

Aspects techniques

Principalement appliquées aux services web, les architectures distribuées auxquelles sont généralement associé un langage dynamique connaissent de nos jours un grand boom autant chez les développeurs indépendants que dans les sociétés de services et d'ingénierie informatique (SSII). La principale raison de l'essor de ce mode de traitement de l'information réside dans l'augmentation des ressources. En effet, le seul fait de distribuer les traitements sur les ordinateurs d'un réseau augmente les ressources disponibles. En théorie, si le réseau est internet, tous les ordinateurs connectés constituent les ressources potentielles. C'est le sens du slogan de **Sun** : "*The Computer is the Network*". Les projets de calculs répartis tirent parti de cette formidable ressource de processeurs que sont les ordinateurs inactifs connectés à internet.

Un célèbre exemple de calcul distribué est SETI@Home issu du projet ***Search for Extraterrestrial Intelligence***, qui met à contribution tous les ordinateurs volontaires d'internet pour détecter une intelligence extra-terrestre. Les ordinateurs effectuent la même tâche (décryptage de signaux spatiaux), chacun sur un coin de ciel différent. Ce projet a fédéré plus de 5 millions de personnes différentes à travers le monde et a crédibilisé le calcul partagé (à défaut d'avoir découvert d'extraterrestres pour le moment). Les projets comportant des calculs parallélisables sont de bons candidats pour une architecture distribuée (séquençage de motifs d'ADN).

Outre les avantages liés à la simplification des traitements et à l'augmentation des ressources, on note également des apports non négligeables liés à la répartition des données et services. Une architecture distribuée courante est l'architecture trois tiers à la base de la plupart des applications distribuées de commerce électronique. Cette architecture permet d'interroger et de mettre à jour des sources de données réparties. Les services web permettent de faire appel à différents serveurs pour enrichir une prestation (l'achat d'un séjour touristique peut comprendre l'achat d'un billet d'avion, d'un séjour hôtelier et d'une assurance annulation auprès de différents vendeurs par l'intermédiaires de services web), donc d'objets distribués sur le réseau et dialoguant par des messages (**Peer-to-Peer**).

Une des évolutions attendues est le remplacement des achats de logiciels informatiques par des locations de ces mêmes logiciels pour le temps nécessaire à leur utilisation. On peut imaginer par exemple qu'il sera possible à partir d'un logiciel de traitement de texte de faire appel à différents services de corrections orthographiques disponibles sur internet et dont on louera les services selon les besoins. Cette évolution doit être mise en rapport avec le concept de **Web 2.0**.

a. Systèmes distribués

Les aspects techniques des systèmes distribués proviennent des objectifs de leur création.

La gestion de la transparence

La gestion de la transparence consiste à dissimuler la complexité de toutes actions effectuées dans l'implémentation physique des systèmes distribués. Ces actions sont :

- la concurrence d'accès aux ressources, cet accès étant uniforme qu'il soit local ou distant. De plus, la gestion de la concurrence d'accès soulève les problèmes de synchronisation et de transaction.
- la localisation répartie des ressources.
- la migration sans interférer la location physique.
- la relocalisation des ressources.
- la réplication des ressources.
- la tolérance aux pannes des ressources n'interceptant un utilisateur.
- la sauvegarde des objets.
- la possibilité de reconfigurer le système pour en augmenter les performances.
- L'augmentation de la taille du système.

Cependant, la transparence totale n'existe pas dans la réalité. En effet, pourquoi masquer la distribution alors que les utilisateurs sont eux-mêmes répartis. De plus, cacher les pannes de réseau est impossible. Ensuite, cette transparence totale a un coût. En effet,

chaque mise à jour conserve les caches à jour. De plus, à chaque écriture sur le cache mémoire on écrit sur le disque.

La gestion de l'interopérabilité des plateformes (ouverture)

La gestion de l'ouverture concerne :

- Les services offerts selon des règles standards qui décrivent la syntaxe et la sémantique de ces services.
- La mise à jour (dont l'ajout) de composant sans affecter les autres (extensibilité).
- La facilité de configuration et d'utilisation (flexibilité).
- l'ouverture vers des fournisseurs différents.

La mise en échelle (scalabilité)

La scalabilité d'un système est une propriété lui permettant de s'adapter facilement à une augmentation de sa charge. Par exemple, un système reste performant lorsque le nombre d'utilisateurs s'accroît. En effet, cet accroissement de charge peut se faire à différents niveaux, au cours desquels il faut éviter la dégradation des performances, lorsqu'on augmente :

- le nombre des sites et la distance entre eux.
- le nombre d'utilisateurs ou de processus.
- le nombre de domaines de responsabilité.
- le volume des données.
- la fréquence des interactions.

Mais la scalabilité a une autre facette : la gestion de la capacité de la croissance techniquement parlant. En effet, cela consiste à :

- répartir les données et les calculs entre plusieurs machines (déplacement du calcul vers le client, systèmes de nommages et d'informations décentralisés).
- rendre des copies de données disponibles à plusieurs endroits (serveurs de fichiers ou bases de données répliqués, sites web miroirs, systèmes de mémoires partagés).

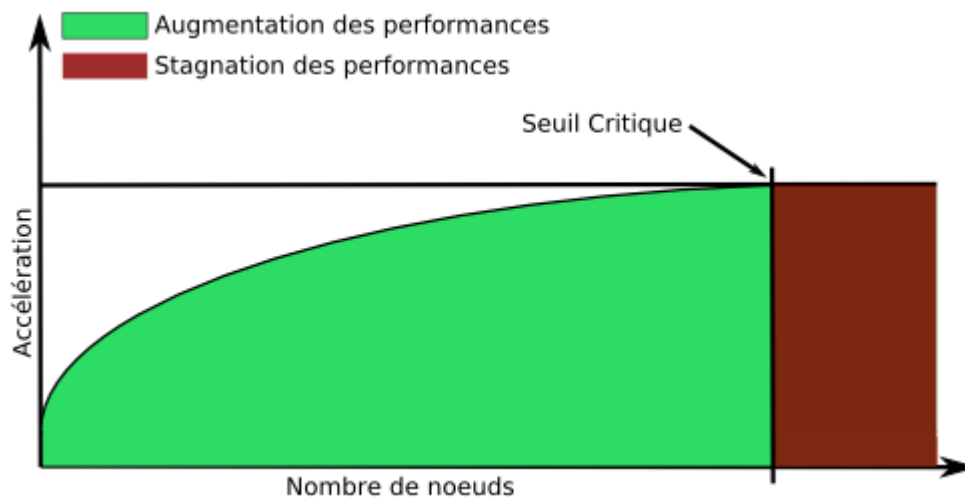
- Permettre aux clients d'accéder à des copies locales (mise en cache web ou des fichiers).

Pour assurer cette scalabilité, il faut mettre en œuvre des techniques de croissance. Ces techniques dérivent de lois qui mesurent et analyses les performances. Décrivons donc maintenant ces mesures et analyses.

- La **mesure des temps d'exécution** : on peut mesurer différents types de temps tels que le temps CPU, appels systèmes, calculs suivant divers outils (une instruction dans le code source) pour détecter d'éventuels problèmes fréquents. De plus, chaque mesure doit être répéter, les séries de mesures doivent être homogènes.
- L'**expression des performances** : elle se traduit par la mesure de l'accélération (Speed Up), de l'efficacité (taux d'utilisation des ressources), par la référence séquentielle (à quel programme et exécution séquentielle se référer suivant la problématique ?).
- Les **sources de perte de performance** : elles sont classées en 3 catégories qui sont les aspects séquentiels, les algorithmes et programmations parallèles, l'environnement de développement.
- Les lois de **Amdhal et Gustafson**. Par exemple, la loi de Amdhal est une fonction qui détermine la gain en rapidité qu'apportera la parallélisations en fonction du nombre de nœud utilisé. Le f représente la fraction en pourcentage de la tâche qui doit être exécuté séquentiellement. Plus cette fraction est petite, plus l'ajout d'un nœud augmentera la rapidité d'exécution.

$$Accélération(N) = \frac{\text{Temps avec 1 processeur}}{\text{Temps avec N processeur}} = \frac{N}{1+(N-1)f}$$

Malheureusement, l'augmentation de la vitesse n'est pas linéaire. Au bout d'un point critique, l'ajout d'un nœud n'augmentera que marginalement la vitesse d'exécution. Après le seuil critique, il n'y a pas d'intérêt à rajouter de nœuds. Ci-dessous un graphique représentant les propos précédents.



La gestion des pannes

La gestion des pannes consiste :

- à la détection, au masquage, à la tolérance et à la récupération des pannes.
- à la redondance des pannes, autrement dit, la reprise sur pannes. C'est une technique de journalisation dans les bases de données.

La gestion de l'hétérogénéité

En effet, l'hétérogénéité doit être gérée à plusieurs niveaux :

- des machines utilisées (puissance, architecture matérielle,...).
- Des systèmes d'exploitation tournant sur ces machines
- des langages de programmation des éléments logiciels formant le système.
- Des réseaux utilisés (impact sur les performances, débit, disponibilité,...).

La gestion de la sécurité

La complexité de la gestion de la sécurité réside dans le fait qu'il a plusieurs points à sécuriser. Elle englobe :

- la confidentialité lors d'une authentification.
- l'intégrité : la protection contre les falsifications et les corruptions doit se faire par l'attribution ou non de droits d'accès. De plus, il faut isoler le système par un pare-feu par exemple.
- la disponibilité des ressources.

Pour cela, il faut une méthode de base qui est la cryptographie.

Les intérêts des systèmes distribués

Le premier avantage qu'on peut citer de cette technologie est le partage et l'utilisation des ressources distantes. Par exemple, un système de fichiers permet de les accéder de n'importe quelle machine appartenant au réseau considéré. C'est aussi le cas de l'imprimante partagée entre toutes les machines. Ensuite, ce système permet d'optimiser l'utilisation des ressources disponibles. Par exemple, les calculs scientifiques sont distribués sur un ensemble de machines. De plus, un système distribué est plus robuste car, étant composé par plusieurs serveurs de fichiers avec sauvegarde par exemple, cela assure la fiabilité.

Les points faibles et les inconvénients

Malgré les différents avantages cités ci-dessus, un système distribué a quand même quelques points faibles et inconvénients. Tout d'abord, si un problème se présente au niveau du réseau (perturbations, problèmes techniques,...), le système peut mal voire ne pas fonctionner. De plus, bien souvent, le serveur est central au système. Ainsi, si ce dernier défaille, tout le système défaille aussi. Enfin, dans le cas où il n'existe pas d'élément central, la gestion du système est aussi décentralisée et cela peut nécessiter la mise en place d'algorithmes plus ou moins complexes.

b. Langages dynamiques

Les Anciens Langages : CGI

Les années des débuts d'internet les sites de documentation étaient essentiellement écrits en HTML simple, juste après peu de temps, l'automatisation de tâches était apparue en utilisant des formulaires en HTML, par la suite les scripts CGI et PERL sont arrivés puis remplacent ce rôle.

CGI (Common Gateway Interface ou Interface de passerelle commune en français), c'est un programme exécuté du côté serveur en affichant des données traitées par le serveur. Un des grands intérêts de l'utilisation de CGI est la possibilité de fournir des pages dynamiques, c'est-à-dire des pages pouvant être différentes selon un choix ou une saisie de l'utilisateur. Dans le cas le plus souvent, avec l'application de CGI (l'utilisation de formulaires HTML), l'utilisateur saisit ou choisit des données, puis clique sur un bouton de soumission du formulaire, alors les données du formulaire sont envoyées en paramètre du programme CGI.

Les langages de programmation les plus utilisés pour l'écriture des CGI sont :

- Langage C, C++
- Java
- PERL

Ces langages ont les caractéristiques en commun :

- Capacité de traiter des chaînes de caractères
- Capacité d'écrire sur le flux standard de sortie
- Capacité de lire le flux de données d'entrée
- Exécutable ou interprétable par le serveur

Il existe une grande partie des scripts CGI qui a été écrite en PERL. PERL est un langage de programmation qui reprend des fonctionnalités du langage C et des langages de

scripts sed, awk et shell, créée en 1986 par Larry Wall qui donne deux interprétations de l'acronyme « PERL » :

- Langage pratique d'extraction et de génération de rapports
- Collectionneur de déchets pathétiquement éclectiques (Documents sur Wikipédia).

PERL est un langage impératif dont la syntaxe est dérivée du langage C et des shells UNIX. Comme en shell, le caractère croisillon(#) introduit un commentaire. La syntaxe de PERL est riche ce qui permet d'exprimer de manière concise des expressions qui, dans d'autres langages, nécessiteraient l'usage d'appel à des fonctions de bibliothèques.

```
my $s = 'toto' ;           # variable scalaire à portée lexical

local $level += 1;        # variable scalaire avec une valeur à
portée dynamique

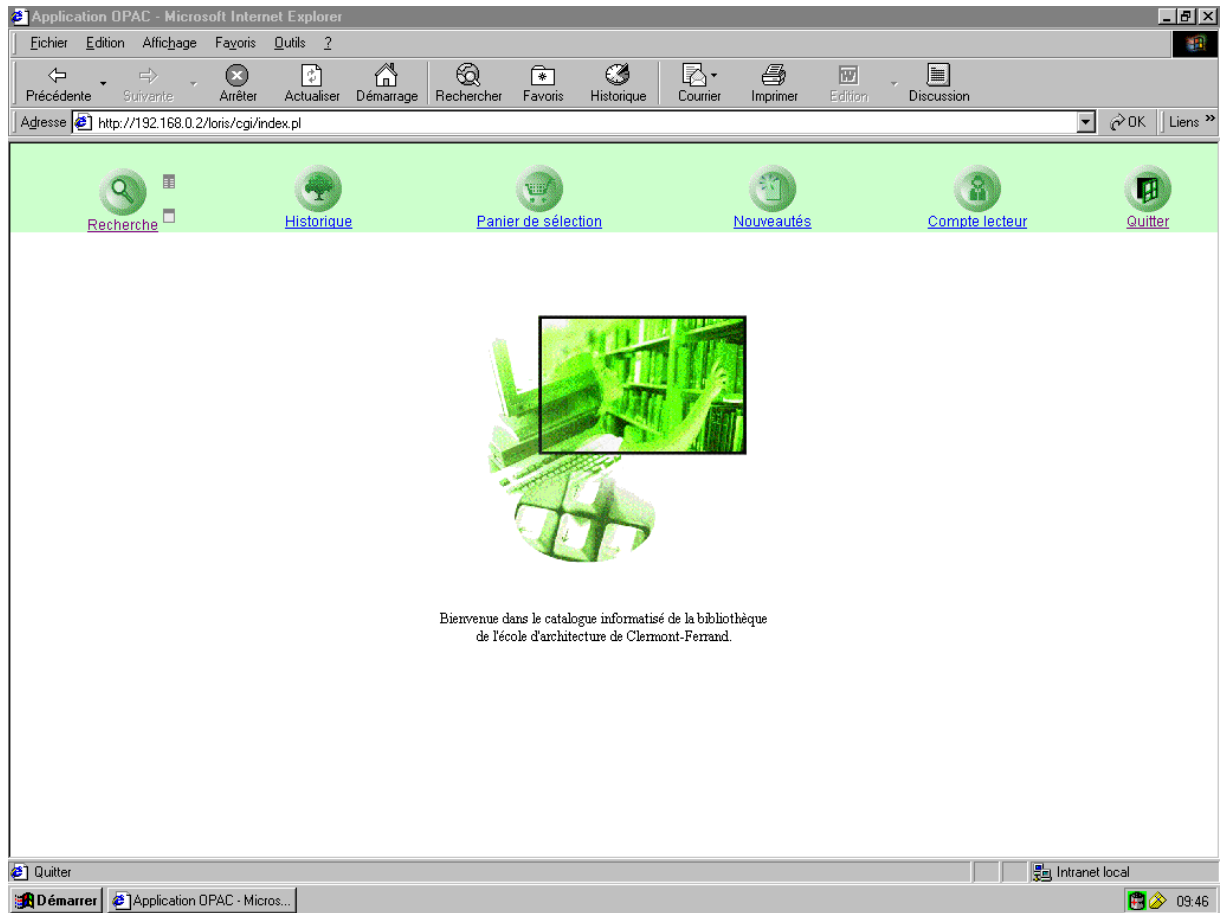
our @s = (1, $s, 3.14);   # variable tableau globale au module
courant

print "la variable toto vaut $toto";

my %a = ( clef1 => [1, 2], clef2 => [ 3, 4] );
```

Un extrait du code en PERL.

Le logiciel DORIS-LORIS est une application de gestion intégrée des bibliothèques et des centres de documentation, spécifiquement conçue dans le respect des règles et des méthodes standardisées du métier des bibliothécaires et documentalistes. Dans l'ancienne version DORIS-LORIS, son application- OPAC web s'écrit en PERL.



Interface de l'OPAC web

Les Langages évolués

La limite de l'utilisation des scripts CGI sont rapidement constaté par les programmeurs et les utilisateurs, car ils dépendent trop des ressources machine- un script est une exécutable, donc il a besoin d'un processus par lancement. Les nouveaux langages de scripts sont arrivés sur la base des autres langages déjà existants (PHP issu du C, ASP issu du VB, JSP issu du JAVA). Ces langages ne consomme pas trop des ressources machine, c'est-à-dire, ils ne lancent pas de processus à chaque exécution, puisque ils sont interprétés directement par un module du serveur Web. Voici les langages utilisés dans ce contexte :

- PHP
- ASP
- JSP

PHP (acronyme récursif pour PHP : HYPERTEXT PREPROCESSOR), est un langage de scripts libre

Principalement utilisé pour produire des pages web dynamiques via un serveur http, mais pouvant également fonctionner comme n'importe quel langage interprété d façon locale, en exécutant les programmes en ligne de commande. PHP est un langage impératif disposant depuis la version 5 de fonctionnalités de modèle objet complètes.

Schéma fonctionnement PHP :



```
<?php
    echo 'Hello, World';
?>
ou
<?php
    print ('Hello, World');
?>
ou encore (obsolète, peu recommandée)
<?='Hello, World' ?>
Résultat affiché :
Hello, World
```

Un extrait du code en PHP.

http://www.nexen.net/chiffres_cles/phpversion/evolution_de_php_sur_internet_decembre_2006.php

nexen.net Nexen.net: Portail PHP/MySQL [Espace membre](#)

A la une Ressources Communauté
 Actualités Articles Conférences [Chiffres clés] English Sécurité

Rechercher...

SFR PRÉSENTE EN EXCLUSIVITÉ JUSQU'AU 22/02/08 :
L'EEPC
 ET LA NOUVELLE CLÉ INTERNET 3G+
 Engagement de 12 mois minimum. Offre soumise à conditions

Chiffres clés

PHP Version

Lettre hebdomadaire
 Nom
 Suggérer par m
 S'inscrire
 435 lettres envoyées
 3076 inscrits

Syndicate
 RSS 0.91
 RSS 1.0
 RSS 2.0
 ATOM 0.3
 OPML SHARE IT!
 FEEDBURNER
 netvibes
 Technorati
 Add to Google

Evolution de PHP sur Internet (décembre 2006)
 Ecrit par Damien Seguy
 mardi 02 janvier 2007

Vous lisez une vieille version : ce document dispose d'une version plus récente : [Evolution de PHP sur Internet \(janvier 2007\)](#)

Voici le récapitulatif des statistiques de diffusion de PHP dans le monde, au mois de décembre 2006. Pour connaître la méthodologie, voyez [la section phpversion](#). 19 millions de serveurs ont été testés durant le mois de décembre 2006, et 9,2 millions ont été retenus pour les statistiques : les domaines sans site web, inaccessibles au robot, les fournisseurs d'accès ou bien encore les parkings à noms de domaines, ont été écartés des statistiques.

- [Evolution de PHP sur Internet](#)
- [Evolution comparées des versions de PHP](#)
- [Evolution des versions 4.x et 5.x de PHP](#)

Evolution de PHP sur Internet

Evolution de la présence de PHP sur Internet

publicité webdistrib
www.webdistrib.com

publicité l'informaticien
 l'informaticien -
 Actualités Informatiques

publicité dvdrama
dvdrama.com - l'actualité du
 DVD

Un exemple d'un site en PHP

ASP (ACTIVE SERVER PAGES)

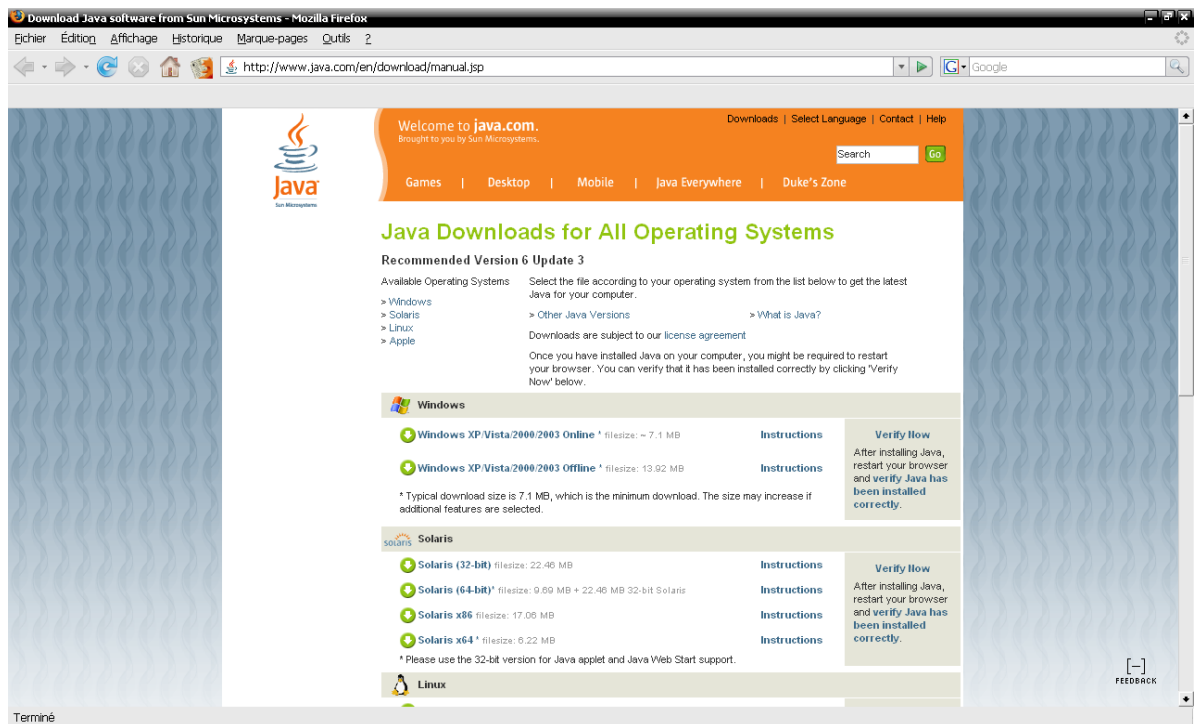
ASP est une technologie développée par Microsoft. C'est une technologie web dynamique, équivalente et concurrente de PHP. Elle a besoin d'une plate-forme Windows avec Internet Information Services installée, ou encore une plate-forme Linux avec une version modifiée d'Apache pour fonctionner. Les pages ASP peuvent aussi être programmées en utilisant de différents langages de programmation (JavaScript, PERL, C++...). Ce langage est très souvent utilisé pour les sites d'E-Commerce.



Un exemple du site en ASP (cddiscount.com).

JSP (Java Server Pages)

Le JSP est une technologie basée sur JAVA qui permet aux développeurs de générer dynamiquement du code HTML, XML ou tout autre type de page web. La syntaxe du JSP ajoute des balises XML, appelées *actions JSP*, qui peuvent être utilisées pour appeler des fonctions. De plus, la technologie permet la création de bibliothèques de balises JSP (*taglib*) qui agit comme des extensions HTML ou XML. Les bibliothèques de balises offrent une méthode indépendante de la plate-forme pour étendre les fonctionnalités d'un serveur http.



Un exemple du site en JSP.

Les langages (les plateformes) puissants et récents

Ruby et Ruby en Rail

Ruby est un langage orienté objet (tout n'est qu'un objet).

```
class Avion
  @@avion_fabriques = 0
  attr_reader :moteurs
  attr_accessor :altitude

  def initialize(moteurs)
    @moteurs = moteurs
    @@avion_fabriques += 1
  end

  def self.quantite
    @@avion_fabriques
  end
end

zingue = Avion.new(2)
zingue.moteurs
zingue.altitude = 33000
Avion.quantite
```

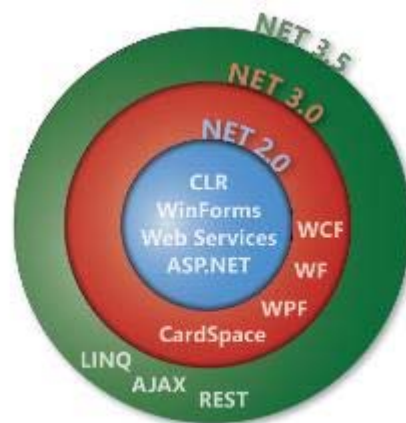
Un extrait du code en Ruby.

Le site web de l'entreprise utilise Ruby : « Artprice », l'entreprise Lyonnaise- Leader Mondial de l'information sur Le marché de L'art (<http://web.artprice.com/start.aspx?l=fr>).

Ruby en Rails est un Framework web libre écrit en Ruby, il suit le motif de conception Modèle-Vue-Contrôleur.

.NET

Le .NET (Framework .NET) est une plateforme réalisée par Microsoft. Il supporte plus de 27 langages de programmation. Les applications dans .NET ne s'exécutent plus directement en code machine natif, mais grâce à une sorte de machine virtuelle, la CLR (Common Language Runtime). La technologie XML est beaucoup utilisée dans .NET.



Versions cumulées du .NET Framework

Avec .NET, les composants (produits et langages) de l'architecture Microsoft sont proposés :

- Un nouveau langage nommé C# « see sharp » : il s'agit d'un langage objet moderne dont la synthèse est entre C++ et Java.
- La technologie ASP.NET.
- Le langage VB.NET.
-

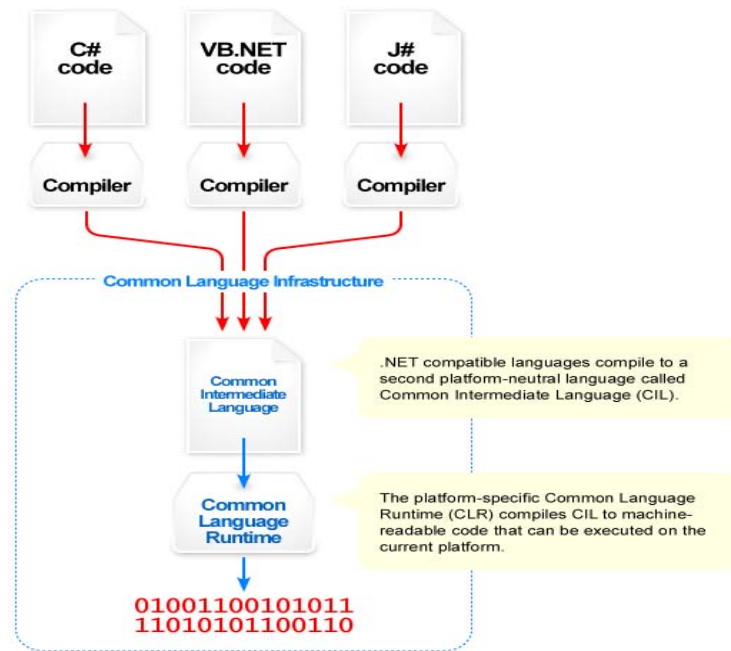


Schéma explicatif du Common Language Infrastructure

Java EE

Java EE (Java Enterprise Edition, ancien-J2EE) est un ensemble de spécifications coordonnées et pratiques qui permettent ensemble des solutions pour le développement, le déploiement, et de la gestion des applications multi-tiers centralisées sur un serveur. Java EE est basé sur J2SE (Java 2 Standard Edition) qui contient les API de bases de java. Elle se divise en 2 parties principales :

- Un ensemble de spécifications pour une infrastructure dans laquelle s'exécutent les composants écrits en Java (nommé serveur d'application).
- Un ensemble d'API (Interfaces de programmations, par exemple : Servlet, Portlet, JSP, JSF...) qui peuvent être obtenus et utilisés séparément.

Ces API se regroupent en trois catégories :

- Composants : Servlet, JSP, EJB
- Services : JDBC, JTA/JTS, JNDI, JCA, JAAS

- Communication : RMI-IIOP, JMS, Java Mail.