

Caractéristiques des logiciels

Table des matières

Objectifs	2
Modèles de déploiement	2
Installation locale	2
SaaS	2
Auto-hébergement	3
Coût initial et coûts récurrents	5
Types de licences de logiciels	5
Logiciel libre	5
Logiciel <i>open source</i>	7
FOSS et FLOSS	7
Licences libres et <i>open source</i>	7
Licence propriétaire	9
Pourquoi se soucier des conditions de licence?	9
Références	9



Objectifs

- Comprendre les différences entre les principaux modèles de déploiement des logiciels, ainsi que leurs avantages et inconvénients
- Comprendre les notions de base d'architecture de haut niveau des logiciels
- Connaître les notions de coût initial et de coûts récurrents selon le modèle de déploiement
- Connaître les différents types de licences de logiciels et comprendre les restrictions propres à chacune

Modèles de déploiement

Avant de choisir un logiciel, il faut vérifier ce qu'implique son déploiement :

- Est-ce un logiciel à installer localement sur les postes de travail?
- Doit-on le déployer sur un serveur? (auto-hébergement ou *self-hosting*)
- Est-ce un SaaS?
- A-t-on le choix entre une version SaaS et une version auto-hébergée?

Installation locale

L'installation locale est le type de déploiement que la plupart des gens connaissent le mieux. Il s'agit d'installer une application directement sur les appareils des utilisateurs. Pour ce type de déploiement, il faut valider la compatibilité de l'application avec les appareils et les systèmes d'exploitation utilisés. En entreprise, des outils spécialisés peuvent être utilisés pour gérer le déploiement et la mise à jour des applications sur tous les appareils.

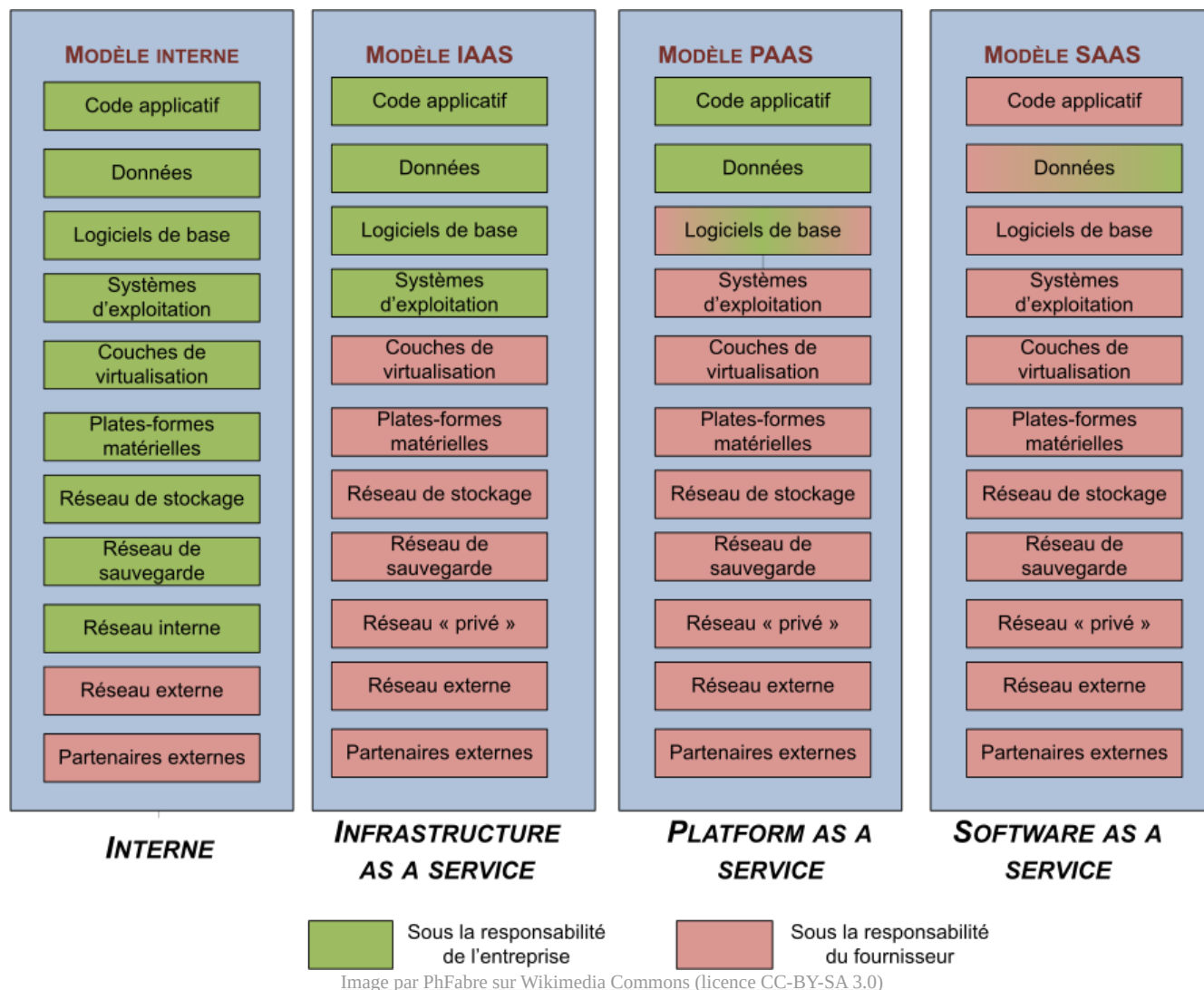
SaaS

Un **SaaS**, pour « *Software as a Service* » (« logiciel en tant que service »), est un logiciel installé sur des serveurs distants maintenus par un fournisseur externe, qui peut être directement l'éditeur du logiciel (ex : Google pour sa suite bureautique en ligne), ou un autre fournisseur. On n'a donc pas à déployer le logiciel soi-même pour l'utiliser. On paie plutôt un abonnement (souvent mensuel) et on accède au logiciel à l'aide d'une application client ou d'un navigateur Web. Par exemple, *Google Suite*, *Discord* et *Kahoot* sont des SaaS.

Les principaux avantages du modèle SaaS sont que l'entreprise cliente n'a pas à gérer les serveurs, la sécurité, les mises à jour, etc. Le coût d'exploitation du logiciel est donc plus prévisible, puisqu'il ne dépend généralement que du nombre de personnes ayant accès au logiciel dans l'entreprise.

Les principaux inconvénients de ce modèle sont une certaine perte de contrôle du logiciel par l'entreprise cliente, de même que le stockage des données de l'entreprise sur les serveurs d'un tiers.

Le modèle SaaS est l'une des trois grandes catégories de services infonuagiques (*cloud*), avec **PaaS** (« *Platform as a Service* ») et **IaaS** (« *Infrastructure as a Service* »).



Auto-hébergement

Dans le cas d'une application auto-hébergée, il faut prévoir l'infrastructure physique et/ou virtuelle nécessaire à son déploiement, soit dans son propre centre de données (« **on-premise** ») ou chez un fournisseur externe (serveur dédié hébergé à l'externe, ou service PaaS ou IaaS). La plupart des applications modernes nécessitent plusieurs composants logiciels pour être mises en œuvre, et ceux-ci doivent parfois être déployés séparément. On peut par exemple avoir à déployer de façon distincte un serveur de bases de données (comme MySQL ou Microsoft SQL Server), un serveur applicatif avec son

environnement d'exécution (propre à l'application, par exemple son code PHP) et un serveur Web (comme Apache ou nginx).

Exemple d'application utilisant plusieurs composants logiciels

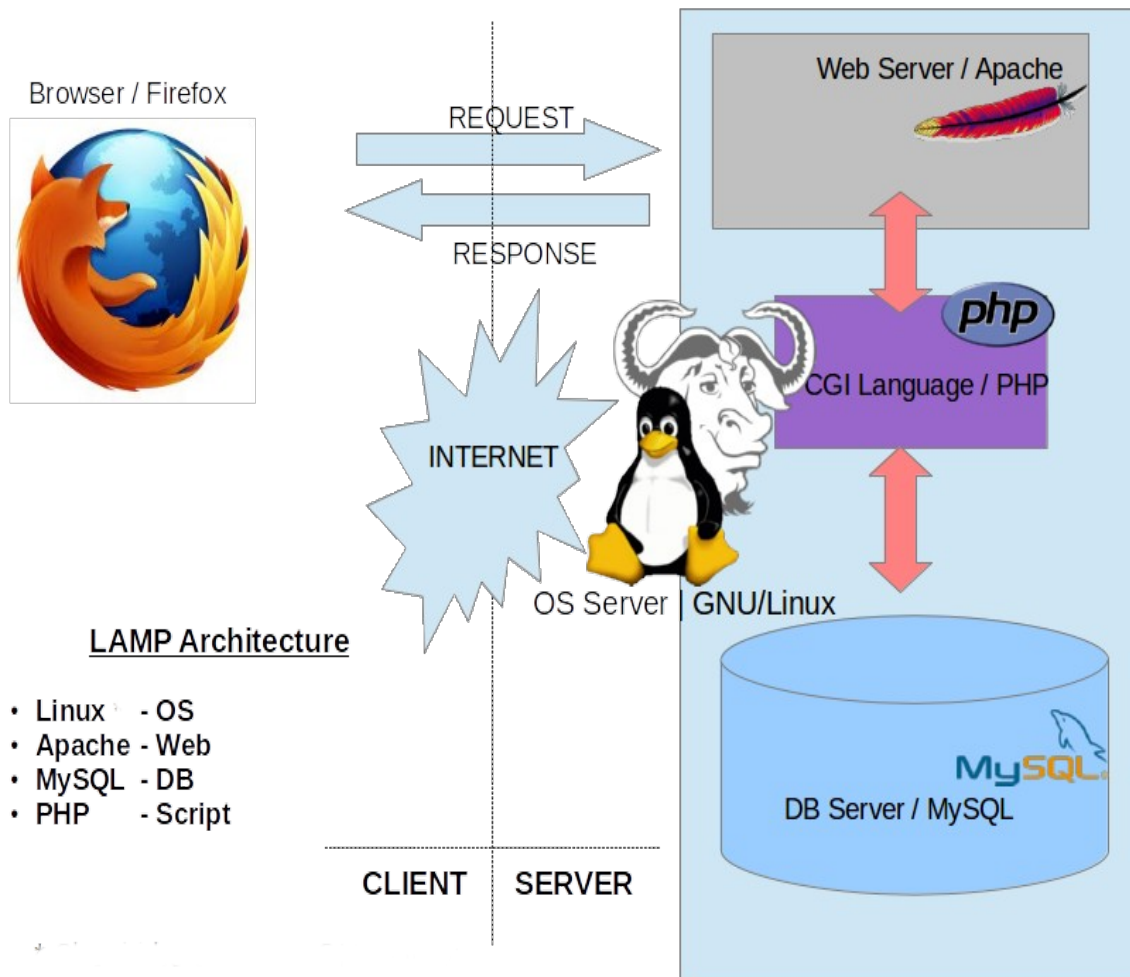


Image par Kesavan Muthuvel (licence CC-BY 3.0)

Les composants logiciels nécessaires à la mise en œuvre d'une application peuvent être installés sur la même machine physique ou virtuelle, ou bien sur des machines distinctes, voire dans des conteneurs (par exemple avec *Docker*¹). Des choix sont donc à faire au niveau de l'architecture du déploiement, des systèmes d'exploitation utilisés (qui doivent bien sûr être compatibles avec les composants), du matériel, des logiciels de virtualisation, etc. Il faut par conséquent s'assurer qu'on a les ressources humaines, matérielles et financières nécessaires pour procéder à ce déploiement, sans oublier qu'il faudra par la suite assurer la maintenance du logiciel et de l'infrastructure. L'aspect de la sécurité informatique n'est par ailleurs pas à négliger lorsqu'on héberge un logiciel soi-même.

1 [https://fr.wikipedia.org/wiki/Docker_\(logiciel\)](https://fr.wikipedia.org/wiki/Docker_(logiciel))

Coût initial et coûts récurrents

Avant de choisir un logiciel, il faut bien entendu se renseigner sur ce qu'il en coûtera à l'entreprise pour le mettre en place et l'utiliser. Dans le cas d'un logiciel à installer localement sur les postes de travail, ou voudra connaître les frais de licence par utilisateur, et s'il s'agit de frais uniques ou récurrents (par exemple, à payer annuellement ou lors de chaque mise à jour). Dans le cas d'un logiciel en auto-hébergement, il faudra aussi prendre en compte les frais de licence, et y ajouter les coûts de déploiement et de maintenance du logiciel. Pour un SaaS, il faudra calculer le coût de l'abonnement selon le nombre d'utilisateurs prévu, de même que l'évolution future de ce coût selon la croissance prévue de l'entreprise.

Types de licences de logiciels

La licence d'un logiciel est un document qui décrit ce qu'on a le droit de faire et ce qu'on n'a pas le droit de faire avec le logiciel. Il existe deux grandes familles de licences de logiciels:

- Licence propriétaire
- Licence libre / *open source*

Une licence propriétaire est une licence qui accompagne un **logiciel propriétaire**, c'est-à-dire un logiciel qui n'est pas *libre* ou *open source*. Commençons donc par définir les notions en lien avec les logiciels libres et *open source*.

Logiciel libre

Contrairement à une certaine croyance populaire, « **logiciel libre** » n'est pas synonyme de « logiciel gratuit », même si les logiciels libres sont souvent gratuits. En anglais, on utilise le terme *free software*, et on précise qu'on utilise le mot *free* comme dans *free speech* (« liberté d'expression ») et non comme dans *free beer* (« bière gratuite »).

Selon la définition de la *Free Software Foundation* (FSF)², un logiciel est libre s'il respecte les quatre libertés suivantes :

- Liberté 0 : La liberté de faire fonctionner le programme comme vous voulez, pour n'importe quel usage.
- Liberté 1 : La liberté d'étudier le fonctionnement du programme, et de le modifier pour qu'il effectue vos tâches informatiques comme vous le souhaitez.
- Liberté 2 : La liberté de redistribuer des copies du logiciel, donc d'aider les autres.
- Liberté 3 : La liberté de distribuer aux autres des copies de vos versions modifiées.

2 <https://www.gnu.org/philosophy/free-sw.html#four-freedoms>

Évidemment, pour que les libertés 1 et 3 soient effectives, on doit avoir accès au code source du logiciel. Un logiciel libre est donc d'abord un logiciel dont le code source est disponible. C'est comme si on vous vendait un gâteau et qu'on vous fournissait la recette en même temps.

Voici quelques exemples de logiciels libres célèbres :

- Linux
- Firefox
- LibreOffice
- VLC media player
- 7-Zip
- La base d'Android
- Chromium
- Apache HTTP Server
- MariaDB

Modèle d'affaires des logiciels libres

Encore une fois, « libre » ne veut pas dire « gratuit ». Il est vrai que les logiciels libres destinés aux particuliers sont souvent gratuits – notons que c'est aussi le cas de nombreux logiciels propriétaires destinés aux particuliers – et le développement de ces exemples spécifiques de logiciels libres est souvent piloté par des bénévoles ou des OSBL. Dans le monde des logiciels destinés entièrement ou partiellement aux entreprises, cependant, il existe bel et bien de nombreux logiciels libres qui sont tout à fait profitables. Tout est une question de modèle d'affaires.

Le modèle le plus typiquement rencontré de nos jours, surtout avec les applications Web, est de permettre l'utilisation du logiciel en autohébergement (*self-hosting*) gratuitement, tout en offrant une option *SaaS* payante permettant à l'entreprise de financer le développement de son logiciel et réaliser un profit. C'est par exemple le modèle d'affaires utilisé par le gestionnaire de mots de passe libre [Bitwarden](#).

Un autre modèle possible est de faire payer aux clients non pas pour l'accès au logiciel lui-même, mais plutôt pour des services connexes, par exemple de soutien technique, de formation, d'intégration, etc. C'est notamment le modèle choisi par l'entreprise *Red Hat* pour sa distribution Linux.

Au moins, avec les logiciels libres, le modèle d'affaires se doit d'être clair. Ce n'est pas toujours le cas avec les logiciels gratuits qui ne sont pas libres: avec de tels logiciels, on dit souvent que si c'est gratuit, c'est que vous êtes le produit, voulant dire que de tels logiciels sont souvent monétisés en utilisant vos données personnelles pour vous afficher des publicités ciblées.

Logiciel *open source*

Selon la définition de l'*Open Source Initiative* (OSI)³, un logiciel est ***open source*** s'il respecte dix conditions qui correspondent grosso modo à la définition d'un logiciel libre. Ainsi, la plupart des logiciels libres sont aussi considérés *open source*, et vice versa. La différence entre les deux mouvements est surtout philosophique. Pour la FSF, l'intérêt des logiciels libres est la liberté, et tous les logiciels propriétaires sont considérés non-éthiques, car ceux-ci « contrôlent » les utilisateurs alors que ce sont les utilisateurs qui devraient contrôler les logiciels. Du côté de l'OSI, on fait la promotion du modèle *open source* pour ses avantages techniques, sans position morale condamnant l'existence de logiciels propriétaires.

De nos jours, la plupart des éditeurs de logiciels utilisent des bibliothèques logicielles qui sont considérés libres et/ou *open source*, et il n'est pas rare de voir de telles entreprises publier des bibliothèques logicielles avec leur code source même si leurs principaux produits sont propriétaires – c'est pour cette raison que le monde de l'entreprise préfère largement utiliser le terme « *Open Source* » plutôt que « Logiciel libre », pour éviter de promouvoir une position morale qui s'oppose à leurs logiciels propriétaires.

FOSS et FLOSS

L'acronyme **FOSS** (pour « *Free and Open Source Software* ») désigne les logiciels libres et *open source*, sans prendre parti pour l'un ou l'autre des mouvements. Par ailleurs, à cause de l'ambiguïté du mot « *free* » en anglais (qui peut vouloir dire « gratuit » ou « libre »), certains préfèrent le terme « ***libre software*** ». On trouve donc parfois l'acronyme **FLOSS**, pour « *Free/Libre and Open Source Software* ».

Licences libres et *open source*

Rappelons que la licence d'un logiciel est un document qui décrit ce qu'on a le droit de faire et ce qu'on n'a pas le droit de faire avec le logiciel. Pour les logiciels libres et *open source*, il existe différentes licences, approuvées par la FSF et/ou l'OSI, qui sont utilisées par de nombreux logiciels. Ces licences se divisent en deux grandes catégories :

- Les licences *copyleft*
- Les licences non-*copyleft* (aussi appelées « licences permissives »)

Une licence est dite ***copyleft*** si celle-ci exige que les versions modifiées du logiciel, si elles sont distribuées, le soient sous les mêmes conditions. En d'autres mots, si on modifie un logiciel libre sous une licence *copyleft* et qu'on publie la version modifiée, on doit publier celle-ci sous la même licence ou sous une licence compatible. C'est le cas, entre autres, du noyau Linux : vous ne pouvez pas le modifier pour en faire un logiciel propriétaire. Vous pouvez cependant le faire avec Chromium, qui est

3 <https://opensource.org/osd>

publié sous une licence non-*copyleft*. Les navigateurs propriétaires Google Chrome, Microsoft Edge et Opera sont d'ailleurs basés sur Chromium.

La licence de type *copyleft* la plus connue est la *GNU General Public Licence*⁴ (GNU GPL, souvent simplement appelée **GPL**). Deux autres licences populaires sont dérivées de la GPL :

- **LGPL** (pour *Lesser GPL*): Conçue pour les bibliothèques. Un programme propriétaire peut utiliser une bibliothèque sous une licence LGPL, mais la librairie elle-même et ses versions modifiées doivent être distribuées librement.
- **AGPL** (pour *Affero GPL*): comme GPL, mais le fait de permettre l'utilisation d'un logiciel à travers un réseau (ex: déploiement sur un serveur Web) est considéré comme une distribution (qui doit donc respecter les termes de la licence).

Quelques licences non-*copyleft* très connues sont les suivantes :

- **BSD**
- **MIT**
- **Apache License**

Le texte d'une licence non-*copyleft* est parfois très court. Par exemple, voici le texte intégral de la licence MIT :

Copyright © 2021 <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

4 <https://www.gnu.org/licenses/gpl-3.0.en.html>

Licence propriétaire

Alors que les logiciels libres sont généralement publiés sous une licence de logiciel libre ou *open source* utilisée par de nombreux autres logiciels, les **licences propriétaires**, pour leur part, sont propres à chaque logiciel.

Une licence propriétaire est une licence qui va à l'encontre d'au moins une des quatre libertés du logiciel libre. Le code source d'un logiciel propriétaire n'est généralement pas disponible, et la licence du logiciel peut imposer des restrictions sur l'utilisation, la copie et la redistribution de celui-ci, en plus d'interdire de modifier le logiciel, voire même d'étudier son fonctionnement.

Il existe aussi des logiciels de type « **open core** », c'est-à-dire qu'il existe une version *open source* de ces logiciels avec des fonctionnalités limitées et une version propriétaire avec les fonctionnalités complètes. On retrouve aussi des logiciels « **source available** », c'est-à-dire des logiciels dont le code source est disponible, mais qui ne respectent pas tous les autres critères pour en faire des logiciels libres ou *open source*.

Pourquoi se soucier des conditions de licence?

Il est important, lorsque vous choisissez ou utilisez un logiciel, de vous soucier des conditions imposées par sa licence pour vous assurer que vous (ou votre organisation) ne faites rien qui soit en violation de celles-ci. Autrement, vous vous exposez à des poursuites par l'éditeur du logiciel. Avant d'utiliser un logiciel propriétaire en entreprise, vous devez vous assurer que sa licence permet une utilisation commerciale, surtout s'il s'agit d'une version gratuite du logiciel.

Si vous travaillez en tant que développeur ou développeuse de logiciels, vous devez faire attention aux licences des bibliothèques que vous utilisez dans votre code. Une bibliothèque libre sous une licence *copyleft* ne devrait pas être utilisée dans le code d'un logiciel propriétaire. Vous devriez toujours vous renseigner sur les types de licences approuvés par votre entreprise avant d'utiliser une bibliothèque provenant d'un tiers.

Références

- https://fr.wikipedia.org/wiki/Software_as_a_service
- <https://www.gnu.org/philosophy/free-sw.fr.html>
- <https://opensource.org/osd>