

1. Introduction :

On entend souvent parler de logiciels libres ou open source mis en opposition à ceux **propriétaires et privateurs**. Mais qu'est-ce que c'est exactement, et pourquoi s'en soucier ?

2. Un peu d'histoire :

L'informatique était essentiellement accessible aux chercheurs au début des années 1970. Ils partageaient couramment entre eux les codes sources des programmes qu'ils développaient. C'est, par analogie, une sorte de « recette », qui une fois « cuisinée » nous donnera notre « gâteau » (**Figure 1**). Remplacer ici la « cuisine » par un processus nommé compilation, qui transforme la « recette » en binaire – suite de zéros et d'uns –, exécutable par un ordinateur, mais illisible pour un programmeur – un « cuisinier » pour continuer l'analogie. La possession du code source est donc impérative pour adapter, améliorer ou réparer un logiciel .

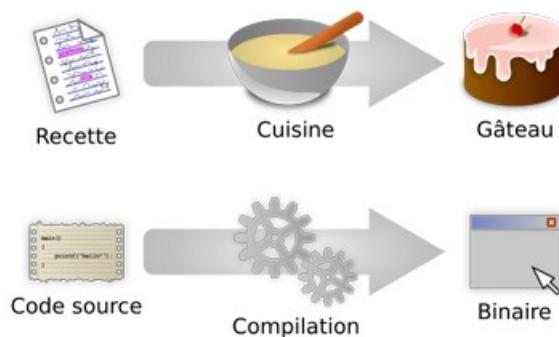


Figure 1: Illustration de l'analogie.

L'échange des sources entre programmeurs bénéficiait à tous ; chacun pouvait adapter les logiciels à ses besoins, tout en participant à leur amélioration, en partageant les modifications apportées.

À savoir qu'à l'époque, les entreprises du secteur informatique basaient leurs revenus sur la fourniture de matériel. Mais c'est à cette période qu'elles ont commencé à s'orienter vers la vente de logiciels. Cette commercialisation a entraîné une forte protection des codes sources sous couvert du secret d'entreprise.

C'est en 1979 que **Richard Matthew Stallman**, étudiant au **MIT(Massachusetts Institute of Technology)** , un prestigieux institut de recherche américain, se heurte à ce nouveau problème. Ayant participé à cette culture de partage, il est choqué qu'un collègue refuse de lui fournir le code source

d'un logiciel de pilotage d'une imprimante, qu'il voulait corriger. Mais celui-ci étant développé par **Xerox**, son détenteur a signé une clause de confidentialité avec l'entreprise.

Cette anecdote témoigne de la disparition de la culture de Stallman. Pour la préserver, il fonde en 1985 avec des collègues la Free Software Foundation (FSF, Fondation pour le Logiciel Libre), dont la notion de logiciel libre est issue. Il publie ensuite son logiciel Emacs sous la nouvelle licence de l'organisation, la GNU General Public License, assurant la disponibilité du code source.



Figure 2: Le GNU (prononcé "gnou" en anglais), mascotte de la FSF et de leurs licences.

(Tête de GNU contrasté par Aurélio A. Heckert sous licence CC BY-SA 2.0 © 2003, Free Software Foundation, Inc.)

1. **Définition :** L'Open Source fait référence à un modèle de développement et de distribution de logiciels où le code source est ouvert, c'est-à-dire accessible et modifiable par tous ses utilisateurs.

Contrairement aux logiciels dits « **propriétaires** », dont le code source est généralement gardé secret et soumis à des licences d'usage restrictives. L'Open Source permet à ses utilisateurs non seulement **d'utiliser le logiciel**, mais également de **l'analyser**, de le **modifier** et de le **redistribuer**. Cette approche est fondée sur la **transparence**, la **collaboration** et la **amélioration continue** par la communauté des développeurs.

Ces logiciels offrent une alternative solide et flexible aux logiciels commerciaux, permettant à des entreprises, des gouvernements, des organisations et des individus :

- D'adopter et d'adapter ces logiciels à leurs besoins spécifiques sans être liés à des contrats restrictifs ou à des frais de licence élevés ;
- De s'assurer de la transparence de fonctionnement et des traitements effectués par le logiciel ;
- Une certaine assurance de pérennité : si l'éditeur est défaillant, le logiciel pourra être repris par d'autres, notamment s'il dispose d'une base d'utilisateurs ou clients suffisante.

2. Les logiciels libres “ free software or open source software ???” en anglais

En français, "logiciel libre" se traduit en anglais par "free software" ou "open source software" (OSS), mais il est important de noter que ces deux termes, bien que souvent utilisés de manière interchangeable, ont des nuances différentes. "Free software" met l'accent sur la liberté de l'utilisateur, garantissant quatre libertés fondamentales comme **l'utilisation, la copie, l'étude et la modification du logiciel**, tandis que "open source software" met davantage l'accent sur les avantages pratiques du partage du code source, comme la collaboration et l'amélioration collaborative.

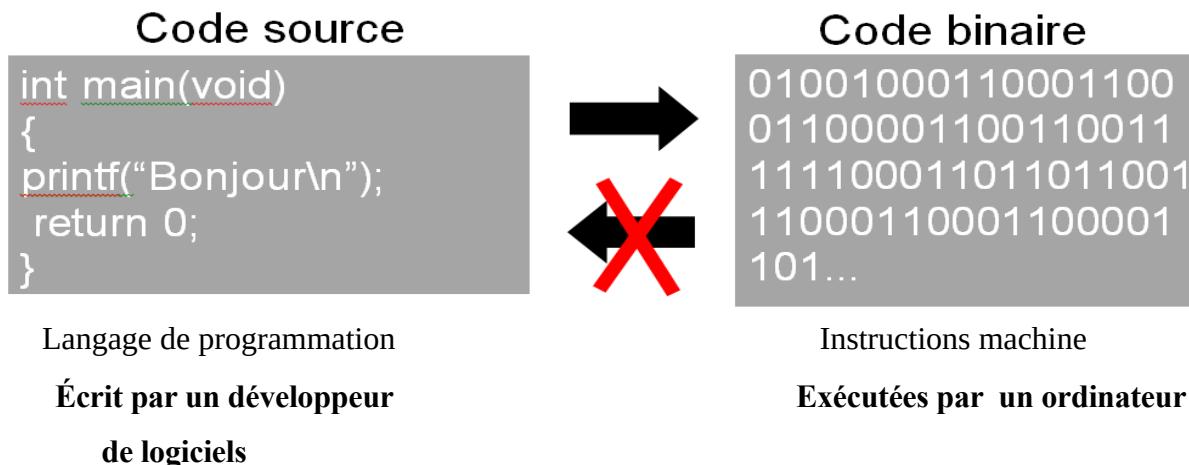


3.

Figure 03 : Logos logiciels libres.

5. Qu'est-ce que le code source ?

Le code source est l'ensemble des instructions écrites par un programmeur dans un langage de programmation lisible par l'humain, qui définissent le fonctionnement d'un logiciel ou d'une application. Il s'agit du texte original d'un programme qui est ensuite traduit, via un compilateur ou un interpréteur, en instructions compréhensibles par l'ordinateur (code machine) pour qu'il puisse exécuter la tâche souhaitée.



6. Que veux dire libre dans Logiciel libre ?

Dans « **Logiciel libre** », le mot « **libre** » fait référence aux **libertés d'utilisation, d'étude, de modification et de redistribution** du logiciel, et **non à la gratuité**. Ces quatre libertés fondamentales, définies par le projet GNU, garantissent aux utilisateurs le droit d'exécuter le programme pour tous les usages, d'étudier son fonctionnement et de l'adapter à leurs besoins, de redistribuer des copies du logiciel, et d'améliorer le programme et de publier ces améliorations pour en faire profiter tout le monde.

Le projet GNU est un acronyme récursif signifiant "**GNU's Not Unix**" (GNU n'est pas Unix). Le projet GNU, fondé par Richard Stallman en 1983, vise à créer un système d'exploitation libre, fonctionnellement **similaire à Unix** mais **sans utiliser de code source propriétaire**. Il est composé de nombreux logiciels libres et est souvent associé au noyau Linux pour former le système d'exploitation **GNU/Linux**.

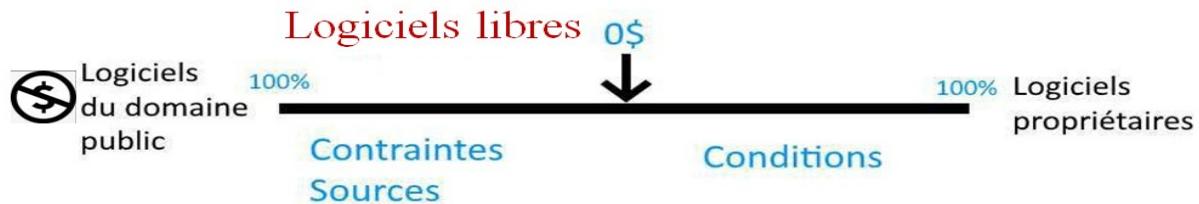
➤ **Différents niveaux de liberté /**

- ◆ Avec ou sans le code source,
- ◆ Avec ou sans contraintes

Exemple de contrainte : distribution avec le nom de l'auteur

➤ **Domaine public** signifie :

- Typiquement sans le code source
- Sans aucune contrainte,



7- Différentes façons de distribuer le logiciel

- Les contraintes et conditions associées aux logiciels et au code source, sont définies dans une **licence logicielle**.

◆ Définition

Une **licence logicielle** est un **contrat juridique** qui précise **les droits et obligations** du créateur du logiciel (le titulaire des droits d'auteur) et de l'utilisateur (ou du développeur tiers).

◆ Ce qu'elle définit

La licence précise notamment :

- **Les droits d'utilisation** : ce que l'utilisateur a le droit de faire (exécuter, installer, copier, etc.).
 - **Les restrictions** : ce qu'il n'a **pas** le droit de faire (modifier, redistribuer, utiliser à des fins commerciales, etc.).
 - **L'accès au code source** : certaines licences permettent l'accès et la modification du code source (licences libres), d'autres non (licences propriétaires).
 - **Les conditions de redistribution** : comment le logiciel peut être partagé ou intégré dans d'autres projets.
 - **Les responsabilités et garanties** : clauses limitant la responsabilité du créateur en cas de problème.
-

◆ Types de licences

- **Logiciel propriétaire** : le code source est fermé, et l'utilisation est strictement encadrée (ex. : Microsoft Windows, Adobe Photoshop).
- **Logiciel libre / open source** : le code source est ouvert et modifiable, sous certaines conditions (ex. : GPL, MIT, Apache, BSD).

8- La principale organisation de licences libres est



- **1.4 milliards d'oeuvres** sont disponibles depuis 2001 publiquement sous une licence Creative Common
- Association à but non lucratif américaine dont l'objectif est de proposer une solution légale (licences) aux personnes souhaitant libérer leurs œuvres.

9- Licences libres de Creative Common



Les licences Creative Commons (CC) sont un ensemble de licences libres conçues pour permettre aux créateurs de partager leurs œuvres (textes, images, musique, vidéos, etc.) de manière souple et légale, tout en conservant certains droits.

Elles ne sont **pas destinées aux logiciels** (pour cela, on utilise plutôt des licences comme la GPL, MIT, Apache...), mais plutôt aux **œuvres de création**.

◆ Objectif des licences Creative Commons

Elles permettent à l'auteur de dire clairement :

« Vous pouvez utiliser mon œuvre, mais à telles conditions. »

Ainsi, au lieu du traditionnel « **tous droits réservés** », on parle de :

« **certains droits réservés** ».

◆ Les 6 principales licences Creative Commons

Chaque licence combine plusieurs **conditions** :

Symbol	Signification	Description
BY	Attribution	L'auteur doit être cité.
NC	Non Commercial	L'œuvre ne peut pas être utilisée à des fins commerciales.
ND	No Derivative	Aucune modification ni adaptation n'est permise.
SA	Share Alike	Les œuvres dérivées doivent être partagées sous la même licence.

Voici les 6 licences CC officielles :

Licence	Nom complet	Conditions principales
CC BY	Attribution	Utilisation, modification et redistribution autorisées, même à des fins commerciales, à condition de citer

Licence	Nom complet	Conditions principales
CC BY-SA	Attribution – Partage dans les mêmes conditions	l'auteur. Comme CC BY, mais les œuvres dérivées doivent être sous la même licence. (Ex. Wikipédia)
CC BY-ND	Attribution – Pas de modification	Redistribution autorisée, mais sans modification.
CC BY-NC	Attribution – Non commercial	Utilisation et modification autorisées, mais pas à des fins commerciales.
CC BY-NC-SA	Attribution – Non commercial – Partage dans les mêmes conditions	Même que la précédente, mais les dérivés doivent garder la même licence.
CC BY-NC-ND	Attribution – Non commercial – Pas de modification	La plus restrictive : partage autorisé uniquement sans modification ni usage commercial.

◆ **Cas particulier : CC0 (Creative Commons Zero)**

C'est une **dédicace au domaine public** :

L'auteur **renonce à tous ses droits**, permettant à quiconque d'utiliser, de modifier ou de redistribuer l'œuvre **sans aucune condition**.

10- Qui sont les développeurs de Logiciels libres ?

Les **développeurs de logiciels libres** ne sont pas un seul groupe précis — ce sont **des personnes ou des organisations** qui créent, maintiennent ou contribuent à des logiciels dont le **code source est ouvert et librement accessible**.

Voici une explication détaillée 

◆ **1. Les types de développeurs de logiciels libres**

 **a) Les développeurs indépendants**

Ce sont des **programmeurs individuels** qui contribuent bénévolement à des projets open source, par passion, pour apprendre, ou pour améliorer un outil qu'ils utilisent.

Exemples :

- Linus Torvalds → créateur du **noyau Linux**
- Guido van Rossum → créateur du **langage Python**
- Richard Stallman → fondateur du **projet GNU** et promoteur du **logiciel libre**

b) Les communautés de développeurs

Ce sont des **groupes collaboratifs** qui maintiennent et font évoluer des projets.

Ces communautés travaillent souvent via des plateformes comme **GitHub**, **GitLab**, ou **SourceForge**.

Exemples :

- **Debian** → communauté qui maintient la distribution GNU/Linux Debian
 - **Mozilla Foundation** → développe le navigateur **Firefox**
 - **Apache Software Foundation** → gère des projets comme **Apache HTTP Server**, **Hadoop**, etc.
-

c) Les entreprises

Beaucoup d'entreprises soutiennent ou développent **des logiciels libres** parce qu'elles y voient un intérêt économique ou stratégique (innovation, sécurité, image, etc.).

Exemples :

- **Red Hat** → maintient **Red Hat Enterprise Linux** et contribue à **Fedora**
 - **Canonical** → développe **Ubuntu**
 - **Google** → contribue à **Android**, **Chromium**, **Kubernetes**, etc.
 - **IBM**, **Intel**, **Microsoft** et d'autres participent activement à des projets open source
-

d) Les fondations et organisations à but non lucratif

Elles coordonnent, financent et protègent les projets libres.

Exemples :

- **Free Software Foundation (FSF)** : fondée par Richard Stallman, promeut le logiciel libre.
 - **Linux Foundation** : soutient le développement du noyau Linux et de nombreux autres projets.
 - **Apache Software Foundation** : gère des centaines de projets open source.
 - **Mozilla Foundation** : promeut un Internet libre et ouvert.
-

2. Motivations des développeurs de logiciels libres

- **Philosophiques / éthiques** → défendre la liberté d'utiliser, d'étudier et de modifier le code
- **Techniques** → améliorer la qualité du logiciel grâce à la collaboration

- **Éducatives** → apprendre en collaborant sur des projets concrets
- **Professionnelles** → se faire connaître et valoriser ses compétences
- **Communautaires** → contribuer à un bien commun numérique

11- Motivations de programmeurs(ses) de logiciels libres

Les programmeurs et programmeuses de logiciels libres sont motivé(e)s par une multitude de facteurs, souvent combinés. Voici quelques-unes des motivations les plus courantes :

1. **Liberté et contrôle** : L'un des piliers du logiciel libre est la liberté de l'utilisateur d'exécuter, d'étudier, de modifier et de distribuer le logiciel. Pour les développeurs, cela se traduit par la liberté de créer ce qu'ils veulent, comme ils le veulent, sans les contraintes commerciales ou propriétaires. Ils ont un contrôle total sur leur code et la direction du projet.
2. **Collaboration et communauté** : Le développement de logiciels libres est intrinsèquement collaboratif. Les développeurs sont souvent motivés par l'opportunité de travailler avec des pairs partageant les mêmes idées, d'apprendre des autres et de contribuer à un projet plus vaste. La reconnaissance au sein de la communauté est également un facteur important.
3. **Apprentissage et amélioration des compétences** : Contribuer à un projet open source est un excellent moyen d'apprendre de nouvelles technologies, de maîtriser de nouveaux langages de programmation, d'améliorer ses compétences en résolution de problèmes et d'acquérir de l'expérience en travaillant sur des bases de code complexes. C'est un environnement propice à l'apprentissage continu.
4. **Résolution de problèmes personnels** : Souvent, les développeurs commencent à contribuer à des projets open source parce qu'ils ont rencontré un problème ou un besoin spécifique qu'ils pensent pouvoir résoudre avec un nouveau logiciel ou en améliorant un logiciel existant. C'est une motivation très pratique.
5. **Idéalisme et éthique** : Beaucoup de développeurs de logiciels libres sont animés par une forte conviction en les valeurs du mouvement open source, comme le partage de connaissances, la transparence et l'accès universel à la technologie. Ils croient en la construction d'un monde meilleur grâce au logiciel libre.
6. **Réputation et reconnaissance** : Contribuer à des projets open source bien connus peut améliorer la réputation d'un développeur, tant au sein de la communauté qu'auprès des employeurs potentiels. C'est un excellent moyen de montrer ses compétences et son engagement.
7. **Fun et passion** : Pour beaucoup, le développement de logiciels est une passion. Travailler sur des projets open source leur permet de s'amuser, d'explorer des idées créatives et de passer du temps à faire ce qu'ils aiment, sans la pression des délais ou des exigences commerciales.

8. **Désir de faire une différence** : Les développeurs peuvent être motivés par le désir de créer un logiciel qui aura un impact positif sur le monde, qu'il s'agisse d'un outil éducatif, d'une solution pour les organisations à but non lucratif, ou d'une technologie qui aide les gens au quotidien.
9. **Opportunités professionnelles** : Bien que ce ne soit pas la motivation principale pour beaucoup, l'expérience acquise en travaillant sur des projets open source peut ouvrir des portes à des opportunités d'emploi, en particulier dans les entreprises qui valorisent l'expertise en logiciels libres.
10. **Curiosité et exploration** : Certains développeurs sont simplement curieux de savoir comment les choses fonctionnent et aiment explorer de nouvelles technologies. Le logiciel libre leur donne l'occasion de plonger dans le code et de comprendre les mécanismes internes des applications.

En résumé, les motivations sont diverses et souvent entremêlées, reflétant un mélange de valeurs personnelles, d'intérêts professionnels et d'un désir de contribuer à un bien commun.

12- Les développeurs de logiciels libres ne sont pas seulement des programmeurs(ses)

Certains utilisateurs participent au développement de logiciels libres :

- ◆ Tests et rapports de bugs
- ◆ Rédaction de documentation
 - ◆ Traduction
- ◆ Infographie et sonorisation

13- Avantages techniques



Indépendance et Souveraineté technologique

L'indépendance technologique est devenue un enjeu majeur. L'Open Source permet de se libérer de la dépendance envers des éditeurs de logiciels propriétaires monopolistiques et des géants technologiques (GAFAM) qui peuvent imposer des conditions commerciales trop restrictives sans négociation ou contre pouvoir.

- **GAFAM** est l'acronyme des cinq plus grandes entreprises technologiques américaines : Google, Apple, Facebook (aujourd'hui Meta), Amazon et Microsoft. Ces entreprises sont devenues des géants mondiaux du numérique grâce à leur domination du marché et à leur influence grandissante sur l'économie.
- ◆ **Google (Alphabet)** : Moteur de recherche, publicité en ligne et systèmes d'exploitation.
- ◆ **Apple** : Matériel informatique (iPhone, Mac), logiciels et services.
- ◆ **Facebook (Meta)** : Réseaux sociaux comme Facebook, Instagram et WhatsApp.
- ◆ **Amazon** : Commerce en ligne, cloud computing et services de streaming.
- ◆ **Microsoft** : Systèmes d'exploitation (Windows), logiciels de productivité et services cloud.

 **Innovation Collaborative**

L'**Open Source** encourage une innovation collaborative unique. Les projets Open Source bénéficient des contributions d'une grande diversité de développeurs, ce qui permet de capitaliser sur des idées et des approches différentes. Cette collaboration conduit à des améliorations continues et à l'introduction rapide de nouvelles technologies dans les logiciels. L'absence de contraintes commerciales dans les projets Open Source permet également aux développeurs d'expérimenter et d'innover librement, sans attendre l'approbation d'une entreprise ou d'un comité de direction.

**Flexibilité et Adaptabilité**

Un des atouts majeurs de l'Open Source est sa flexibilité. Les entreprises peuvent adapter le logiciel à leurs besoins spécifiques en modifiant le code source. Cela permet une personnalisation poussée, impossible à atteindre avec des logiciels propriétaires.

L'adaptabilité des logiciels Open Source permet également une meilleure intégration avec d'autres systèmes. Les entreprises peuvent ajuster les solutions à leurs environnements informatiques sans avoir à payer pour des modules supplémentaires ou des versions "premium".

**Pérennité et Durabilité**

Les logiciels propriétaires peuvent être abandonnés par leurs éditeurs ou voir leurs conditions de licence changer de manière arbitraire. À l'inverse, les logiciels Open Source sont généralement pérennes, car le code source est ouvert et accessible. Même si l'équipe de développement d'origine ou l'éditeur cesse son activité, la communauté ou d'autres entreprises peuvent reprendre le projet et en assurer la continuité.

Cela confère une plus grande sécurité à long terme, car les organisations ne sont pas dépendantes des décisions commerciales d'une seule entité pour continuer à utiliser le logiciel et un garde-fou est présent au cas où l'équipe de développement ou l'éditeur sortiraient des limites acceptables par la majorité des utilisateurs.

**Sécurité**

L'Open Source a parfois été perçu comme moins sécurisé que les logiciels propriétaires, mais cette idée est de plus en plus réfutée. En fait,

les logiciels Open Source sont souvent plus sûrs précisément parce que leur code source est ouvert et que tout le monde peut l'examiner. Cette transparence permet de détecter et de corriger rapidement les failles de sécurité.



Interopérabilité et Compatibilité

Les logiciels Open Source sont souvent conçus pour être interopérables avec d'autres systèmes et normes. Cela permet aux entreprises d'intégrer plus facilement des solutions Open Source dans des infrastructures hétérogènes, tout en assurant la compatibilité avec des outils et des systèmes déjà en place. Les standards ouverts sont un principe fondamental de l'Open Source, garantissant que les systèmes puissent interagir sans verrouillage propriétaire.



Maintenabilité

Les projets Open Source sont souvent conçus pour être facilement maintenables et évolutifs. Les entreprises peuvent contribuer directement à l'amélioration du code ou faire appel à des prestataires externes pour assurer la maintenance de leurs solutions Open Source. L'accès au code source permet également aux équipes techniques internes d'assurer une maintenance proactive, d'ajuster les systèmes en fonction des besoins, et de corriger rapidement les erreurs sans devoir attendre l'intervention de l'éditeur.

14- Avantage Économiques

➤ Mutualisation des développements

L'un des avantages économiques majeurs de l'Open Source est la mutualisation des coûts. Ce modèle repose sur une collaboration ouverte où plusieurs organisations partagent les coûts de développement, de maintenance et d'amélioration des logiciels ou composants. Contrairement aux solutions propriétaires, où chaque entreprise paie des licences et souvent des frais de maintenance élevés, l'Open Source permet de distribuer ces coûts entre un nombre potentiellement illimité de parties prenantes.

➤ Réduction des Coûts de Licence

Les logiciels ou composants Open Source techniques sont souvent gratuits ou disponibles sous des licences très permissives. Cela permet aux entreprises de

réaliser des économies importantes en supprimant les coûts de licences, qui peuvent représenter une part considérable du budget informatique des organisations. Même si certaines solutions Open Source nécessitent des services payants (versions premium, abonnements, support, formation, etc.), ces coûts sont généralement inférieurs à ceux des licences des logiciels propriétaires.

15- Avantages Sociétaux

Partage des Connaissances

L'Open Source est basé sur le partage des connaissances. En ouvrant le code source, les développeurs permettent à d'autres de comprendre comment fonctionne le logiciel et de l'améliorer. Cela contribue à la création d'une culture de collaboration et d'apprentissage dans l'ensemble de la communauté technologique. Les projets Open Source servent également de base à des projets éducatifs et académiques. Les étudiants et les chercheurs peuvent accéder librement aux technologies avancées, apprendre des concepts complexes en les appliquant directement et contribuer eux-mêmes aux projets.

Développement Communautaire et Collaboratif

L'Open Source crée et favorise le développement de communautés de développeurs et d'utilisateurs. Ces communautés sont des lieux de collaboration où les gens échangent des idées, partagent des solutions à des problèmes communs et participent à la co-création de nouvelles technologies.

➤ *Soutien à l'Économie*

L'adoption de solutions Open Source peut renforcer l'économie locale et nationale en favorisant la création d'entreprises locales spécialisées dans le développement, la personnalisation et le support des logiciels Open Source. Plutôt que de dépendre de grands éditeurs étrangers, les entreprises locales peuvent développer des compétences en interne et offrir des services et solutions souveraines à d'autres entreprises ou organisations publiques. Cette approche encourage également le développement des talents locaux, la formation de nouveaux professionnels et la création d'un écosystème technologique durable.

16- Pourquoi utiliser du logiciel libre

- • accès simple à des nombreux logiciels de qualité

- • essai immédiat
- • aspect budgétaire très intéressant
- • capacité d'avoir de l'aide de la communauté
- • capacité de comprendre
- • capacité de faire des contrôles du code (*debug* et audits de sécurité)
- • apporter des améliorations ou des personnalisations
- • diffusion rapide (*time to market*)
- • aspects idéologiques et éthiques
- • pérennité des solutions
- • souveraineté numérique : état, entreprises, particuliers

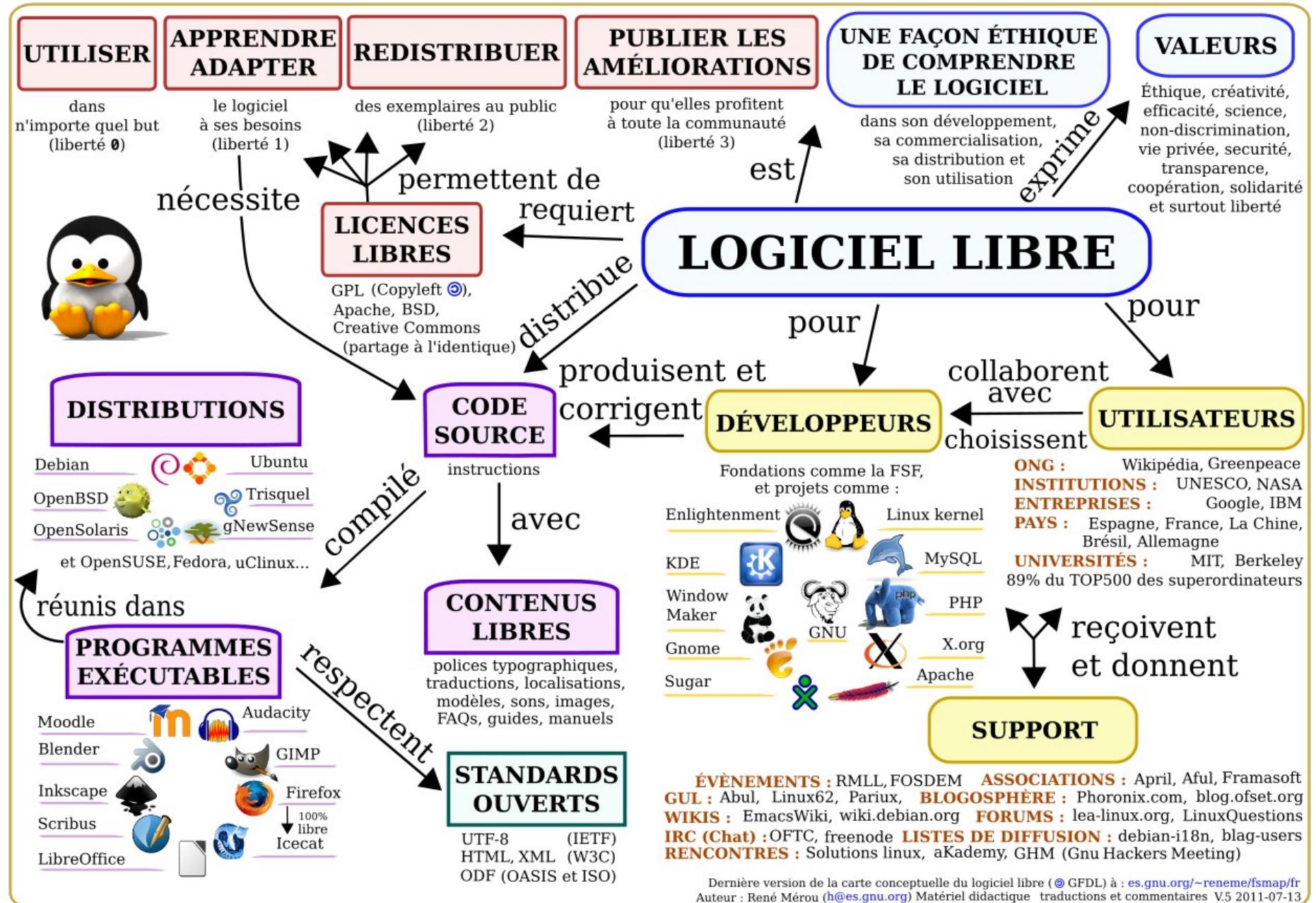
17- Libre vs. Gratuit

- ◆ En anglais : *free* = gratuit ou libre !!!
- ◆ Dans le cas du logiciel, libre ne veut pas nécessairement dire gratuit.
- ◆ Le caractère *commercial* est généralement compatible avec le caractère libre.
- ◆ Mais en pratique, la grande majorité des logiciels libres sont gratuits.
- ◆ **Libre et gratuit** : Linux, FreeBSD, Android, Firefox, Apache, Python, . . .
- ◆ **Libre et non gratuit** : distributions commerciales de Linux
- ◆ **Non libre et gratuit** : Internet Explorer, Acrobat Reader, freeware, . . .
- ◆ **Non libre et non gratuit** : je vous laisse deviner. . .

18- Conclusions et perspectives

- Le logiciel libre de plus en plus diffusé, des offres soutenues par des prestataires
- Logiciel libre + formats/standard ouverts => le couple indispensable.
- Le nuage : garder les pieds sur terre et ne pas jouer sans solides garanties ...

Carte conceptuelle sur le logiciel libre de Wikipédia



Alternatives libres

Ci-dessous une liste non exhaustive de logiciels libres, avec leurs équivalents privateurs s'il y a.

Fonction	Logiciels privateurs	Alternatives libres	Description
Suite bureautique	Microsoft Office	LibreOffice	Suite bureautique complète avec éditeur de texte, tableur, mise en page, présentation et base de données. Ce document a été rédigé avec LibreOffice Writer.
Lecteur PDF	Adobe Acrobat	Sumatra PDF	Lecteur de fichier PDF simple et léger.
Navigateur Web	Google Chrome Microsoft Edge Apple Safari	Mozilla Firefox	Convivial et accès sur la protection de la vie privée, personnalisable avec des modules, comme des bloqueurs publicitaires.
Client mail	Microsoft Outlook	Mozilla Thunderbird	Regroupe toutes vos boîtes mails dans une seule interface dédié, sans passer par votre navigateur Web. Propose de nombreuses fonctions comme un carnet de contact et le filtrage des spams.
Correcteur grammatical	Synapse Cordial	Grammalecte	Là où de nombreuses logiciels offrent un correcteur d'orthographe, Grammalecte vient fait de même avec la grammaire. Compatible avec Firefox, Thunderbird et LibreOffice.
Cartographie mentale	OpenGenius Ayoa	Freeplane	Créez des cartes mentales mises en forme exportables en PDF pour les diffuser.
Prise de notes	Evernote	CherryTree	Hiérarchise vos notes mises en forme dans un fichier qui peut être protégé par mot de passe.
Lecteur multimédia	Microsoft Windows Media Player	VLC	Lit la majorité des formats de vidéos et musicaux, à partir de fichiers ou de disques.
Compression	WinRAR	7-Zip	Ouvre et compresse les archives de multiples formats, dont ZIP, RAR et TAR.
Maintenance système	Piriform CCleaner	BleachBit	Maintient et nettoie le système d'exploitation et de nombreux logiciels.
Gestionnaire de mots de passe	1Password	KeePassXC	Centralise vos identifiants et mots de passe dans une base de données stockée hors

	Dashlane		ligne.
Personnalisation	Stardock Start10	Open Shell	Retrouvez le menu Démarrer classique sous Microsoft Windows 8 et 10.
Traitement d'images	PhotoFiltre	Photoflare	Modifiez vos photos et créez des graphismes avec ce logiciel conçu spécialement pour les éditions simples et les néophytes.
	Adobe Photoshop	GIMP	Complet et puissant, mais pas nécessairement intuitif, ces capacités vont de la retouche photo au dessin avec tablette graphique.
Dessin vectoriel	Adobe Illustrator	Inkscape	Créez des illustrations, graphiques, diagrammes et mises en page en partant d'opérations
			sur des formes géométriques simples.
Edition audio	Apple GarageBand Image-Line FL Studio Steinberg Cubase	Audacity	Mixez, corrigez et éditez facilement vos pistes audios avec ce puissant outil. Il permet aussi l'enregistrement via un microphone.
Musique assistée par ordinateur		LMMS	Ce logiciel très complet donne à tous l'accès à la musique assistée par ordinateur professionnelle.
Edition vidéo	Microsoft Windows Movie Maker	OpenShot	Spécialement conçu pour une prise en main rapide, il reste tout de même capable de réaliser des montages complexes.
	Sonic Foundry Vegas Pro	Kdenlive	Complet, il se rapproche des logiciels privateurs professionnels.
Architecture	Micro Applications 3D Architecte	Sweet Home 3D	Réalisez des plans architecturaux et simulez leurs aménagements.
Modélisation 3D	BrickLink Studio	LeoCAD	Inventez vos propres modèles à partir de briques LEGO.
	Autodesk 3ds Max	Blender	Performant mais complexe, il est particulièrement utilisé pour le jeu vidéo et l'animation 3D. Il permet même l'édition vidéo et inclus un moteur de jeu complet.
Photogrammétrie	Agisoft Metashape	Regard3D	Créer des modèles 3D à partir de séries de photos, sans le besoin d'un scanner dédié.
Moteur de jeu	Unity Yoyo Games Game Maker	GDevelop	Créez vos propres jeux vidéos 2D avec cet outil conçu pour être utilisé facilement, même sans expérience préalable.
Conception assistée par ordinateur	Autodesk AutoCAD	LibreCAD	Dessinez des plans et autre schémas techniques en 2D.
		OpenSCAD	Créez des modèles 3D grâce à son puissant langage de programmation.
		FreeCAD	Conceptualisez plans et pièces techniques pour une réalisation matérielle.