUPE POMONA
Olivier DU MERLE - AIR FRANCE KLM
Cédric DUFOUR - AGIRC ARRCO
Ewen DUGUE - GEODIS INTERSERVICES
Mélanie DUVEAU - AGIRC ARRCO
David ELOY - EUROPCAR
Fatima FAR - ESSILOR INTERNATIONAL
Philippe GALICHET - AP-HP
Julie GARANÇON - GROUPAMA
Patrick GERVAIS - EDF
Thierry GRISELAIN - CREDIT AGRICOLE SA
Christine GROSSETÊTE - AIR LIQUIDE
Marie-Pierre GUGLIELMI - PMU
Marie-Laure GUIDOT - GRDF
Coraline HAYRAUD - ARKEMA
Alain ISSARNI - CNAM
Yannick JOBARD - ORANGE
Diane JOLY - AIRBUS
Pascal LAFOND - COVEA
Michel LAMRANI - DASSAULT AVIATION
Raynald LASOTA - FRANCE TELEVISIONS
Pierre LE BER - EDENRED
Cédric LE BERRE - GEODIS INTERSERVICES
Sylvie LE GALL - NAVAL GROUP
Hervé LE MEN - NEXITY
Patrick LEFEUVRE - EDF
Olivier MARCHAIS - GIP MDS
Catherine MAURICE - PLASTIC OMNIUM
Malika MEZAIER-POUPIN - CARREFOUR
Erwan MOYSAN - AXA GROUP
Ravi NADARADJANE - SCOR
François-Gaudéric OLIVE - EDF
Pierre PAN - ESSILOR INTERNATIONAL
Christian PATERSON - ORANGE
Chantal PEYRAT - DASSAULT AVIATION
Jean-Claude PIGEON - RTE
Béatrice POTIER - SAFRAN
Elizabeth PUGEAT - CREDIT AGRICOLE SA
Laurent QUÉRÉ - AVIVA
Anaïs ROMAND - AXA GROUP
Philippe ROUAUD - FRANCE TELEVISIONS
Florence PERROT - CNAM
Christophe SARRE - EDF
Nathalie SENARD - VEOLIA
Frédéric SERET - SFR SI
Grégory SILVAIN - EURO DISNEY
Jean-François STRICHER - ENEDIS
Teodora TOMA - TOTAL
Axel TOUBERT - CAISSE DES DÉPÔTS
Hubert TOURNIER - RAMSAY GÉNÉRALE DE SANTÉ
Géraldine TRIBONDEAU - AXA GROUP

Le Cigref remercie également sincèrement les personnalités extérieures suivantes pour leurs interventions et leurs contributions aux réflexions :

Aimery ASSIRE - EDF
Simon CLAVIER - SNCF
Xavier GUIMARD - Gendarmerie
Alain ISSARNI - CNAM
Laurent JOUBERT - DINSIC
Etienne JULIOT - Obeo
Hervé LEMAITRE - Red Hat
Bruno POIRIER COUTANSAIS - Gendarmerie
Alexandre ZAPOLVSKY - Linagora

Ce document a été rédigé par Marine de SURY, chargée de mission Cigref, avec la participation de Stéphane Rousseau.

ÉDITO DU PILOTE

« Parce que les organisations diffèrent, parce qu'il faut être de plus en plus flexible, agile pour répondre aux besoins métiers, parce qu'il faut garantir la plus grande indépendance de manœuvre, le *sourcing* logiciel est devenu un enjeu majeur pour le directeur des systèmes d'information. Cependant, la concentration du marché autour de grands acteurs éditeurs déplace sensiblement le curseur des prises de décision. Il est urgent par conséquent de favoriser un écosystème alternatif dynamique et maîtrisable par les entreprises utilisatrices. Dans les logiciels d'infrastructures, cet écosystème existe : c'est le monde de l'*open source*.

Comparé au modèle traditionnel de l'édition de logiciels, qui montre les limites de sa pertinence, l'*open source* correspond davantage aux nouveaux codes de co-création, de production collaborative et d'*open innovation*.

L'*open source* est devenu un composant incontournable d'une stratégie informatique. »

Stéphane Rousseau, Pilote du groupe de travail
CIO Eiffage et administrateur Cigref

TABLE DES MATIÈRES

| | |
|--|----|
| Introduction | 1 |
| 1. Ecosystème <i>open source</i> : caractéristiques clés | 2 |
| 1.1. Types de licences | 2 |
| 1.2. Modèles économiques de l'éditeur <i>open source</i> | 4 |
| 1.3. Propriété intellectuelle | 5 |
| 1.4. Précision sur le coût du déploiement d'un logiciel <i>open source</i> | 6 |
| 1.4.1. Emploi d'une licence | 7 |
| 1.4.2. Coût de développement des fonctionnalités futures (mutualisé ou pas) | 7 |
| 1.4.3. Adaptations pour l'intégrer au SI..... | 8 |
| 1.4.4. Coût support professionnel des logiciels libres | 8 |
| 1.4.5. Coûts induits à ne pas omettre | 9 |
| 2. Inconvénients et avantages de l' <i>open source</i> | 10 |
| 2.1. Inconvénients | 10 |
| 2.2. Spécificités en matière de compétences | 11 |
| 2.3. Avantages stratégiques..... | 12 |
| 2.3.1. Au niveau système | 12 |
| 2.3.2. Au niveau de la marge de manœuvre | 13 |
| 2.4. Avantages compétitifs..... | 14 |
| 2.4.1. Ouvrir et collaborer..... | 14 |
| 2.4.2. Mutualiser | 15 |
| 3. Gestion des risques des logiciels libres | 17 |
| 3.1. Types de risque des logiciels libres et propriétaires..... | 17 |
| 3.1.1. Capacité à faire | 17 |
| 3.1.2. Risque opérationnel | 18 |
| 3.2. Réponses aux risques | 18 |
| 3.2.1. Pas de support spécifique | 18 |
| 3.2.2. Choix d'une distribution commerciale | 18 |
| 3.2.3. Acquisition de support « éditeur » ou tiers..... | 18 |
| 3.2.4. Choix d'aucune politique de réassurance..... | 19 |
| 4. Engager le passage à l' <i>open source</i> | 20 |
| 4.1. Acculturation des collaborateurs et des dirigeants | 20 |
| 4.2. Gouvernance du choix du logiciel | 20 |
| 4.3. Consultations et appels d'offres | 21 |
| 4.4. Exemple de l'État | 22 |
| 5. Choix d'un logiciel libre : processus et critères d'analyse | 23 |
| 5.1. Processus de choix d'un logiciel <i>open source</i> | 23 |
| 5.2. Critères d'analyse des solutions <i>open source</i> | 24 |
| 5.3. Processus de choix de support de solutions <i>open source</i> hors solution à souscription... | 26 |
| Conclusion | 28 |

TABLE DES FIGURES

| | |
|--|----|
| Figure 1 : Évolution possible de l'attribution d'une licence à un logiciel libre | 3 |
| Figure 2 : Coût total du déploiement d'un logiciel <i>open source</i> | 7 |
| Figure 3 : Modèles économiques des logiciels <i>open source</i> | 25 |
| Figure 4 : Maintenance et support en fonction du logiciel | 26 |

PRÉAMBULE

A l'occasion de ses vœux 2018, le Cigref a défini [sept résolutions](#) pour réussir le passage du numérique en France et en Europe d'ici 2020. Sept résolutions pour faire de 2018 l'an 1 de la reconquête de notre autonomie stratégique dans toutes les dimensions du numérique, au service de la croissance de l'économie, de la compétitivité des entreprises et de la création d'emplois. L'une des résolutions vise à créer un environnement de régulation, en termes de normalisation, de concurrence et d'infrastructure pour que les systèmes numériques puissent être déployés et gouvernés dans les entreprises et les administrations publiques de manière agile et sans entrave excessive.

Enjeux avec les fournisseurs de logiciels

Le Cigref accompagne ses membres dans leurs relations avec les éditeurs au sein de son club « relation fournisseurs » et leur offre un espace de réflexion, d'action et de partage entre pairs, notamment sur les difficultés que ces relations sont susceptibles de générer. Depuis le début des années 2000, le Cigref a choisi de considérer les fournisseurs comme des partenaires et de travailler avec eux pour trouver des solutions aux pierres d'achoppement rencontrées par les grandes entreprises et administrations membres.

Le marché mondial des licences logicielles et des services numériques se concentre entre les mains de quelques grands acteurs internationaux, essentiellement américains. Dans un contexte où l'innovation métier repose en grande partie sur les outils et services numériques, la relation avec les fournisseurs est cruciale et essentielle au regard des enjeux d'innovation, d'agilité et de performance, conditions de leur compétitivité future.

Objectifs budgétaires des entreprises, des organisations publiques et des fournisseurs

Les grandes entreprises et organisations publiques partagent toutes aujourd'hui le même objectif de **maîtrise de leur budget informatique**. Elles optimisent autant que possible ce budget pour répondre à leur feuille de route. Elles se tournent alors vers leurs fournisseurs afin de continuer à dégager des marges de manœuvre.

Cependant, la stratégie des prix des grands fournisseurs engendre des tensions du fait des trop grandes incertitudes qu'elle génère concernant l'évolution des budgets SI des entreprises. Les prix des fournisseurs sont variables selon la négociation ; les métriques évoluent et induisent une augmentation de la facture, amplifiée par la digitalisation des métiers et le recours grandissant aux services numériques. En fin de contrat de support aux logiciels, l'entreprise a le « choix » entre migrer ou souscrire une extension. Les entreprises se sentent par conséquent en situation de dépendance de plus en plus avérée. La captivité des entreprises s'accroît encore avec le développement des solutions en mode SaaS (*Software as a Service*). De plus, les accès indirects démultiplient les coûts d'achat de licences.

Enfin, de nombreux logiciels sont maintenant des commodités avec une valeur d'innovation limitée, et les clients acceptent de moins en moins de payer un prix élevé pour des produits jugés communs et rentabilisés, et d'être liés à un fournisseur.

Introduction

Des solutions alternatives aux grands éditeurs de logiciels propriétaires existent déjà et sont opérationnelles au sein d'entreprises. En effet, des acteurs qui ne sont pas d'aussi grande taille ni dans une position dominante - typiquement les acteurs de taille moyenne - peuvent offrir la flexibilité nécessaire pour assurer l'équilibre budgétaire de leurs clients. Les entreprises savent « sourcer » ces fournisseurs alternatifs en fonction de leurs besoins. Cela inclut également les *startups* susceptibles, avec le temps, d'offrir elles aussi des alternatives.

Les solutions *open source* constituent également un moyen de réduire le budget des systèmes d'information (SI) des entreprises et organisations publiques. Les membres du Cigref s'intéressent à l'*open source* depuis plus de 10 ans. Son dernier rapport sur le sujet, « [Maturité et gouvernance de l'Open Source](#) », publié en mars 2011, étudiait la gouvernance de l'*open source*, la migration des entreprises vers *open office* et la perception de la maturité de l'*open source* dans les grandes entreprises.

Un rapide état des lieux effectué fin 2017 des entreprises participant au groupe de travail Cigref révèle l'utilisation courante de solutions *open source*, principalement dans l'infrastructure, moins dans les applications métiers. Les solutions libres sont plus ou moins utilisées en fonction du SI patrimonial de l'entreprise, de sa stratégie ou de ses relations avec ses fournisseurs historiques.

Cependant, les logiciels libres offrent bien d'autres possibilités que celle de réduire le budget des SI. Les entreprises sont **convaincues des avantages stratégiques qu'ils leur confèrent ainsi que des leviers d'action qu'ils leur donnent pour agir sur leur compétitivité**. Le choix de solutions *open source* leur permet **de développer les compétences et d'attirer les talents**. Enfin, **la maîtrise du SI** qu'elles visent est accessible par l'*open source*.

Cependant, choisir une solution *open source* nécessite de bien appréhender les spécificités de cet écosystème afin d'éviter de se fourvoyer. Aider les entreprises à franchir le pas de l'*open source* est le but de ce rapport qui s'adresse aux collaborateurs de la direction des SI et plus généralement à l'ensemble des collaborateurs. En effet, ils sont pour la plupart concernés puisque le numérique s'étend désormais à tous les secteurs d'activité.

La première partie de ce rapport présente les caractéristiques clés de l'écosystème *open source* à avoir en tête au moment du choix de solutions. Dans la deuxième partie, les inconvénients ainsi que les avantages compétitifs et stratégiques sont détaillés. Le développement de l'*open source* dans les entreprises questionne la gestion des risques. En effet, la gestion des risques des logiciels libres diffère de celle des logiciels des grands éditeurs. C'est l'objet du troisième chapitre. Nous verrons ensuite que les entreprises se doivent d'anticiper le passage aux logiciels *open source* et de s'organiser en conséquence. Enfin, le dernier chapitre propose des éléments - processus et critères d'analyse - communs à toute l'entreprise pour aider au choix d'un logiciel.

1. Ecosystème *open source* : caractéristiques clés

Cette partie a pour objectif de brièvement rappeler les éléments clés à connaître sur l'*open source* afin de faire un choix éclairé. Ces notions sont essentielles. En effet, les entreprises doivent préalablement bien appréhender l'écosystème *open source* pour ne pas faire fausse route.

Le logiciel est dit *open source* lorsque le code source est ouvert et libre d'accès. Cela correspond en anglais au *Free Open Source Software*, *FOSS*.

Le logiciel libre, ("*free software*"), *Free Libre Open Source Software*, *FLOSS*, apporte une **dimension éthique sur la liberté**. 4 types de liberté sont sous-jacents :

- la liberté d'étudier le code source (accès) et le fonctionnement du programme pour l'adapter à ses besoins ;
- la liberté de modifier/améliorer le code source (accès) et le partager ;
- la liberté d'exécuter le programme ;
- la liberté de redistribuer et copier le code.

Ces 4 libertés sont parfois couplées à des règles (contraintes).

À l'exception des logiciels du domaine public, les logiciels libres n'ont pas moins de propriétaires que les autres, encore que le terme **auteur** soit plus approprié. Les auteurs de logiciels libres conservent certains droits qui découlent directement du type de licence libre employée.

1.1. Types de licences

Un contrat suppose des obligations réciproques avec son co-contractant, exploitant ou utilisateur. Une licence est une **autorisation unilatérale**. Dans la licence, le propriétaire des droits d'auteur sur un programme informatique définit les conditions dans lesquelles ce programme peut être utilisé, diffusé ou modifié. **Les logiciels libres sont distribués avec des licences, généralement sans la moindre garantie.**

Pour avoir le droit d'utiliser un logiciel, il faut que le titulaire des droits l'autorise. La licence est le document dans lequel il énumère les droits accordés au licencié (installer le logiciel, l'utiliser, faire une copie de sauvegarde). Souvent, le titulaire des droits ne se contente pas de concéder la licence, il ajoute également des exigences comme l'interdiction d'étudier le codage du logiciel, de publier des mesures de ses performances, etc.

Dans le cas des logiciels *open source*, différentes licences libres récapitulent les libertés et les règles. Elles se répartissent en 3 grands types de licence :

- Les licences **permissives** également nommées *copyfree*. Il n'y a pas de restriction sur la redistribution ;

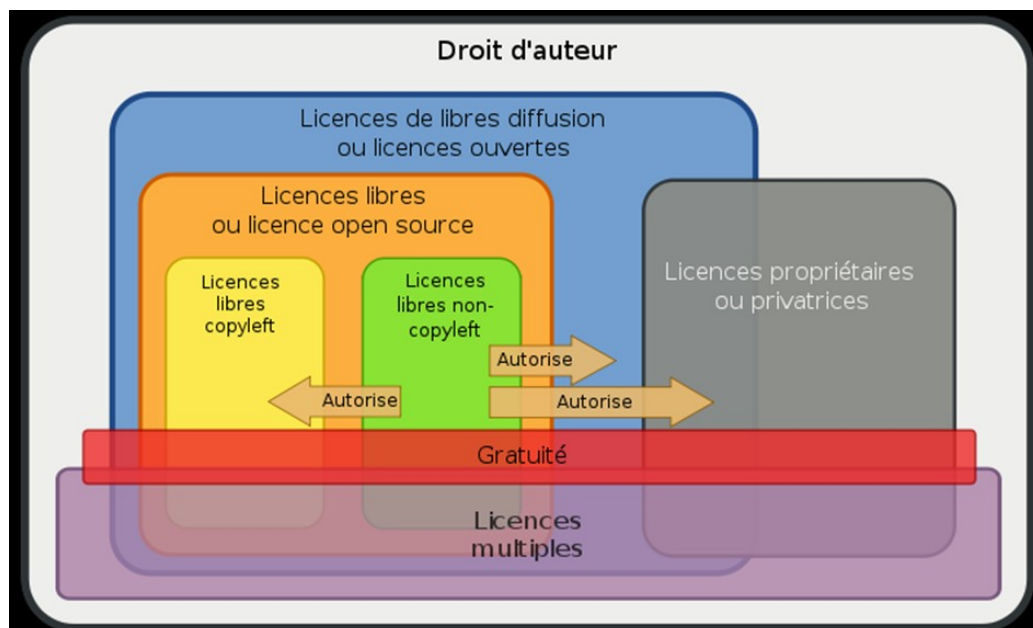
L'open source, une alternative aux grands fournisseurs

Franchir le pas de l'open source

- Les licences **non permissives**, appelées aussi *copyleft* (pour s'opposer au *copyright*¹) : l'auteur garde un *copyright* et la redistribution doit se faire sous la même licence. Dans son principe, les licences *copyleft* empêchent celui qui utilise le code de s'approprier l'effort de la communauté sans rien lui reverser des améliorations ou des corrections. La contribution à l'effort collectif devient un principe et permet de maintenir la dynamique de leurs développements. Il existe une gradation dans les licences non permissives qui va de la licence *copyleft* faible à forte. En effet, les licences non permissives fortes, c'est-à-dire les plus « contraignantes », imposent que tout projet qui réutilise le code source du logiciel libre devienne lui-même un logiciel libre. L'utilisation d'un logiciel *open source* avec ce type de licence n'est donc pas possible pour une entreprise qui souhaite faire des développements spécifiques qu'elle ne veut pas partager...
- Les licences du domaine public. Le *copyright* a expiré ou il n'y a pas de *copyright*.

En conclusion, il est donc important que l'entreprise analyse le type de licence du logiciel libre qu'elle souhaite utiliser, pour identifier les droits et les éventuelles obligations et exigences associées occasionnées.

Le schéma ci-après récapitule comment l'attribution d'une licence à un logiciel libre peut évoluer.



Source : Wikipedia

Figure 1 : Évolution possible de l'attribution d'une licence à un logiciel libre

¹ Le *copyright* (littéralement droit de copier), souvent indiqué par le symbole ©, est, l'ensemble des prérogatives exclusives dont dispose une personne physique ou morale sur une œuvre de l'esprit originale. Il désigne donc un ensemble de lois en application, notamment, dans les pays du Commonwealth et aux États-Unis ; et qui diffère du droit d'auteur appliqué dans les pays de droit civil (tels que la France ou la Belgique).

Le *copyright* relève plus d'une logique économique et accorde un droit moral restreint, là où le droit d'auteur assure un droit moral fort en s'appuyant sur le lien entre l'auteur et son œuvre. (Source Wikipedia)

L'open source, une alternative aux grands fournisseurs

Franchir le pas de l'open source

Ainsi un logiciel libre en *copyleft* restera à vie en *copyleft*. Il ne peut pas évoluer. En revanche, un logiciel libre en non *copyleft* peut avoir une évolution, un *fork*², ou être repris par un autre éditeur. Et ce dernier a le droit de (est autorisé à) modifier la licence en *copyleft* ou en licence propriétaire ou tout simplement en licence ouverte.

Deux types de *fork* existent dans l'*open source* :

- le **fork agressif** : c'est un compétiteur au projet existant qui se base sur le projet pour le faire évoluer dans une nouvelle direction ;
- le **fork amical**, rendu célèbre par l'usage de la technologie Git, permet d'expérimenter l'ajout d'une nouvelle fonctionnalité en vue de la faire intégrer au projet initial si le résultat est pertinent.

La commercialisation des logiciels libres est possible mais la liberté de redistribuer et copier le code en interdit l'exclusivité. C'est pourquoi les logiciels libres ne permettent pas la rétribution directe de leurs auteurs. Ceux-ci doivent se limiter à la vente de services associés à l'utilisation desdits logiciels. **Les modèles d'affaires consistent donc à vendre un savoir-faire ainsi qu'une expertise, plutôt qu'un droit d'usage sur le logiciel.**

Il est très important, une fois le type de licence identifié, de comprendre le modèle économique des éditeurs *open source* qui est rappelé ci-après.

1.2. Modèles économiques de l'éditeur *open source*

L'éditeur *open source* est une entreprise à but lucratif comme une autre. Le développement de ses logiciels représente un coût que l'éditeur finance sur la base d'un retour sur investissement. Il est important de bien comprendre le *business model* de l'éditeur de logiciel *open source* afin d'éviter les mauvaises surprises ultérieures.

Quatre grandes catégories de *business model* sont possibles :

- **Les doubles licences.** Le logiciel développé est libre sous une licence à *copyleft* fort et ne peut donc pas être intégré dans un produit ou service propriétaire. Une deuxième licence propriétaire est alors vendue aux utilisateurs qui souhaitent s'affranchir du *copyleft*.
- **Le modèle mutualisé.** Certains besoins génériques requièrent de lourds efforts de R&D difficiles à consentir car les recettes attendues ne pourront être collectées qu'à long terme. Regrouper et mutualiser les besoins de plusieurs clients et donc de plusieurs financements peut être une solution pertinente pour se lancer dans de tels développements. Dans ce cas, un logiciel de base est développé pour motiver des clients à co-développer des modules associés à ce logiciel basique.
- **Le modèle de distribution packagée ou *open core*.** L'éditeur développe un noyau technologique *open source* et un produit propriétaire par-dessus ce noyau, ou bien développe deux produits au-dessus d'un noyau commun, l'un en *open source* et l'autre avec des fonctionnalités additionnelles. Dans ce dernier cas, les deux variantes du

² Le *fork*, fourche en anglais, est la duplication du code source d'un projet existant pour en faire un nouveau projet (source Obeo).

L'open source, une alternative aux grands fournisseurs

Franchir le pas de l'open source

logiciel sont souvent baptisées « *community* » et « entreprise ». La version gratuite a pour objectif d'élargir la communauté et de la diffuser au maximum.

- **Le modèle de services ou de valorisation indirecte.** L'éditeur tire dans ce cas ses recettes de services professionnels, de la maintenance et du support associés au logiciel qu'il distribue.

Pour approfondir, les [modèles économiques](#)³ des éditeurs sont détaillés dans *les livrets bleus du GTLL* (Groupe Thématique du Logiciel libre de Systematic). Le [livret bleu juridique](#)⁴ rappelle le cadre juridique du logiciel, les spécificités du logiciel libre et détaille les droits et obligations en cas d'usage ou d'exploitation de logiciels sous licence libre.

Lors de l'emploi de logiciel *open source*, il est donc important de comprendre son modèle économique afin de bien appréhender les coûts de la solution, qu'il s'agisse de logiciel *open source* à souscription en modèle fermé, en modèle ouvert, ou libre et gratuit.

Les logiciels *open source* à souscription en modèle fermé consistent en une souscription obligatoire, une *roadmap* imposée par l'éditeur, et une communauté dominée par un seul acteur. Java SE d'Oracle illustre le changement de *business model* d'une licence *software* : début 2018, Oracle a modifié en profondeur le modèle de licence, de commercialisation et de support de Java SE (à compter de la version 9). En créant un *licensing dual*, Oracle sort désormais tous les 6 mois une nouvelle version « *freemium* » *open source* de Java pour les développeurs (cycle court) et tous les 3 ans des licences commerciales de plus long terme (*Long Terme Support - LTS*), avec support intégré pour les clients « entreprises ».

Dans le cas des logiciels *open source* à souscription en modèle ouvert, le logiciel et l'ensemble des modules sont *open source* et ouverts à tous. Cependant le **contrat impose des limitations aux libertés du logiciel libre**. L'éditeur vend le service d'intégration des composants sur des projets complexes et certifie l'industrialisation et la stabilité (cycle de vie, sécurité). Il assure également la maintenance et le support.

Dans le modèle économique de l'*open source*, hors produits *open source* à souscription, l'utilisateur paie le service associé au juste besoin et non la rente du logiciel. Il choisit le fournisseur du support avec le service associé.

1.3. Propriété intellectuelle

Avant tout achat, il est primordial d'identifier là où se trouve la propriété intellectuelle du logiciel *open source* afin d'éviter toute surprise ultérieure. Les droits patrimoniaux peuvent être hébergés au sein d'une fondation, d'une communauté ou enfin chez un éditeur. Avant de continuer, rappelons brièvement l'écosystème *open source*.

L'*open source* fonctionne au sein d'un écosystème qui comprend les clients, l'éditeur, l'ESN (Entreprise de Services Numériques) et la fondation.

³ Document disponible à l'url : https://systematic-paris-region.org/wp-content/uploads/2017/07/LivretBleu_ModelesEconomiques_GT-LogicielLibre_Systematic.pdf

⁴ Document disponible à l'url : https://systematic-paris-region.org/wp-content/uploads/2017/07/LivretBleu_Juridique-2eEdition_GT-LogicielLibre_Systematic_Nov2016_web.pdf

L'open source, une alternative aux grands fournisseurs

Franchir le pas de l'open source

- Les **clients** financent les évolutions et le support ;
- L'**éditeur** crée le logiciel. Les développeurs et utilisateurs de logiciels libres sont généralement indépendants, avec des motivations diverses. Ils réalisent ou utilisent des outils fondés sur un mode de production reposant sur la collaboration, l'entraide, le partage, la mutualisation ;
- L'**ESN** intègre le logiciel dans le système d'information. Les ESN peuvent être amenées à contribuer à des développements *open source* mais elles ne sont pas éditrices pour autant ;
- La **fondation** héberge le code avec des règles de neutralité et de gouvernance. Les fondations ne sont en effet pas là pour créer le logiciel qu'elles hébergent. Elles s'occupent de l'animation de la communauté et de la promotion des logiciels.

Si les droits patrimoniaux sont hébergés au sein d'une fondation, alors l'éditeur ne pourra pas passer en mode propriétaire ni modifier son modèle d'affaires.

La marque Sirius appartient à la fondation Eclipse. Par conséquent Obeo, un des principaux éditeurs contributeurs aux développements, ne peut pas faire passer cette solution en mode propriétaire sous la marque Sirius.

Obeo, Etienne Juliot

Si la propriété intellectuelle n'est pas logée au sein d'une fondation, une investigation s'impose afin d'identifier la structure dans laquelle se concentre la recherche et développement. En effet, si la **R&D** et la **propriété industrielle** résident dans une même structure alors il y a un risque d'enfermement. Autrement dit, l'entité constituée de la R&D, forte de la propriété industrielle, peut à tout moment décider de modifier son modèle économique.

Enfin, identifier qui **finance les développements** d'un logiciel *open source* constitue le deuxième point à considérer. En effet, les financeurs contrôlent réellement la marque et influencent les choix stratégiques.

Obeo est le principal contributeur de IS Designer mais c'est le Ministère des Armées qui en finance la très grande majorité. Si une autre entreprise souhaite utiliser ce projet et proposer des évolutions, elle devra négocier avec le ministère soit directement, soit *via* le sous-traitant.

Obeo, Etienne Juliot

Le projet peut devenir communautaire si plusieurs entreprises *via* un ou des sous-traitants y contribuent. D'un cadre centralisé, on passe alors à un projet communautaire.

1.4. Précision sur le coût du déploiement d'un logiciel *open source*

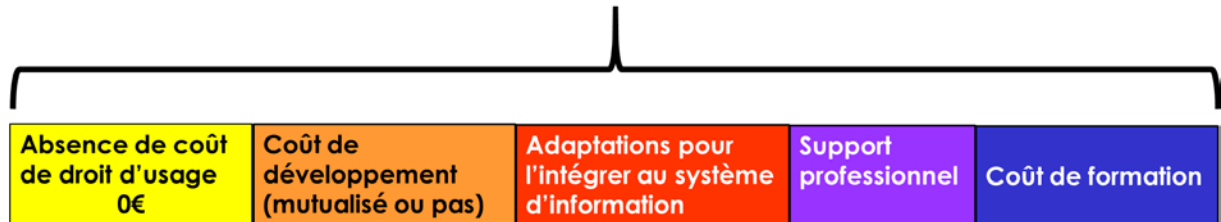
Le coût total du déploiement d'un logiciel *open source* inclut les éléments ci-après :

L'open source, une alternative aux grands fournisseurs

Franchir le pas de l'open source

Coût total = absence de coût de droit d'usage (0 €) + coût de développement (mutualisé ou pas) + adaptations pour l'intégrer au système d'information + support professionnel + coût de formation.

Coût total du déploiement d'un logiciel open source



Source : Cigref

Figure 2 : Coût total du déploiement d'un logiciel *open source*

1.4.1. Emploi d'une licence

Le coût d'achat d'un logiciel *open source* est nul. Son développement a déjà été payé par d'autres.

1.4.2. Coût de développement des fonctionnalités futures (mutualisé ou pas)

Concernant le développement de fonctionnalités futures intéressant l'entreprise, deux cas sont possibles. Premier cas, l'entreprise est la seule intéressée par ces développements : elle les finance intégralement. Deuxième cas, plusieurs entreprises sont intéressées : elles choisissent alors de mutualiser le coût des développements du tronc commun.

Lorsqu'une entreprise intègre des évolutions à un logiciel libre pour ses propres besoins, elle crée une nouvelle branche de développement. À chaque mise à jour de version du logiciel commun, elle devra alors réintégrer ses propres codes dans le code source mutualisé, à moins qu'elle ne parvienne à convaincre la communauté d'intégrer ses apports à la branche de développement communautaire. S'ils sont intégrés, alors les coûts de maintenance et d'intégration seront mutualisés : l'entreprise diminuera ainsi ses coûts opérationnels à chaque nouvelle version installée.

C'est pourquoi une entreprise (co-)finançant un développement cherchera à faire en sorte qu'il soit accepté par la communauté. Cela lui évite le coût d'un *fork*. L'entreprise doit pour ce faire vérifier que ses contributions sont acceptables et acceptées par la communauté. Elle s'assurera dans un premier temps que la communauté accepte ces reversements. Pour cela, elle analysera le type de licence retenue pour le code commun et l'écosystème des contributeurs. En effet, si la communauté est portée essentiellement par un éditeur offrant par ailleurs des solutions « entreprise », il est très probable qu'elle refusera toute autre contribution. L'entreprise s'assurera ensuite que ses contributions sont conformes aux attentes de la communauté : les développements ne touchent pas au code natif, original ; cohérence avec la *roadmap* ; qualité du code de la documentation, etc.

1.4.3. Adaptations pour l'intégrer au SI

Cela inclut l'intégration et les paramétrages spécifiques. Les coûts d'intégration des logiciels libres sont parfois plus onéreux que ceux des solutions des éditeurs, de par leur complexité. En effet, les produits *open source* sont plus compliqués à intégrer lorsqu'ils portent chacun sur des fonctionnalités limitées. Pour avoir une couverture globale, il faut passer par des distributions qui, lorsqu'elles sont communautaires, n'intègrent pas toujours les facilités de déploiement et de montée des versions successives.

Lorsque des développements pour l'intégration au SI sont acceptés par la communauté, il n'est pas nécessaire de redéfinir tout le paramétrage et la personnalisation à chaque mise à jour. Cela participe à la maîtrise des coûts opérationnels.

La CNAM se fait accompagner dans l'intégration des solutions *open source* dans le SI, dans leurs évolutions ou dans le développement de nouvelles fonctionnalités. Le contrat précise bien que tout développement doit être reversé à la communauté.

CNAM, Alain Issarni

1.4.4. Coût support professionnel des logiciels libres

L'entreprise a la possibilité de choisir le fournisseur qui assurera le support et la maintenance des logiciels *open source* à modèle communautaire. Cela n'est pas le cas des logiciels *open source* à souscription qui imposent le support et la maintenance ni des logiciels propriétaires.

Les garanties des logiciels libres sont la plupart du temps identiques et parfois même plus importantes que celles des grands éditeurs. Pour mémoire, un grand éditeur ne prend aucun engagement de résultat quant à la résolution de bug. En fonction du contrat, le périmètre pour les logiciels libres, inclut le support 7j/7j, la maintenance corrective, la garantie du bon fonctionnement, la conformité de la solution. L'entreprise choisit le niveau de support en fonction de la criticité du service ou de l'application. Le choix du support est détaillé dans le paragraphe 5.3 - Processus de choix du support de solutions *open source* hors solution à souscription.

Lorsqu'un logiciel est développé en *open source* par une entreprise et est accepté par la communauté, l'entreprise n'a alors pas à payer la garantie. En effet, dans ce cas, la garantie est assurée par la communauté.

La CNAM a fait le choix stratégique « *Open First* ». Pour accompagner cette orientation elle s'est dotée d'un marché forfaitaire de support de l'ensemble de ses logiciels libres utilisés (dans la limite de 100 logiciels et 3 versions par logiciels choisis librement par la CNAM et dont la liste peut évoluer librement en fonction de ses besoins) auprès d'un prestataire. Elle bénéficie donc d'un contrat de support à la « mode éditeur classique » avec un interlocuteur unique pour l'ensemble de ses logiciels libres utilisés. Contrat de support dont les conditions d'intervention et de corrections sont définies par la CNAM en fonction de la criticité des applications. Ce marché global et forfaitaire offre également

l'agilité qui permet de faire supporter les logiciels alors qu'ils sont encore en phase d'évaluation.

CNAM, Alain Issarni

Des entreprises préconisent de préciser dans le contrat avec le sous-traitant et prestataire que tout développement *open source* doit être reversé à la communauté. C'est aussi une façon de contribuer.

1.4.5. Coûts induits à ne pas omettre

Le passage à des solutions *open source* induit parfois des coûts masqués qu'il faut garder à l'esprit. Comme c'est le cas pour beaucoup de solutions logicielles, un investissement est nécessaire dans les compétences en interne ou en externe. Certaines compétences sont rares et donc chères.

L'entreprise doit prévoir un coût supplémentaire lorsqu'elle est amenée à changer ses pratiques, son organisation pour adopter le travail collaboratif, mutualiser des développements ou encore se lancer dans le *make*.

Un autre coût peut survenir lorsque l'entreprise souhaite influencer, ou même définir, la *roadmap* (animation / contribution) en s'investissant dans les communautés.

Le gain global d'une stratégie *open source* est difficilement quantifiable en général car chaque cas est différent. En outre, l'arrêt d'un marché génère des économies. Mais choisir une solution logicielle libre nécessite de nouveaux investissements (compétences, communauté) et marchés (support via une ESN).

2. Inconvénients et avantages de l'open source

Les systèmes d'information et de communication sont des **outils au service des métiers**. Ils constituent un **levier de la performance** pour les équipes. Le système d'information doit offrir la possibilité de s'ouvrir aux clients, partenaires, fournisseurs et plus généralement à l'ensemble de l'écosystème de l'entreprise. Les logiciels *open source* répondent à cet enjeu d'ouverture et offrent un certain nombre d'avantages stratégiques ainsi que des leviers d'action sur l'amélioration de la compétitivité de l'entreprise. Cependant, l'*open source* présente également quelques inconvénients à ne pas négliger. Enfin les compétences induites par l'*open source* sont à la fois un plus et un moins au sein de l'organisation. C'est pourquoi cette thématique est traitée dans un même sous-chapitre.

2.1. Inconvénients

L'offre de services liés aux logiciels *open source* ne couvre pas l'intégralité des services d'un éditeur, à moins d'être un modèle à souscription ou propriétaire basé sur l'*open source* ... L'entreprise doit dès lors s'adapter en conséquence.

Un autre inconvénient se résume à la **jungle juridique** qui l'encadre : il faut être à l'aise avec les licences de l'écosystème, les différents modèles économiques possibles et bien comprendre où se trouve la propriété intellectuelle pour connaître les risques associés au logiciel libre. Il convient enfin de bien connaître les obligations associées à un logiciel libre.

La connaissance des *roadmaps* de logiciels *open source* - hors modèles à souscription - est également moins facile à obtenir.

Comme nous l'avons vu ci-dessus, les entreprises participant au groupe de travail Cigref utilisent des solutions *open source* principalement dans l'infrastructure, moins dans les applications métiers. En effet dans les couches basses, des solutions *open source* existent et sont implémentées : elles font en particulier partie des solutions déployées dans le cadre des offres *Cloud*. Mais plus on monte, moins d'alternatives en *open source* existent. En effet, les offres *open source* sur les SI métiers sont moins matures et moins consolidées, même si des suites logicielles métiers ont commencé à apparaître dès 2015. Leurs retours sur investissement sont moins élevés que sur les couches techniques. De plus, la gouvernance des logiciels *open source* sur les SI métiers est décentralisée ou partagée au sein de plusieurs directions ou filiales. Il semble également assez complexe d'intégrer des briques fonctionnelles *open source* au sein des systèmes d'information *Corporate*, finances, gestion parfois composés de briques hétérogènes (Oracle appli, SAP, MSBI, progiciels de niches, développements spécifiques, solutions SaaS, etc.). Les solutions *open source* imposent un travail d'intégration informatique complémentaire par rapport aux solutions propriétaires. Si celui-ci relève du ressort des opérateurs IT, c'est de moins en moins vrai lorsqu'on se rapproche des sphères métiers. L'effort d'intégration important nécessite de nouvelles compétences difficiles à trouver sur le marché, ce qui accroît les coûts.

Franchir le pas de l'open source

Enfin, les offres métier semblent dimensionnées pour des filiales ou pour des PME mais ne sont pas forcément « *scalables* », c'est-à-dire dimensionnées à l'échelle du périmètre des multinationales.

2.2. Spécificités en matière de compétences

Le recrutement et la fidélisation des profils capables de travailler dans le domaine de l'*open source* sont des enjeux majeurs pour les entreprises.

L'open source, un levier attractif de recrutement : travailler sur des logiciels *open source* avec l'écosystème associé constitue un levier attractif de recrutement : ouverture aux communautés, acceptation du partage, ...

La gestion des talents dans l'open source : la mise en visibilité des talents *open source* est importante au risque de les voir attirés par d'autres sociétés de l'écosystème *open source*.

La formation et la reconnaissance dans le milieu de l'open source : l'écosystème *open source* facilite la montée en compétence avec la collaboration, l'entraide et l'émulation qui leur sont inhérentes. En effet, dans cet écosystème, les contributeurs s'aident mutuellement et aspirent à devenir *committers*. Ce passage représente une reconnaissance de la qualité de leurs contributions et de leur implication dans la communauté. Pour rappel, le rôle des contributeurs consiste à développer, traquer les bugs, vérifier les performances mais également à documenter leurs développements. Les *committers*, qui sont en nombre limité, ont accès en écriture au code *open source* et valident la qualité du code des contributeurs. Un développeur devient *committer* parce qu'il est choisi par ses pairs *committers*.

L'intégration dans les grandes entreprises des profils open source : l'intégration de ces profils peut générer des tensions culturelles. Elles ont en général des cultures bien différentes de celles pratiquées dans les grandes entreprises. Cela représente un véritable enjeu de changement de culture et de manière de produire : capacité à changer les modes de *delivery*, l'organisation, etc.

La multiplication des compétences requises : on assiste à une multiplication des compétences pour gérer, et les logiciels propriétaires et les logiciels libres.

Choisir des solutions *open source* demande d'avoir une vision globale du « marché », de son évolution, de la compréhension des jeux d'acteurs (éditeurs, communautés, etc.). De plus, des ressources techniques pointues peuvent être requises pour développer des spécificités ou basculer dans le *make* pour répondre aux besoins de l'entreprise.

Choisir le type et le périmètre du support en fonction des besoins, avoir plusieurs interlocuteurs au lieu d'un seul chez les grands éditeurs nécessitent également des compétences. En effet, les entreprises ont été habituées avec les éditeurs propriétaires ou de modèle à souscription à traiter avec un responsable unique qui s'occupait de tout : l'évolution, le support et la maintenance. La multiplicité des types de supports possibles dans le libre, le périmètre de la prestation de support et garantie à définir, présentent une certaine complexité qu'il convient de savoir gérer.

2.3. Avantages stratégiques

Il est important de noter que l'*open source* force l'éditeur au *fair play* et à l'excellence de son modèle. Puisque le code source est public, le client ou une partie de la communauté peut à tout moment choisir de faire un fork en cas de désaccord.

2.3.1. Au niveau système

Par rapport à celui d'un « grand fournisseur », le logiciel *open source* offre l'opportunité aux entreprises de reprendre la main sur la maîtrise du système d'information concernant :

- La conformité ;
- La sécurité ;
- L'obsolescence ;
- L'interopérabilité ;
- La réversibilité ;
- La rationalisation du SI ;
- Le choix du fournisseur de support.

Conformité - sécurité :

Les logiciels *open source* ont un avantage structurel décisif en matière de conformité et sécurité informatique sur les logiciels propriétaires dont le code est fermé : le code est ouvert et donc analysable. L'entreprise a la possibilité de vérifier par elle-même la conformité avec les réglementations. Cela induit également une sécurité par transparence. En effet, si la disponibilité du code source ne garantit pas la sécurité, la capacité de modifier celui-ci garantit au moins la possibilité d'obtenir un correctif en cas d'exposition à une vulnérabilité. Cependant les entreprises veulent éviter autant que possible d'avoir à corriger les vulnérabilités : elles comptent sur les communautés.

Obsolescence - pérennité - interopérabilité - réversibilité :

Lors du choix d'un outil, la pérennité peut être clé. Dans l'aviation par exemple, la garantie porte sur une durée de 50 ans. La confiance liée à la pérennité d'un outil est plus facilement donnée à un fournisseur bien connu qu'à du logiciel *open source*. Et pourtant, le logiciel *open source* n'a pas de limite temporelle à son usage. Les corrections sont possibles même à très long terme. Puisque le logiciel est *open source*, même si l'éditeur est racheté ou fait faillite, le code source reste disponible. Enfin, si l'éditeur choisit un cap qui ne convient pas, le *fork* est possible. L'entreprise choisit la version du logiciel *open source* souhaitée. Elle n'est pas tributaire de l'obsolescence programmée « voulue » par les éditeurs. Autrement dit, elle n'a plus besoin de passer en support étendu ; elle ne court plus le risque de faire reposer son SI sans support sur des versions trop anciennes du logiciel dès lors qu'elle investit dans les compétences nécessaires à l'exercice de cette indépendance. De par sa nature, le logiciel *open source* s'interopère facilement et rend possible la réversibilité.

L'open source, une alternative aux grands fournisseurs

Franchir le pas de l'open source

Rationalisation du SI :

De plus en plus, les entreprises cherchent à rationaliser leurs systèmes d'informations. Pour cela, elles découpent les projets en modules ou briques avec des interfaces standardisées et indépendantes. Elles cherchent à supprimer peu à peu les interdépendances propriétaires entre projets et entre outils propriétaires. Elles développent un socle commun et limitent les développements spécifiques. En plus de l'imposition de standards ouverts à la frontière de chaque projet, certaines d'entre elles visent l'indépendance des plateformes logicielles vis-à-vis du système d'exploitation et limitent les technologies utilisées. Cela participe à la maîtrise des coûts opérationnels.

Le système d'information de l'Etat est vaste et complexe, peu centralisé, et avec des anciennetés d'applications pouvant aller jusqu'à plusieurs décennies. Avec l'arrivée permanente de nouveaux besoins métiers qui s'ajoutent au SI existant, l'effort de rationalisation est nécessaire et permanent pour maîtriser les coûts. Plutôt que d'opter pour une approche classique d'urbanisation en s'intéressant successivement à différents quartiers fonctionnels pour ne plus proposer qu'une seule application, avec souvent une justification de retour sur investissement plus que d'adéquation au besoin, la stratégie de mutualisation mise en place ne cherche pas systématiquement une solution unique. Ainsi la stratégie « Etat plateforme » vise une certaine « permaculture du logiciel », afin de proposer un socle de base, dénominateur commun répondant à la majorité des usages, mais permettant à chacun de l'enrichir ou d'y déroger dans un principe de subsidiarité. Cette approche offre une plus grande résilience et permet de mieux intégrer les innovations qui peuvent venir de petits projets pilotes pour être progressivement intégrées au socle. Cette approche nécessite une maîtrise de l'architecture pour définir la structure du système d'information et permettre différents assemblages à la manière d'un Lego. L'*open source* et les interfaces par API sont des prérequis pour permettre cette rationalisation adaptative et maîtrisée du système d'information.

DINSIC, Laurent Joubert

Choix du fournisseur du support :

Contrairement au logiciel propriétaire, le logiciel libre donne la liberté à l'entreprise de choisir le prestataire qui assurera les services de maintenance et de support. De plus, en cas de force majeure, l'entreprise a la possibilité de prendre la main sur le logiciel *open source* pour corriger elle-même un bug important qui peine à être résolu. Cela doit rester bien évidemment exceptionnel. Ceci est bien sûr sous réserve de disposer des ressources compétentes.

2.3.2. Au niveau de la marge de manœuvre

Dans le modèle économique de l'*open source*, hors produits *open source* à souscription, l'utilisateur paie le service associé au juste besoin et non la rente du logiciel. C'est une des raisons pour lesquelles l'alternative *open source* est étudiée avec attention.

Aujourd'hui, pour répondre à un cas d'usage déterminé, les entreprises incluent dans leur recherche l'ensemble des solutions possibles disponibles auprès des éditeurs, auprès des communautés *open source*, et auprès des prestataires qui effectueront les développements

L'open source, une alternative aux grands fournisseurs

Franchir le pas de l'open source

sur mesure. Fortes des solutions qui répondent à leur cas d'usage, elles choisissent de façon pragmatique la solution avec le meilleur retour sur investissement. Étudier des solutions alternatives assure aux entreprises une certaine marge de manœuvre dans les négociations : les grands éditeurs savent que l'entreprise s'est ouvert la possibilité de basculer vers des solutions *open source*. D'ailleurs, elles ont parfois déjà basculé vers ces solutions au niveau des infrastructures.

Un grand nombre d'entreprises intègrent au moins un logiciel *open source* dans leurs appels d'offres. Cette pratique fait partie de la gouvernance et permet d'une part de rester à l'écoute de ce que peut proposer le monde *open source* et d'autre part de contenir les prix des concurrents à l'appel d'offres. Ces derniers savent qu'un logiciel *open source* donc « gratuit » est en lice contre eux, ce qui les incite à rester raisonnables sur le prix de leurs licences notamment.

Lagardère Ressources, Cyril Bartolo

Enfin, le choix de l'*open source* participe à la Responsabilité Sociétale des Entreprises (RSE). En effet, l'intégration dans le système d'information s'effectue en local avec des contrats de support en France. Les grands fournisseurs, eux, facturent en général les entreprises à partir de leur entité étrangère.

2.4. Avantages compétitifs

2.4.1. Ouvrir et collaborer

La transformation numérique donne la capacité à une entreprise de trouver de nouveaux relais de croissance sur des marchés inédits. Les solutions *open source* permettent de s'affranchir de la lourdeur des règles des éditeurs classiques, d'accélérer le processus et de tester rapidement ces relais de croissance dans un contexte économique qui évolue très vite. En effet, le logiciel *open source* évite la phase de prêt de licences ou de négociations liées au prêt. Les logiciels *open source* permettent d'initier des POC sans avoir à discuter avec l'éditeur. Le test se fait avec ses propres ressources. Si besoin est, l'entreprise fait appel à des prestations de service comme elle en a l'habitude.

L'*open source* représente aussi un véritable accélérateur de programmes d'innovation car les logiciels libres offrent des leviers technologiques, particulièrement dans les domaines de l'agilité IT, du DevOps (ce sont principalement des solutions *open source* qui sont déployées), du cloud, des APIs ouvertes, de l'IoT, etc.

L'utilisation de solutions *open source* facilite la collaboration et l'agilité.

De nombreux services numériques sont basés sur l'OSS Open Support System comme les services SMACS : Social, Mobilité, Analytics, Cloud et Sécurité. Pour prendre des parts de marché dans les domaines de l'innovation et plus précisément de l'Intelligence Artificielle, de la réalité augmentée, du *big data*, de l'*analytics*, les GAFAM mettent en *open source* les algorithmes. Ainsi tout le monde les utilise et ils deviennent les solutions employées. Les R&D peuvent travailler en réseau sur les algorithmes en toute transparence.

L'open source, une alternative aux grands fournisseurs

Franchir le pas de l'open source

Enfin l'écosystème *open source*, véritable laboratoire d'expériences technologiques, permet aux entreprises d'identifier les tendances technologiques émergentes en vogue ou en croissance et d'analyser leur évolution. Cet écosystème représente donc un baromètre qui indique la pérennité des différentes technologies et leur potentiel pour trouver des relais de croissance.

Certaines entreprises ont des besoins précis sans réponse disponible sur le marché. En effet, plus la technologie répond spécifiquement à un besoin précis, moins il est probable qu'elle intéresse d'autres entreprises. Cela pose la question du *make or buy*. Choisir le *make* en *open source* représente un vrai accélérateur de projet dans la mesure où l'entreprise a adopté le « *mindset* », l'état d'esprit *open source*. Les logiciels libres permettent de s'affranchir de certaines contraintes éditeurs et donnent plus d'autonomie. Le rapport au temps est important à prendre en compte.

Plusieurs entreprises choisissent enfin de collaborer lorsqu'elles ont des besoins communs ou tout au moins une base commune. Elles décident alors de développer un cœur *open source* générique maintenu par un éditeur *open source* qu'elles cofinancent. Chacune développe ensuite les extensions permettant de répondre à son cas spécifique. L'extensibilité et la genericité du logiciel *open source* sont alors cruciales. Pour rappel, le principe d'extensibilité repose sur une plateforme *open source* : cette base de *plugin* permet d'offrir des solutions ouvertes d'un point de vue cas d'usage. Passer un logiciel en *open source* permet de toucher une grande communauté y compris à l'international. Cette diffusion élargie permet aux éditeurs *open source* de propulser leur logiciel.

Lorsque Obeo est devenue spécialiste *open source*, les 5 000 visites annuelles sont passées à 5 000 mensuelles et sont aujourd'hui à 5 000 hebdomadaires. Les visiteurs hier, surtout français, viennent aujourd'hui, à 75%, de l'international.

Obeo, Etienne Juliot

2.4.2. Mutualiser

Les grandes entreprises mutualisent leurs ressources pour développer des logiciels *open source* en commun au sein d'alliances, d'associations, ou de secteurs industriels (industries verticales).

Les grandes entreprises ont en effet déjà commencé à mutualiser des développements software afin de se concentrer sur la partie applications et services qui les différencient sur le marché. C'est le cas par exemple de l'alliance GENIVI. Ce consortium à but non lucratif fondé en 2009 par BMW Group, Delphi, GM, Intel, Magneti-Marelli, Groupe PSA, Visteon, et Wind River Systems, a pour objectif de créer un standard de système d'exploitation basé sur Linux pour l'industrie des systèmes multimédia et systèmes de navigation automobile. L'alliance GENIVI a ouvert son projet *open source* sous l'égide de la Linux Foundation. L'alliance compte aujourd'hui plus de 160 entreprises membres qui travaillent ensemble pour fournir une plateforme logicielle basée sur Linux, ouverte et cohérente pour toute l'industrie automobile.

L'open source, une alternative aux grands fournisseurs

Franchir le pas de l'open source

La mutualisation a également lieu au sein des communautés de développeurs mais aussi dans les communautés d'utilisateurs qui fiabilisent et stabilisent le produit.

Les entreprises n'hésitent plus aujourd'hui à se regrouper au sein de partenariats inter-entreprises pour partager leurs retours d'expérience et même mutualiser des développements. C'est le cas d'associations comme le PGGTIE et TOSIT.

PGGTIE, groupe de Travail Inter-Entreprise de l'association PostgreSQL.fr, association loi 1901, a pour but de **promouvoir le système de gestion de bases de données PostgreSQL** dans les pays francophones. L'objectif du groupe de travail est de mutualiser les efforts pour 3 actions concrètes sur PostgreSQL :

- Partager des retours d'expérience de mise en œuvre. Les membres échangent sur leurs architectures respectives, sur la consolidation d'ordres de grandeur sur le parc et sur les volumes. Une boîte à outils est mise à disposition pour partager et consolider les documentations respectives afin d'en publier les résultats ;
- Contribuer à la pérennisation, l'amélioration et l'exploitabilité ;
- Encourager l'utilisation et l'adoption de PostgreSQL par les entreprises. Encourager également le support par les éditeurs de progiciels.

L'association [TOSIT](#), née en juillet 2017 et forte de 8 membres à ses débuts, est une structure de soutien à l'émergence de logiciels *open source* et de solutions informatiques sous licence libre, notamment en matière de cybersécurité. La spécificité de TOSIT est d'offrir une structure juridique permettant **d'engager** les entreprises membres **dans l'action rapidement**. L'association TOSIT a plusieurs objectifs :

- Favoriser les échanges avec des retours d'expériences industrielles et des partages de bonnes pratiques ;
- Collaborer activement en lançant des **études stratégiques ou techniques et des veilles**, en lançant des chantiers de sécurisation des briques les plus critiques des systèmes d'information, en formalisant et en mettant en œuvre **des chantiers via des accords de partenariats** ;
- Appliquer les principes structurants de TOSIT qui consistent à sélectionner les produits dans une logique d'industrialisation pour de grands groupes, à reverser autant que possible à la communauté les développements issus de TOSIT, à éviter les solutions dont les droits patrimoniaux ne sont pas largement partagés, à sortir de la dépendance vis-à-vis des solutions *open source* à souscription, en passant d'une dépense liée au déploiement à une dépense sur une base forfaitaire.

Le Cigref engage les grandes entreprises à contribuer au sein de communautés *open source*, à se regrouper dans des GIE pour mutualiser des développements *open source* autour d'intérêts communs et à les reverser autant que possible aux communautés.

3. Gestion des risques des logiciels libres

Maintenir les risques à un niveau acceptable est primordial pour l'entreprise. La gestion des risques en général répond à plusieurs objectifs :

- Créer et préserver la valeur, les actifs et la réputation de l'entreprise ;
- Sécuriser la prise de décision et les processus de l'entreprise pour favoriser l'atteinte des objectifs ;
- Favoriser la cohérence des actions avec les valeurs de la société ;
- Mobiliser les collaborateurs de l'entreprise autour d'une vision commune des principaux risques.

Comme les logiciels éditeurs, les logiciels *open source* présentent un risque auprès des entreprises. Cependant les fournisseurs propriétaires bénéficient d'une antériorité historique favorable avec des solutions connues et dotées d'une aura de réputation dans l'entreprise. Il est important de préciser le contexte historique afin de bien avoir en tête les enjeux concernant les logiciels. Des éditeurs jouissaient de certains monopoles au XX^e siècle. Les modèles de ces fournisseurs propriétaires proposaient des solutions « culturellement implantées » dans l'entreprise. Le choix de leurs solutions se justifiait donc simplement en interne. Leur puissance marketing est importante et ils n'hésitent pas à faire du lobbying auprès des décideurs. Les grands éditeurs sont des marques cautions, une assurance, un gage de sécurité pour le COMEX et le CEO. De plus, ils proposent un responsable unique qui s'occupe de tout : l'évolution, le support et la maintenance. Ce modèle diminue la prise de risque de l'individu ... au risque de l'augmenter pour l'entreprise. En effet, ce modèle présente un risque pour l'entreprise avec les incertitudes autour de l'évolution des coûts et des métriques, la *roadmap* non maîtrisée, etc.

Il ressort deux grands types de risques liés aux logiciels : la capacité à faire et le risque opérationnel. Les entreprises favorisent une approche pragmatique lors du choix d'une solution : elles cherchent le logiciel *open source* qui réponde le mieux au cas d'usage et s'assurent que le niveau de support souhaité est disponible. Le niveau de support est fonction de la criticité des domaines du cas d'usage. Cependant, lorsque le logiciel *open source* peut se passer de support, cette option est choisie.

3.1. Types de risque des logiciels libres et propriétaires

3.1.1. Capacité à faire

La « capacité à faire » concerne le cas de logiciels qui nécessitent un travail d'intégration important sur des plateformes complexes avec beaucoup de composants. Le développement *software* pour l'intégration n'est pas le cœur de métier de l'entreprise : il n'y a pas de ressources pour faire l'intégration. Autrement dit, les ressources et les budgets sont contraints. Cela représente un risque de maîtrise.

3.1.2. Risque opérationnel

Le second risque est opérationnel. Une fois le logiciel intégré, il faut assurer la continuité de service des systèmes critiques. C'est pourquoi l'entreprise doit disposer d'un **dispositif de support** et d'**intervention** en cas d'incidents sur les systèmes critiques. En effet, une interruption de service sur certains logiciels n'est pas acceptable au niveau métiers, prospects, ou autre.

3.2. Réponses aux risques

Plusieurs types de réponses sont possibles en ce qui concerne les logiciels libres.

3.2.1. Pas de support spécifique

Cette réponse concerne des **applications avec un SLA, Service Level Agreement⁵ standard ou moyennement élevé** (impact modéré en termes économique, juridique, de ressources humaines ou d'image). Il s'agit de produits logiciels simples, fiables avec une qualité éprouvée avec une évolutivité et des améliorations en continu. Ces logiciels sont largement installés et utilisés par une communauté importante et active qui fait évoluer le produit. L'entreprise a une maîtrise en interne suffisante de la version communautaire pour contourner et traiter les incidents. Elle n'a donc pas besoin de support. Les produits cités en exemple dans ce cas sont Tomcat, la base de données MySQL.

3.2.2. Choix d'une distribution commerciale

Les entreprises participantes au groupe de travail font le choix d'une distribution pour les **produits ou plateformes complexes d'infrastructure à SLA élevé** comme Linux Suse, OpenStack par exemple. L'ensemble des composants de cette plateforme **nécessite un travail d'expert important pour l'intégration**. Le dispositif retenu par les entreprises est dans ce cas une distribution commerciale. La société choisie ou le distributeur assure **l'intégration, le support et les services annexes**, comme l'assurance par exemple en mode « traditionnel ». Cette solution peut être une disposition transitoire, le temps pour l'entreprise de faire monter en compétences ses propres ressources afin de pouvoir l'assurer elle-même après.

3.2.3. Acquisition de support « éditeur » ou tiers

Le choix de l'acquisition de support « éditeur » ou tiers s'adresse également aux applications ou plateformes d'infrastructure avec une **exigence de disponibilité élevée**, autrement dit à **SLA élevé**. Les produits concernés sont des plateformes ou des produits

⁵ Le *service-level agreement* (SLA) ou « entente de niveau de service » est un document qui définit la qualité de service, prestation prescrite entre un fournisseur de service et un client. Autrement dit, il s'agit de clauses basées sur un contrat définissant les objectifs précis attendus et le niveau de service que souhaite obtenir un client de la part du prestataire et fixe les responsabilités (Source Wikipedia)

Franchir le pas de l'open source

complexes, ou encore des produits-clés des applications comme les bases de données. Dans ce cas, **seul le support est concerné et non l'intégration**. Deux dispositifs sont retenus, la **souscription de support** auprès de l'éditeur du logiciel libre ou la souscription d'un support auprès d'une société tierce. RedHat Fuse qui exige la disponibilité de la plateforme, est une illustration de cas-là.

Des entreprises du groupe de travail ont fait appel à ce type de support pour assurer PostgreSQL car des applications critiques l'utilisent.

3.2.4. Choix d'aucune politique de réassurance

La Gendarmerie Nationale a fait le choix stratégique de n'avoir aucune politique de réassurance. Elle s'est organisée en conséquence et assure le support avec, à ce jour, un niveau de support au moins équivalent à celui des grands éditeurs. Elle a investi en fonction de ses besoins, en interne pour avoir des compétences qui maîtrisent les logiciels, et/ou dans les communautés *open source*. Plusieurs cas se présentent en fonction de la solidité de la communauté qui développe le logiciel *open source* :

- Tout d'abord, lorsque le logiciel *open source* est largement utilisé et soutenu par une communauté solide (comme open office, open LDAP, etc.), il n'y a pas de gestion particulière de risque car la communauté s'en charge. Le bug détecté est transmis à la communauté qui le corrige généralement aussi rapidement qu'un « grand » éditeur.
- Lorsque des logiciels primordiaux pour l'organisme sont soutenus par des communautés limitées ou jugées pas assez solides, la Gendarmerie Nationale met des ressources pour participer à la *roadmap* et adapter les produits aux besoins. La reprise de la communauté OCS-Inventory de 2005 à 2010 en est une illustration. Si cela s'avère nécessaire, elle demande à des prestataires externes de développer les nouveaux modules dont ils ont besoin. Ces modules sont reversés à la communauté, ce qui évite ainsi le maintien en condition opérationnelle.
- La Gendarmerie Nationale a créé des communautés pour les logiciels critiques lorsque l'offre n'est pas satisfaisante. Par exemple pour le SSO LemonLDAP::NG. Elle s'assure que la communauté puisse ensuite prendre son envol afin de s'en désengager. Le maintien en condition opérationnelle est généralement pris en charge par des prestataires qui utilisent le logiciel dans leur modèle d'affaires.

Gendarmerie Nationale

4. Engager le passage à l'*open source*

Afin d'encourager les salariés de l'entreprise à utiliser des solutions *open source*, il faut préparer la **communication**. En effet, certaines entreprises avaient présenté le passage aux solutions *open source* comme une façon de participer aux restrictions budgétaires, aux économies. Cela n'a pas tenu longtemps, certains managers ayant choisi de revenir au logiciel propriétaire antérieur. Il est **primordial de présenter un message positif, le présenter à un moment opportun, montrer l'avancée intéressante pour les collaborateurs**.

Engager les collaborateurs à choisir pour un cas d'usage donné la solution *open source* - si c'est la plus appropriée - nécessite un accompagnement. En effet, les collaborateurs ne connaissent pas toujours les spécificités de cet écosystème. La maturité des salariés au sein d'une même entreprise est très variable. La méconnaissance engendre un manque de confiance. De plus, omettre l'analyse de certains points peut entraîner des surprises, au niveau du budget par exemple. Ce chapitre propose également des recommandations concernant la gouvernance.

4.1. Acculturation des collaborateurs et des dirigeants

Utiliser des solutions *open source* ne correspond pas juste à un changement de technologie. Les solutions *open source* ont également des modèles d'affaires avec une stratégie de prix propre à leur écosystème. L'**acculturation** des collaborateurs est donc importante et à plus forte raison l'**acculturation des dirigeants**, afin que les choix soient faits en connaissance de cause, en toute sérénité. Ce sont en effet les dirigeants qui sont démarchés par les équipes commerciales des éditeurs propriétaires, ressources dont ne disposent pas les éditeurs *open source* en général. Bien analyser où loge la propriété intellectuelle, identifier le type de licence utilisée, choisir le support en adéquation avec ses besoins, comprendre l'écosystème pour pouvoir contribuer ou influencer la *roadmap*, etc., nécessitent une culture de base pour éviter de se fourvoyer.

Les directions métiers n'hésiteront pas à étudier aussi les solutions *open source*, pour le cas d'usage recherché, s'ils ont **confiance** dans le produit et dans le support associé. Le support semble parfois défaillant pour certaines applications métier ou solutions sur la partie *middleware* gérée par les Ops. D'où l'importance d'expliquer en quoi le support et les garanties sont équivalents à ceux des éditeurs.

4.2. Gouvernance du choix du logiciel

Une clé dans l'utilisation des logiciels libres au sein d'une DSI réside dans la mise en place de la gouvernance relative aux choix des solutions à différents échelons.

L'open source, une alternative aux grands fournisseurs

Franchir le pas de l'open source

Tout d'abord, pour choisir un fournisseur de ce type, il est nécessaire de mettre en place une organisation commune à l'ensemble des collaborateurs avec des outils qui facilitent le choix du logiciel. L'exemple suivant a été partagé :

- **Cartographier les besoins en solution logicielle** par famille et par critère pour les cas d'usage concernés.
- Rechercher systématiquement les alternatives libres.
- **Instaurer une gouvernance** de choix technologiques :
 - Mettre en place les **mécanismes de consultation et de validation** de ces choix par les parties prenantes ;
 - **Contrôler l'application du choix** et les dérogations.

La CNAM se fait accompagner pour alimenter sa réflexion sur les évolutions technologiques, pour étudier la maturité et pérennité des logiciels libres dans un domaine. Fort de cette analyse menée avec une équipe interne, elle installe une gouvernance de choix des logiciels. Ces choix sont ensuite insérés dans son catalogue interne de logiciels libres qui rentrent dans son marché unique de support ce qui permet ainsi de définir une gouvernance d'usage des logiciels utilisés en son sein (seuls les logiciels du catalogue peuvent être utilisés).

CNAM, Alain Issarni

La méthode d'analyse QSOS, *Qualification and Selection of Open Source Software*, met en valeur ce qui différencie les solutions libres étudiées. Cela impose le contact des informaticiens avec les communautés *open source*. Ce contact des informaticiens avec la communauté permet aussi d'avoir un retour sur la *roadmap*.

La DINSIC mène actuellement une expérimentation pour obtenir des métriques sur la non utilisation de solutions libres référencées dans leur catalogue maison. Afin de mieux maîtriser les usages, elle cherche à affiner les métriques d'usage.

DINSIC, Laurent Joubert

Une fois le logiciel *open source* sélectionné, il est préconisé d'effectuer un audit de l'application en production et d'avoir un retour de sa mise en œuvre et de son intégration dans le SI.

Enfin, il est intéressant de mettre en place une gouvernance des logiciels inscrits au support pour décider mensuellement quels sont les logiciels et les versions à déployer.

4.3. Consultations et appels d'offres

Des entreprises précisent dans leurs consultations le reversement systématique des développements *open source* aux communautés, lorsque c'est possible.

Comme nous l'avons déjà vu, les entreprises rationalisent leur système d'information. Un des participants au groupe de travail requiert systématiquement dans les appels d'offre l'indépendance des plateformes logicielles vis-à-vis du système d'exploitation et limite les technologies utilisées.

De plus, afin de comparer à périmètre équivalent les réponses aux appels d'offre, des administrations publiques et des entreprises imposent des règles de standardisation des briques et des exigences de modularité. Ils demandent en plus de la solution technique une estimation des **coûts opérationnels dans la durée**. Les fournisseurs doivent par conséquent adapter leurs solutions aux besoins et contraintes de l'entreprise, et non l'inverse. Par exemple, lorsqu'un modèle propriétaire est proposé, le fournisseur doit intégrer dans son offre tous les « à côté », comme les coûts de formation de la « techno » à l'ensemble des équipes internes qui fonctionnent 24h sur 24h, 365 jours par an.

4.4. Exemple de l'État

L'Etat positionne l'*open source* comme un levier stratégique pour la France depuis 2012, avec la circulaire Ayrault⁶. Cette circulaire pose les bases de l'utilisation de l'*open source* dans le cadre public après plusieurs années au cours desquelles la question de l'usage du logiciel libre a pu faire l'objet de nombreuses discussions dans l'administration. Elle propose une série d'orientations et de recommandations sur le bon usage du logiciel libre. Désormais, pour répondre aux besoins métiers, le logiciel libre doit être considéré à égalité avec les autres solutions, éditeurs ou mixtes.

Pour dégager le maximum d'efficacité aussi bien économique qu'en termes de qualité, il convient d'utiliser le logiciel libre de manière concertée et coordonnée. Un cadre de convergence des logiciels à privilégier dans le développement des systèmes d'information de l'Etat, défini en 2012, est maintenu en concertation interministérielle. Il touche en priorité les systèmes les plus déployés, sur les serveurs comme sur les postes de travail. Ce cadre ne fait pas obstacle à l'innovation par essai de nouvelles souches, qui pourront aider à l'évolution du cadre. Ce cadre ne rend pas non plus obligatoire l'évolution adaptative des applications existantes non conformes. Par contre il définit des versions de référence à privilégier et indique les solutions à abandonner, avec des réserves éventuelles pour des contextes d'usage particuliers. Il participe ainsi à la convergence progressive des contextes d'exploitation et à la mutualisation de certains moyens. A ce titre il doit être intégré dans tous les cadres technologiques des ministères et pris en compte à l'occasion de nouveaux développements et de refontes majeures. L'ensemble des logiciels libres préconisés se présente sous la forme du socle interministériel de logiciels libres (SILL) et est maintenu par la DINSIC (Direction Interministérielle des Systèmes d'Information et de Communication de l'Etat).

⁶ Cf. url http://circulaire.legifrance.gouv.fr/pdf/2012/09/cir_35837.pdf

5. Choix d'un logiciel libre : processus et critères d'analyse

Rechercher les alternatives libres implique d'évaluer la catégorie du *business model* du logiciel mais également les éléments techniques de la solution et le support associé. Des processus de choix et des critères d'analyse de logiciels libres ainsi que des processus de choix de support logiciel sont proposées dans ce chapitre.

5.1. Processus de choix d'un logiciel *open source*

Les entreprises ont une approche pragmatique dans les achats de logiciels. Elles choisissent le meilleur compromis **besoin métier/ prix/ support** en fonction de la criticité du besoin. Les **solutions libres sont systématiquement évaluées**. Cependant, si les logiciels libres ne répondent pas au besoin alors le choix se portera vers des produits éditeurs propriétaires.

La stratégie d'achat d'un logiciel est basée sur le **besoin utilisateur** (capacité du logiciel à couvrir le besoin) et la **valeur d'usage**. Il est en effet nécessaire de poser **une expression de besoin fonctionnel**. Cela évite à certains collaborateurs de venir avec un nom de logiciel à acheter qui ne correspond finalement pas au besoin.

Certaines entreprises ou administrations publiques définissent les composants à partir de la valeur d'usage et cartographient la chaîne de valeur rendue. Cela leur permet ensuite d'envisager des composants libres. Cette approche permet d'offrir une base, c'est-à-dire une solution par défaut qui répond au besoin utilisateur pour la majorité des différentes entités. Ainsi chaque entité intéressée par la valeur d'usage a la latitude de développer par-dessus ses solutions pour adapter ses spécificités. Des administrations publiques ou entreprises souhaitent architecturer leur système d'information de façon à offrir les éléments sous forme de micro-services.

Lors de son séminaire de stratégie logiciels libres tenu en avril 2017, la DINSIC a travaillé sur la segmentation suivante des besoins :

1. **Besoin de commodités** qui se traduit par une forte capacité de réversibilité avec des données standards et un marché mature. Dans ce cas, le prix est un facteur prépondérant et tous les types de solution peuvent être envisagés pour répondre au mieux au besoin, dont le SaaS. L'*open source* est toujours un plus quelle que soit la solution retenue ;
2. **Besoin d'expertise ou de niche**. Cet achat répond à un besoin particulier qui nécessite une expertise en plus du logiciel. Le choix de la solution prendra en compte ce besoin de service et la réversibilité ne se fera pas sans un effort important. Les solutions disponibles sur le marché sont généralement des *COTS (commercial off the shelf software)*, majoritairement propriétaires, même si le marché évolue de plus en plus vers du SaaS ou des souscriptions et bien qu'il soit possible de trouver des solutions *open source* avec des services d'expertise. Le choix s'oriente dans ce cas

sur des solutions qui garantissent la meilleure intégration, avec l'ordre de préférence suivant : *Brique déjà existante dans l'administration* > *SaaS ou COTS open-source* > *COTS propriétaire* > *SaaS propriétaire*. Cela afin de minimiser la dépendance à l'éditeur et le besoin de réactivité en cas d'augmentation des coûts de licences.

3. **Besoin critique spécifique ou besoin présentant un avantage concurrentiel ou de différenciation** autrement dit besoin unique sur un problème non résolu. Dans ce cas un développement spécifique est nécessaire et *l'open source* donne des garanties de pérennité par rapport à des *frameworks* de développements propriétaires. Pour un besoin unique non résolu, particulièrement dans la sphère publique, la DINSIC incite à ouvrir le code en *open source* au tout début des développements pour favoriser la collaboration, avoir la plus grande porosité avec un écosystème d'innovateurs et réduire les coûts de développement.

DINSIC, Laurent Joubert

5.2. Critères d'analyse des solutions *open source*

L'entreprise a tout intérêt à proposer une méthode d'analyse des logiciels *open source* commune à l'ensemble des collaborateurs qui prend en compte la prise de risque à long terme. Une liste de critères d'analyse est proposée ci-dessous afin de classer les logiciels *open source*.

- **Provenance du logiciel** : logiciel communautaire ou à souscription par ses auteurs ;
- **Propriété intellectuelle** autrement dit les droits patrimoniaux : hébergés dans une fondation, partagés entre contributeurs (cas de PostgreSQL par exemple) ou concentrés ?
- **Licence** : le logiciel est-il libre ? copyleft ?
- **Communauté d'utilisateurs** : qui utilise le logiciel ? La communauté utilisateurs est-elle composée d'industriels ? La communauté des utilisateurs a une influence proportionnelle à sa taille puisqu'elle a comme rôle principal de faire remonter les dysfonctionnements et les suggestions.
- **Communauté de développeurs** : quelle est la politique concernant la contribution ? Les contributions sont-elles encouragées ou limitées ? La qualité du logiciel est souvent proportionnelle au nombre de développeurs. Plus la communauté de développement s'étend, plus elle devient un gage de qualité et de réactivité.
- **Technique et sécurité** : le logiciel est-il mature, c'est à dire utilisé depuis un certain temps déjà et propose-t-il des évolutions régulières ou de niche ? Est-il sécurisé ?
- **Financements R&D centralisé ou communautaire** : qui financent la R&D du logiciel ? (Ce sont eux qui décident de la stratégie. Ils fournissent les priorités et définissent la *roadmap*).
- **Version communautaire** : est-elle complète ou est-ce un produit d'appel ?
- **Caractère innovant** : est-ce innovant ?

Franchir le pas de l'open source

EDF a défini une grille d'analyse des solutions *open source* avec une méthodologie associée. Elle inclut un code couleur qui permet de noter le logiciel.

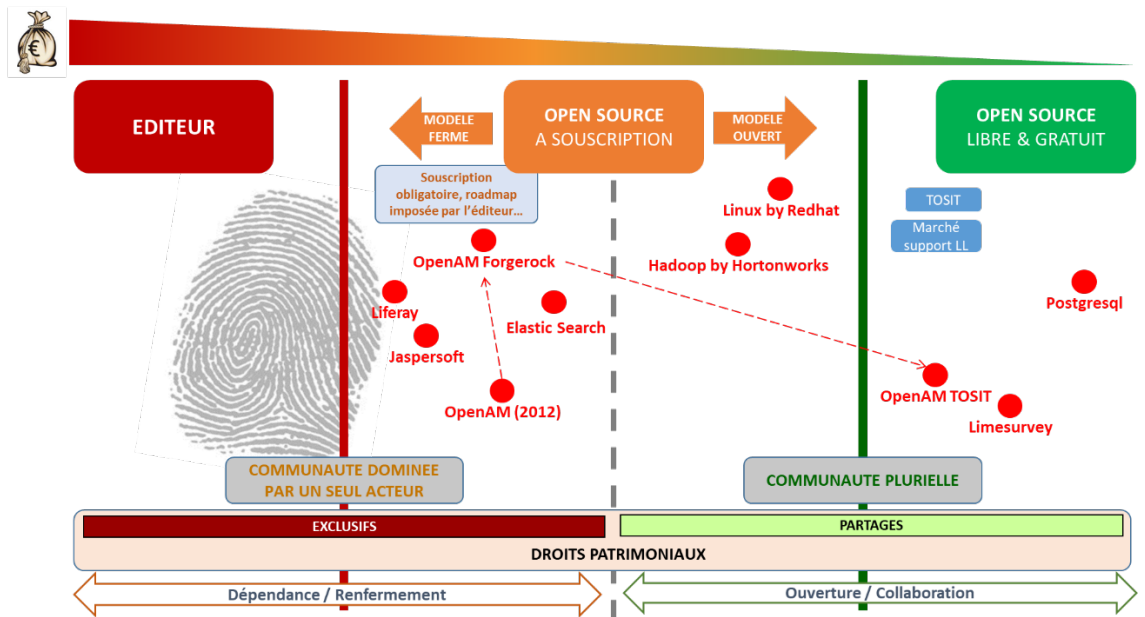
Lors d'un groupe de travail, EDF a présenté l'arbre de décision utilisé dans le choix de logiciels libres. Il intègre notamment le ROI par rapport aux solutions éditeurs et les produits à souscription sous conditions.

EDF, Aimery Assire

EDF visualise différents logiciels *open source* en les classant à l'aide du tableau ci-dessous. Cette classification qui inclut sur l'axe vertical la dépense, est fonction de la propriété intellectuelle, exclusive ou partagée, du type de communauté, plurielle ou dominée par un acteur, de la dépendance et l'enfermement ou de l'ouverture et la collaboration.

EDF, Aimery Assire

Modèles économiques des logiciels open-source



Source : EDF - 2017

Figure 3 : Modèles économiques des logiciels *open source*

Lorsque les solutions logicielles *open source* sont proposées par des *startups*, le grand groupe doit s'assurer que le taux de dépendance de la *startup* à la grande entreprise utilisatrice de solution est limité.

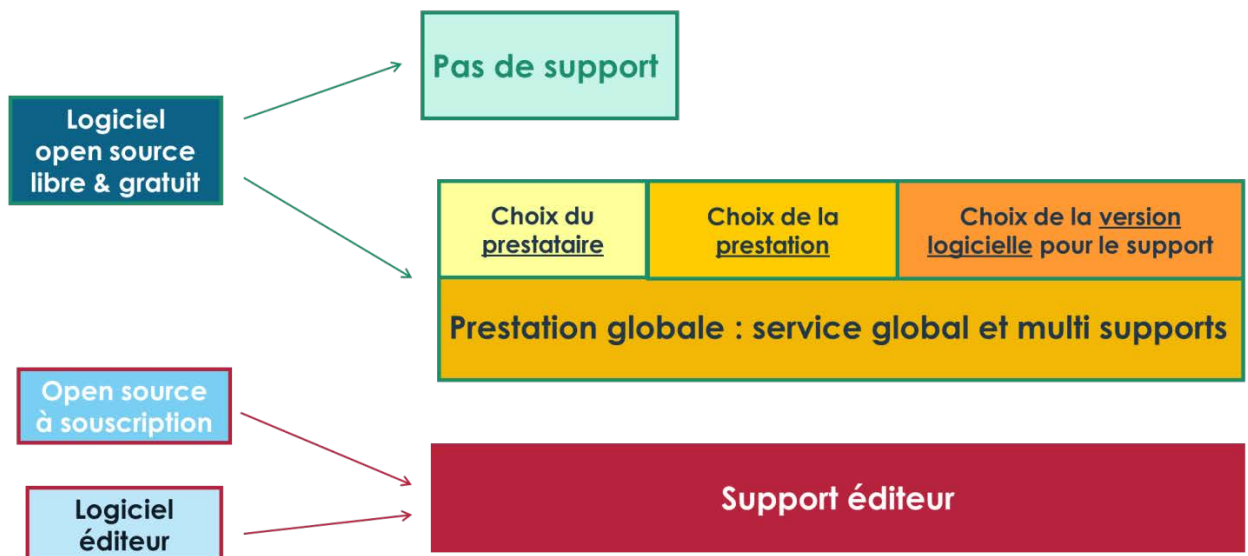
5.3. Processus de choix de support de solutions *open source* hors solution à souscription

Le logiciel libre nécessite comme tout logiciel une maintenance et un support. Pour les solutions *open source*, l'entreprise a la possibilité de définir le **périmètre de la prestation**, souplesse que n'offrent pas les éditeurs classiques. Cela inclut la possibilité de bénéficier d'une **prestation globale sur plusieurs logiciels *open source***, de **choisir les versions logicielles à supporter**, de **demande le rétroportage de tous les correctifs implémentés dans les versions plus récentes sur la version utilisée par l'entreprise**. Concernant le rétroportage, si la dernière version corrige un *bug* détecté, l'entreprise demande un rétroportage de la correction du *bug* sur la version utilisée : elle n'est donc pas obligée de passer à la dernière version pour avoir la correction du bug.

Dans son marché de support unique et dans le cadre de résolution de *bugs* dans sa version utilisée, la CNAM peut demander que soit rétroporté dans sa version, l'éventuel correctif qui pourrait exister dans une version plus récente. Cela permet d'éviter le risque classique de l'éditeur qui demande d'abord d'installer la dernière version avant de les analyser. Cela évite le risque de devoir faire des travaux lourds pour migrer vers la dernière version alors qu'il y a urgence à corriger les *bugs*. Le risque de la pression de l'obsolescence programmée est ainsi limité.

CNAM, Alain Issarni

Maintenance & Support



Source : Cigref

Figure 4 : Maintenance et support en fonction du logiciel

Certaines entreprises choisissent un seul prestataire pour assurer la maintenance et le support de l'ensemble des logiciels *open source* utilisés dans l'entreprise : **service global et**

Franchir le pas de l'open source

multi-supports. Il s'agit dans ce cas d'une prestation forfaitaire de support sur un nombre maximum et déterminé de logiciels *open source* lors de l'appel d'offres.

Les demandes classiques de support et maintenance sont assurées : la résolution des problèmes rencontrés par les utilisateurs lors de l'installation ou de l'utilisation des logiciels *open source* ; la réponse aux questions que pourront se poser les responsables de produits dans les domaines de l'industrialisation, de la préparation du déploiement, de la sécurité ou de la maintenance ; la couverture des facteurs de risques dont les risques opérationnels et de sécurité de chacune des versions (veille sécuritaire en particulier) ; le suivi de la maintenance évolutive du logiciel (versions mineures et majeures) couvert par le marché et disponible auprès de la communauté.

Les garanties proposées par les prestataires sont au moins identiques à celles des grands éditeurs. Elles incluent le déploiement, le support 7j/7j, la maintenance corrective, la garantie du bon fonctionnement, voire de la conformité de la solution et enfin les évolutions.

Les niveaux de service sont déterminés par l'entreprise pour répondre à ses besoins. En effet, les niveaux de support, critique et standard, sont déterminés avec les délais de réponse selon qu'il s'agisse d'un incident bloquant ou non bloquant. Les 4 actions définies pour chacun de ces supports sont le rappel de l'utilisateur, la réponse à une demande d'information, la fourniture d'une solution de contournement, la fourniture d'une solution définitive.

Des compétences sont nécessaires pour mener à bien les actions précédemment listées : support généraliste, support spécialisé ou autonomie. Un choix alors s'impose : internaliser ou sous-traiter ces compétences.

Conclusion

Les logiciels libres permettent aux entreprises et administrations publiques de gagner en **indépendance** et en **autonomie** vis-à-vis des grands fournisseurs.

Fortes d'une approche pragmatique, les entreprises étudient l'ensemble des solutions qui répondent au cas d'usage demandé, logiciels libres inclus. Elles choisissent la solution qui génère un gain avéré à la clé. Ce gain peut être de différente nature : économique, stratégique ou technique.

Les logiciels *open source* permettent aux entreprises de répondre à l'objectif de réduire le budget de la direction des systèmes d'information, même si les logiciels libres ont un coût. En effet, avant d'ajouter les solutions *open source* au champ de leurs solutions possibles, les entreprises acculturent les collaborateurs en leur présentant les spécificités de cet écosystème. Cela nécessite également, en interne ou par externalisation, une montée en compétences pour bien saisir la compréhension des jeux d'acteurs, ainsi que des ressources techniques pointues si l'entreprise souhaite développer certaines spécificités ou basculer dans le *make* pour répondre à son besoin.

Au regard de la digitalisation des métiers, l'entreprise doit continuer à trouver d'autres sources d'économies pour maîtriser son budget des systèmes d'information. L'engagement dans des solutions *open source* ne se fait pas en général sur le parc existant. Car s'en désengager coûterait d'abord trop cher et serait ensuite compliqué à justifier. Heureusement, une stratégie hybride avec des logiciels libres et éditeurs est possible. Disposer d'une solution alternative assure une marge de manœuvre et permet à l'entreprise de faire bouger les lignes lors de ses négociations avec les grands fournisseurs historiquement présents.

En outre, les logiciels *open source* confèrent des avantages stratégiques et technologiques à la transformation numérique des entreprises comme l'attractivité des talents et le développement des compétences. De plus, les solutions *open source* répondent aux enjeux actuels, le *time to market*, la maîtrise du système d'information, l'agilité à l'échelle de l'entreprise et l'*open innovation*. Convaincues de la puissance de l'intelligence collective et des avantages à mutualiser investissements, développements logiciels ou partages d'expérience au sein de GIE ou d'associations, les entreprises et administrations publiques s'organisent pour collaborer.

Ce sont autant de raisons pour lesquelles certaines entreprises, fortes du soutien du Comex, ont institué une stratégie « *open source first* ».

Les logiciels *open source*, devenus courants, sont aujourd'hui principalement implémentés sur les couches basses du système d'information. Les entreprises souhaitent augmenter le nombre de solutions libres sur les couches plus hautes. Elles veulent explorer les applications métier du logiciel libre. Le Cercle « Alternatives aux grands fournisseurs » du Cigref va se poursuivre avec pour objectif d'identifier des solutions métiers *open source*.

À PROPOS DU CIGREF ACTEUR DE LA SOCIÉTÉ NUMÉRIQUE

Association des grandes entreprises et administrations publiques françaises, le Cigref se donne pour mission de développer leur capacité à intégrer et maîtriser le numérique.



RÉSEAU DE GRANDES ENTREPRISES

Association loi 1901 créée en 1970, le Cigref n'exerce aucune activité lucrative. En 2018, il regroupe près de **150 grandes entreprises et organismes français utilisateurs de systèmes numériques**, dans tous les secteurs d'activité.



ACTEUR DU NUMÉRIQUE

Par la qualité de sa réflexion et la représentativité de ses membres, **il est un élément fédérateur et acteur important de la société numérique.**



AU SERVICE DE SES MEMBRES

Sa gouvernance est assurée par **15 Administrateurs**, élus en Assemblée générale. Son activité est animée par une équipe de **10 permanents**.