

## Eléments de base d'architecture logicielle

Ce document présente des notions générales concernant l'architecture logicielle dans son ensemble, qui englobe la problématique du système d'exploitation.

### La notion de couche

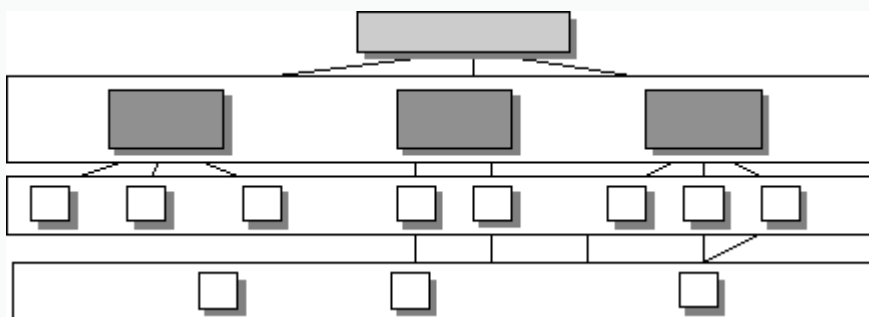
En informatique technique, on parle souvent de "couche logicielle". Qu'est-ce que ça veut dire exactement ?

Une application aussi complexe soit-elle, est toujours le résultat d'un déroulement d'instructions. Malgré tous les efforts que l'industrie logicielle fait pour s'échapper de cette réalité, il existe néanmoins un "point de vue" stable qui est celui du "code exécutable". Ceci reste vrai, même derrière l'interface graphique la plus sophistiquée.

Les actions voulues par l'utilisateur sont des "cas d'usage", comme la méthodologie d'analyse les qualifie. Ces cas d'usage, au départ définissable comme des intentions "humaines" d'action de l'utilisateur, pénètrent à un moment donné le monde de la machine, ne serait-ce que par l'action d'un élément d'interface, l'envoi d'une commande ou tout autre "signal" d'activation.

Tout ce qui va suivre dans la machine est le déclenchement d'une certaine séquence de code.

Ce code s'appuie sur des actions plus fines, des primitives reproductibles. Un jeu de fonctions que l'on a isolées pour pouvoir les appeler souvent, et masquer la complexité de l'algorithme qu'elles représentent. Ces fonctions sont en général proposées dans un "plan sémantique cohérent", (par exemple, les fonctions de manipulation du système de fichiers ont à peu près toutes la même "puissance" ou "angle de vue"). Ces fonctions appellent elles-même des fonctions plus "fines", plus "élémentaires".

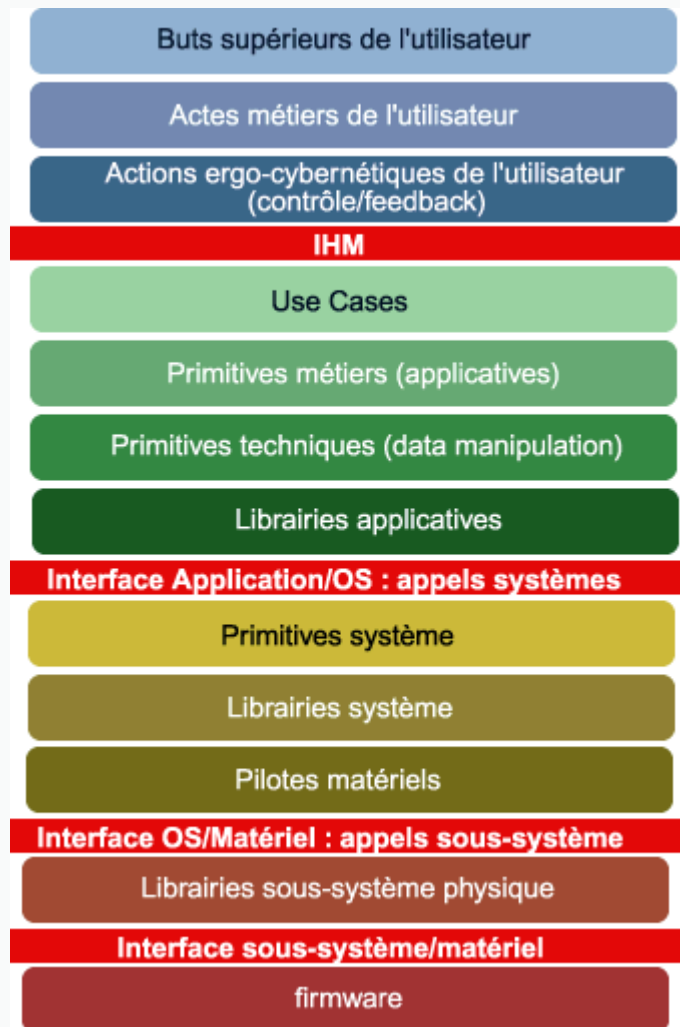


*Division par couches logicielles*

Un "niveau sémantique" qui définit un jeu de primitives est une couche logicielle. Cette couche se définit en général par quelque chose qui qualifie cette "granularité" ou "puissance" d'action.

Ainsi la couche des "cas d'usage utilisateur" n'a pas la même valeur sémantique que la couche technique de manipulation des secteurs sur un disque dur.

### Les principales couches logicielles à l'échelle d'une machine



*Couches logicielles traversées par un utilisateur*

### **Comment se matérialise une couche ?**

En général, en programmation procédurale impérative (C, comme meilleur exemple), une couche se matérialise par un ensemble de librairies qui permettent de manoeuvrer un dispositif matériel ou logiciel, à un certain niveau d'angle de vue.

En programmation objet, la notion de couche est plus difficile à percevoir. En effet, à l'échelle de la machine toute entière, les langages objet ne sont souvent utilisés que dans les parties "hautes" ou "applicatives" (couches vertes). A l'échelle d'une application, on peut bien sûr définir plus finement des couches par la même technique, et qui concerneront à ce moment un univers programmé entièrement objet.

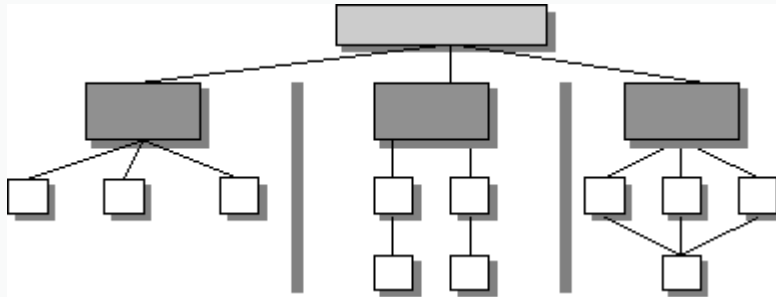
Dans ce contexte, il s'agira principalement de certains groupes d'objet qui participent à la couche logicielle. Il existera donc des objets de "haut niveau" et des objets de "bas niveau". En général, par la technique de composition, les objets de "haut niveau" (ou de "hautes couches") engloberont des objets de bas niveau (ou de couches basses) dans leurs entrailles.

### **La notion de composant**

Les couches définissent une division verticale de la problématique logicielle. Elles sont liées à des "niveaux sémantiques", à une certaine hauteur de vue sur la réalité matérielle

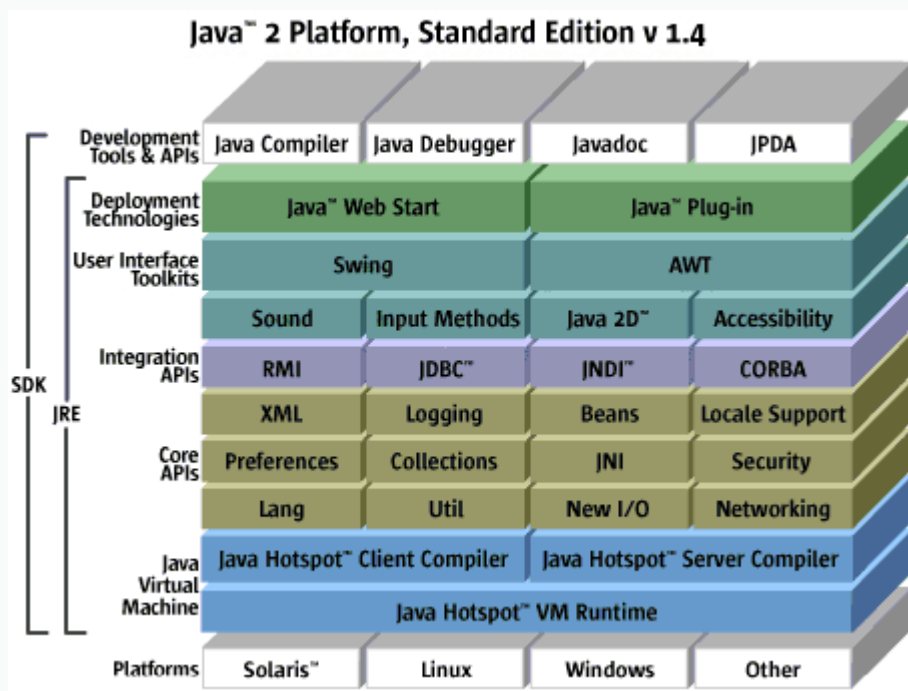
de l'information stockée dans la machine. La notion de couche est indépendante de la notion de fonctionnalité.

La notion de composant est par contre une idée de découpage vertical, où le système d'information est découpé en tranches, chaque tranche étant responsable de la fourniture d'un nombre précis de fonctionnalités.



*Division par modules ou "composants"*

L'API Java démontre bien cette double division :



*Plan de masse de la technologie Java*

Le plan ci-dessus montre deux choses : d'une part le découpage en "couches" à l'intérieur même de la technologie Java, elle-même contribuant à construire en général une couche applicative de la problématique générale. Le deuxième constat est celui de la division d'une couche en différents composants (chaque bloc peut être considéré comme un "composant-librairie" qui règle un certain nombre de problèmes spécifiques à un besoin particulier.