

Des routes et des ponts (3), de quoi est fait un logiciel

*Après l'introduction du livre Des routes et des ponts de **Nadia Eghbal** (si vous avez raté le début...) que le groupe Framalang vous traduit au fil des semaines, voici un aperçu tout simple des composants de base d'un logiciel.*

Nos lecteurs les plus au courant n'y trouveront rien qu'ils ne sachent déjà, mais l'intérêt de cette présentation c'est justement qu'elle rend abordables et compréhensibles au grand public des objets techniques qui peuvent s'avérer très complexes à comprendre (...et maîtriser !). On peut dire que la démarche choisie ici, pragmatique et imagée, ne manque pas de pédagogie.

Merci à nos lecteurs soucieux des valeurs du libre de noter que l'auteur n'introduit les notions que progressivement : c'est seulement dans quelques chapitres qu'elle établira clairement une distinction claire qui nous est chère.

Vous souhaitez participer à la traduction hebdomadaire ? Rejoignez Framalang ou rendez-vous sur un pad dont l'adresse sera donnée sur Framasphère chaque mardi à 19h... mais si vous passez après vous êtes les bienvenu.e.s aussi !

De quoi sont faits les logiciels

par **Nadia Eghbal**

Traduction framalang : Luc, woof, Diane, xi, serici, Lumibd, goofy, alien spoon, flo, salade, AFS, lyn., anthony

Tous les sites web ou les applications mobiles que nous utilisons, même les plus simples, sont constitués de multiples composants plus petits, tout comme un immeuble est fait de briques et de ciment.

Imaginez par exemple que vous désiriez poster une photo sur Facebook. Vous ouvrez votre appli mobile Facebook, ce qui déclenche le logiciel de Facebook pour vous afficher votre fil d'actualités.

Vous téléchargez une photo depuis votre téléphone, ajoutez un commentaire, puis

vous cliquez sur « Envoyer ». Une autre partie du logiciel de Facebook, en charge du stockage des données, se souvient de votre identité et poste la photo sur votre profil. Finalement, une troisième partie de ce logiciel prend le message que vous avez saisi sur votre téléphone et le montre à tous vos amis à travers le monde.

Bien que ces opérations aient lieu sur Facebook, en réalité, ce n'est pas Facebook qui a développé toutes les briques nécessaires pour vous permettre de publier sur son application. Ses développeurs ont plutôt utilisé du code libre, public, mis à disposition de tous sur Internet par des bénévoles. Facebook ne publie pas la liste des projets qu'ils utilisent, mais une de ses filiales, Instagram, le fait et remercie certains de ces projets sur sa page d'accueil et dans son application mobile

Utiliser du code public est plus efficace pour des entreprises comme Facebook ou Instagram que de développer à nouveau tous les composants par elles-mêmes. Développer un logiciel est comparable à la construction d'un immeuble. Une entreprise du bâtiment ne produit pas ses marteaux et ses perceuses, et n'ira pas non plus chercher du bois pour découper les troncs en planches.

Elle préférera les acheter à un fournisseur et se fournir en bois auprès d'une scierie pour finir le travail plus rapidement.

Grâce aux licences permissives, les sociétés telles que Facebook ou Instagram ne sont pas obligées de payer pour ce code, mais sont libres d'en profiter grassement. Ce n'est pas différent d'une entreprise de transport (Instagram) qui utilise les infrastructures routières publiques (code public) pour acheminer ses produits à des fins commerciales (application Instagram).

Mike Krieger, un des co-fondateurs d'Instagram, a insisté sur ce point en 2013 et encouragé d'autres entrepreneurs à :

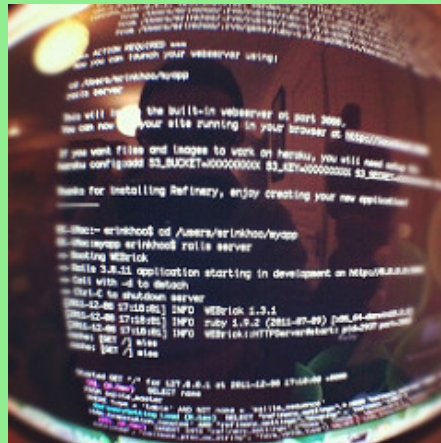
emprunter plutôt que construire à chaque fois que c'est possible. Il existe des centaines d'excellents outils qui peuvent vous faire gagner du temps et vous permettre de véritablement vous concentrer sur le développement de votre produit. (source)

Voici quelques-uns des outils utilisés par une entreprise de logiciels :



Frameworks (environnements de développement)

Les *frameworks* offrent une base, une sorte d'échafaudage, une structure. Imaginez cela comme un schéma pour toute une application. Comme un plan, un *framework* définit la manière dont l'application se comportera sur mobile, ou comment ses données seront sauvegardées dans une base de données. Par exemple Rails et Django sont des *frameworks*.



Ruby on Rails, RefineryCMS
& Heroku, image par Erin
Khoo (CC-BY 2.0)

Langages

Les langages de programmation constituent l'épine dorsale de la communication des logiciels, comme la langue anglaise (NdT : aux USA bien sûr) qu'emploient les ouvriers du bâtiment sur un chantier pour se comprendre. Les langages de programmation permettent aux divers composants du logiciel d'agir et de communiquer entre eux. Si par exemple vous créez un compte sur un site internet et que vous cliquez sur « S'enregistrer », cette application peut utiliser des langages comme le JavaScript ou le Ruby pour sauvegarder vos informations dans une base de données.

Parmi les langages de programmation les plus populaires on peut mentionner le Python, le JavaScript et le C.

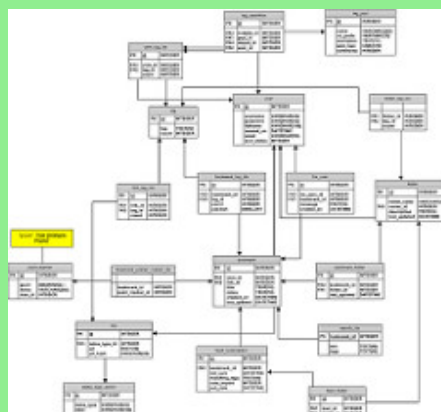
Bibliothèques

Les bibliothèques sont des fonctions pré-fabriqués qui accélèrent l'écriture

du code d'un logiciel, tout comme une entreprise du bâtiment achète des fenêtres préfabriquées au lieu de les assembler à partir des composants de base. Par exemple, au lieu de développer leur propre système d'identification pour les utilisateurs de leur application, les développeurs et développeuses peuvent utiliser une bibliothèque appelée OAuth. Au lieu d'écrire leur propre code pour visualiser des données sur une page web, ils ou elles peuvent utiliser une bibliothèque appelée d3.

Bases de données

Les bases de données stockent des informations (profils d'utilisateurs, adresses électroniques ou codes de cartes bancaires par exemple) qui peuvent être utilisées par l'application. À chaque fois qu'une application a besoin de se souvenir d'une information qui vous concerne, l'application la stocke dans la base de données. Parmi les systèmes de gestion de bases de données (SGBD) les plus populaires on trouve notamment MySQL et PostgreSQL.



Database par gnizr (CC BY 2.0)

Serveurs web et d'applications

Ces serveurs gèrent certaines requêtes envoyées par les utilisateurs sur Internet : on peut les voir comme des centrales téléphoniques qui répartissent les appels. Par exemple, si vous saisissez une URL dans la barre d'adresse de votre navigateur, un serveur web vous répondra en vous envoyant la page concernée. Si vous envoyez un message à un ami sur Facebook, le message sera d'abord envoyé à un serveur d'applications qui déterminera qui vous essayez de contacter puis transmettra votre message

au compte de votre ami.

Parmi les serveurs web très répandus, citons Apache et Nginx.

Certains de ces outils, comme les serveurs et les bases de données, sont coûteux, surtout à l'échelle d'une entreprise, ce qui les rend plus faciles à monétiser. Par exemple, Heroku, une plate-forme sur le *cloud* (sur un serveur distant) qui fournit des solutions d'hébergement et de bases de données, propose une offre de base gratuite, mais il faut payer pour avoir accès à plus de données ou de bande passante. De grands sites reposent sur Heroku comme ceux de Toyota ou de Macy, et l'entreprise a été rachetée en 2010 par Salesforce.com pour 212 millions de dollars.

Il est plus difficile de faire payer d'autres types d'outils de développement, tel que les frameworks, de nombreuses bibliothèques et langages de programmation, qui sont souvent créés et maintenus par des bénévoles.

Ce type d'outils ressemble plus à des *ressources* qu'à des services qui peuvent être activés ou désactivés : les rendre payants limiterait fortement leur adoption. C'est pourquoi n'importe qui, que ce soit une entreprise milliardaire ou un adolescent apprenti développeur, peut utiliser gratuitement ces composants pour créer ses propres logiciels.

Par exemple, selon la page d'accueil d'Instagram, l'une des bibliothèques utilisées par l'entreprise est Appirater. Il s'agit d'une bibliothèque qui facilite les fonctions de rappels automatiques pour l'évaluation d'une application, à destination des utilisateurs d'iPhone. Elle a été créée en 2009 par Arash Payan, un développeur indépendant basé à Los Angeles. Le projet n'apporte aucun revenu à Payan.

C'est comme si des scieries, des centrales à béton et des magasins de matériel donnaient leurs matériaux de base à une entreprise du bâtiment, puis continuaient à approvisionner cette entreprise.