Introduction au développement pour le web

Plan

1. **Introduction**
2. **Les environnement de développement intégrés (IDE)**
3. **Les langages du web**
4. **Serveurs d'applications**
5. **Internet et le Web**
6. **Evolution des architectures client-serveur**
7. J2EE
8. Objets répartis et architecture client-serveur
9. Systèmes transactionnels
10. Exigence d'un projet informatique
11. Constat

1. Introduction

La programmation structurée est basée sur une décomposition en actions.

La programmation orientée objet travaille avec une décomposition en objet.

Les instructions élémentaires sont les mêmes, mais leur regroupement est différent.

On trouve dans l'approche objet trois principes fondamentaux :

- l'encapsulation : un objet regroupe à la fois ses attributs et ses opérations associées,
- l'indépendance temporelle : le comportement d'un objet est indépendant du contexte dans lequel il est appelé,
- l'indépendance spatiale : les informations relatives à une même entité sont physiquement dans le même module.

1. Introduction

Traduction en Java

- on utilisera des classes et des interfaces.
- une classe modélise une entité à l'aide d'attributs et de méthodes.
- un programme construit une instance de classes qui est alors appelée objet
- les classes peuvent être rangées dans des paquetages

Exemple de prgm

Sous linux, on écrit avec un éditeur (gedit ou emacs) le fichier MaClasse.java.

(attention : Java distingue majuscules et minuscules)

```
public class MaClass {
    public static void main (String[] args){
        System.out.println("Affichez ce que vous voulez");
    }
}//Ceci est un commentaire
```

Exemple de prgm

- class mot réservé qui indique que l'on commence la description d'une classe, ce mot est nécessairement suivi du nom de la classe : c'est l'entête de la classe (ligne 1)
- { marque le début d'un bloc ; en ligne 2 c'est le début du corps de la classe MaClass
- une classe contient des attributs et des fonctions ou des méthodes ; ici il n'y a qu'une méthode nommée main (ligne 2)
- public static sont des modificateurs
- void indique que la méthode main ne renvoie aucune valeur

Exemple de prgm

- l'exécution d'un programme en Java se fait toujours dans une méthode principale qui se nomme toujours main ; l'entête de cette méthode est toujours comme dans la ligne 2, on ne peut changer que l'identificateur arg
- String est une classe du paquetage java.lang ;
- String [ ] arg est le paramètre de la méthode main, et c'est un tableau de chaînes de caractères
- System.out.println("affichez ce que vous voulez") ; est une instruction qui écrit à l'écran affichez ce que vous voulez puis passe à la ligne ; println est une méthode de l'objet out qui appartient à la classe System, cet objet sert à écrire dans le fenêtre d'exécution
- toute instruction se termine par un point-virgule

Compilation

Pour compiler ce fichier, dans une fenêtre de commande, on exécute la commande

javac MaClass.java

On peut alors voir dans le répertoire courant un fichier MaClasse.class, ce fichier contient le bytecode de la classe MaClass.

Ce code est indépendant de la machine (et de son système d'exploitation).

Puis on fait appel à la machine virtuelle Java qui interprète le bytecode au fur et à mesure de l'exécution du programme : on utilise la commande

java MaClass

qui exécute la méthode main contenue dans la classe MaClass

Compilation

On prendra l'habitude suivante :

le nom du fichier sera MaClasse.java pour la déclaration

class MaClass { ....

D'autre part, on a tout intérêt à séparer les fichiers source des fichiers bytecode .class.

Pour cela on créera deux répertoires jumeaux Source et Class ; on écrira les fichiers source .java dans Source.

Toujours dans le répertoire Source, on compilera par la commande

javac -d ../Class/ MaClass.java

puis toujours dans le répertoire Source, on exécutera par la commande

java -cp ../Class/ MaClass

Librairies

Java fournit de nombreuses librairies de classes remplissant des fonctionnalités très diverses : c'est l'**API** Java (Application and Programming Interface /Interface pour la programmation d'applications).

Ces classes sont regroupées par catégories en paquetages (ou ≪packages≫).

Les principaux paquetages :

- java.util : structures de données classiques
- java.io : entrées / sorties
- java.lang : chaînes de caractères, interaction avec l'OS, threads
- java.awt : interfaces graphiques, images et dessins
- java.applet : les applets sur le web
- javax.swing : package proposant des composants légers pour la création graphiques

JDK

L'environnement de développement fourni par Sun est le **JDK** (Java Development Kit / Kit de développement Java). Il contient :

- les classes de base de l'API java (plusieurs centaines),
- la documentation au format HTML (dans le répertoire où est installé le JDK - /jdk1.. . ./docs/api/index.html - ou bien à l'adresse suivante http ://java.sun.com/docs/index.html)
- le compilateur : javac
- la JVM (machine virtuelle) : java
- le visualiseur d'applets : appletviewer
- le générateur de documentation : javadoc

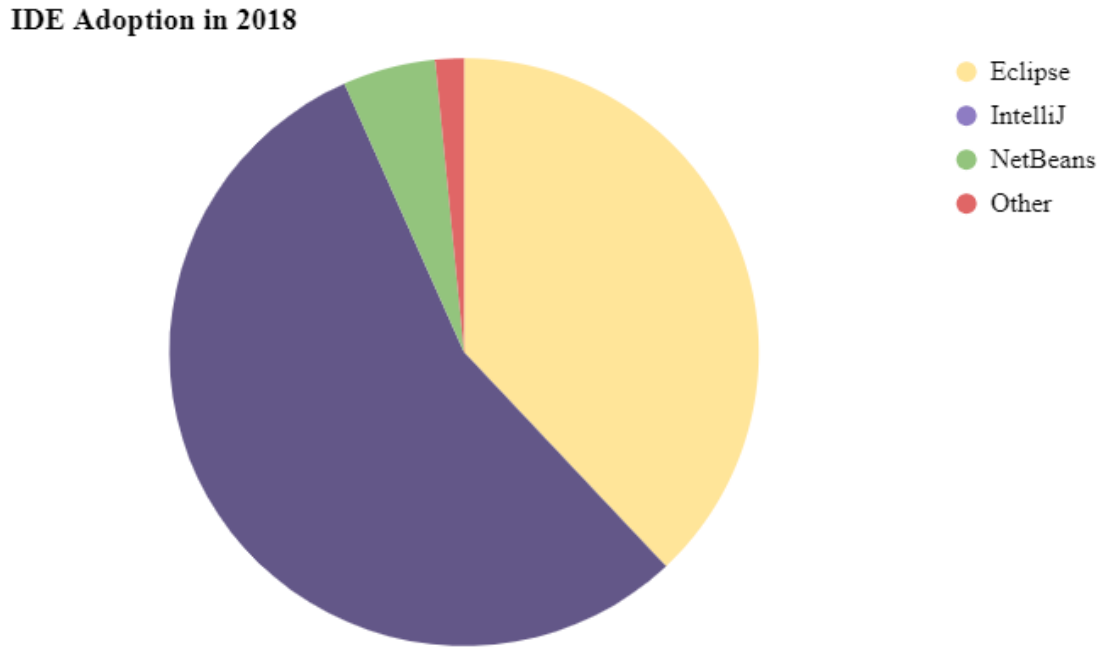
2. Environnements intégrés

Il existe de nombreux IDE (Integrated Development Environment) parmi lesquels

- Eclipse
- **Netbeans**
- IntelliJ
- ...

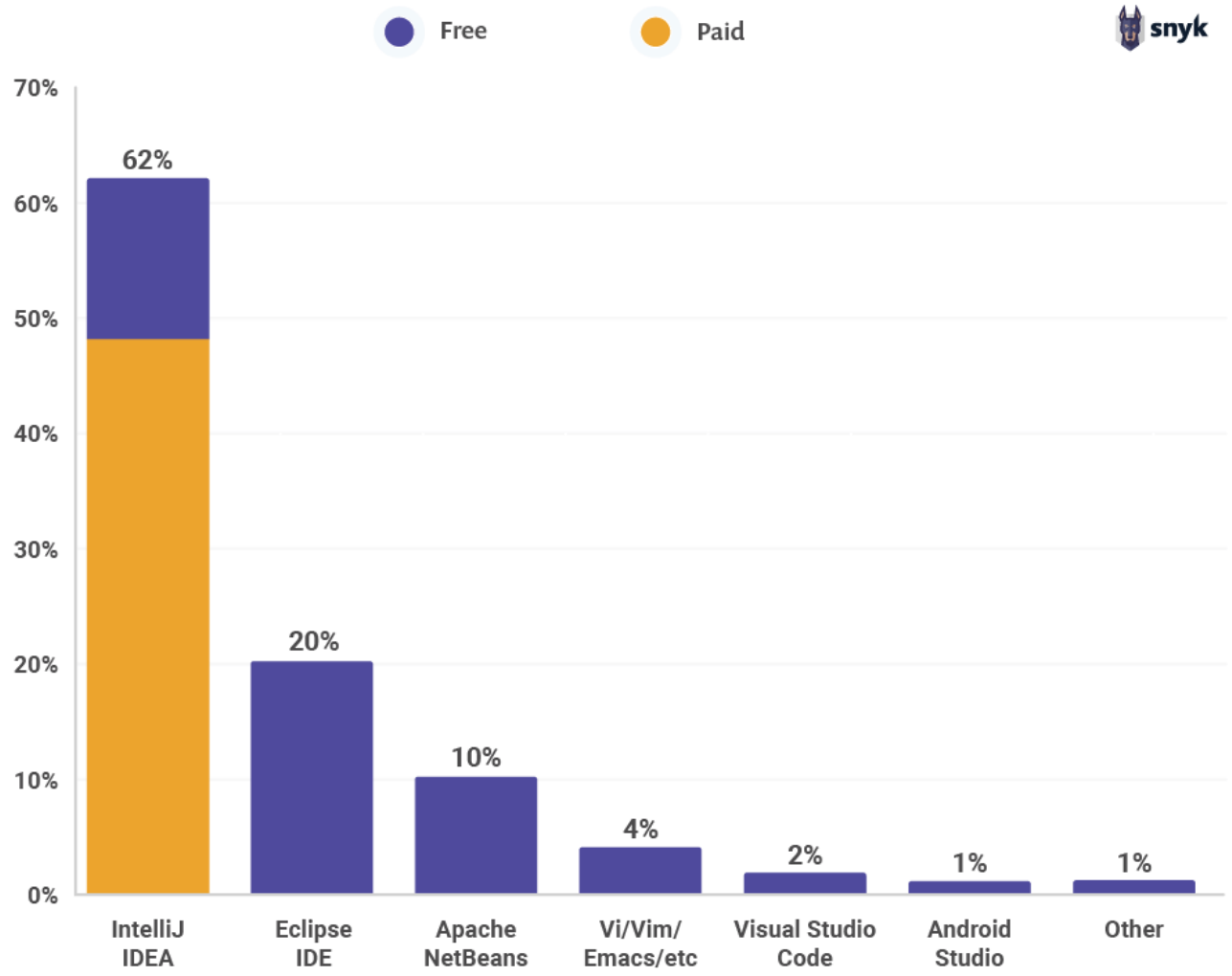
Bien sûr il faut apprendre à s'en servir car ils offrent de nombreuses possibilités.

2. Environnements intégrés

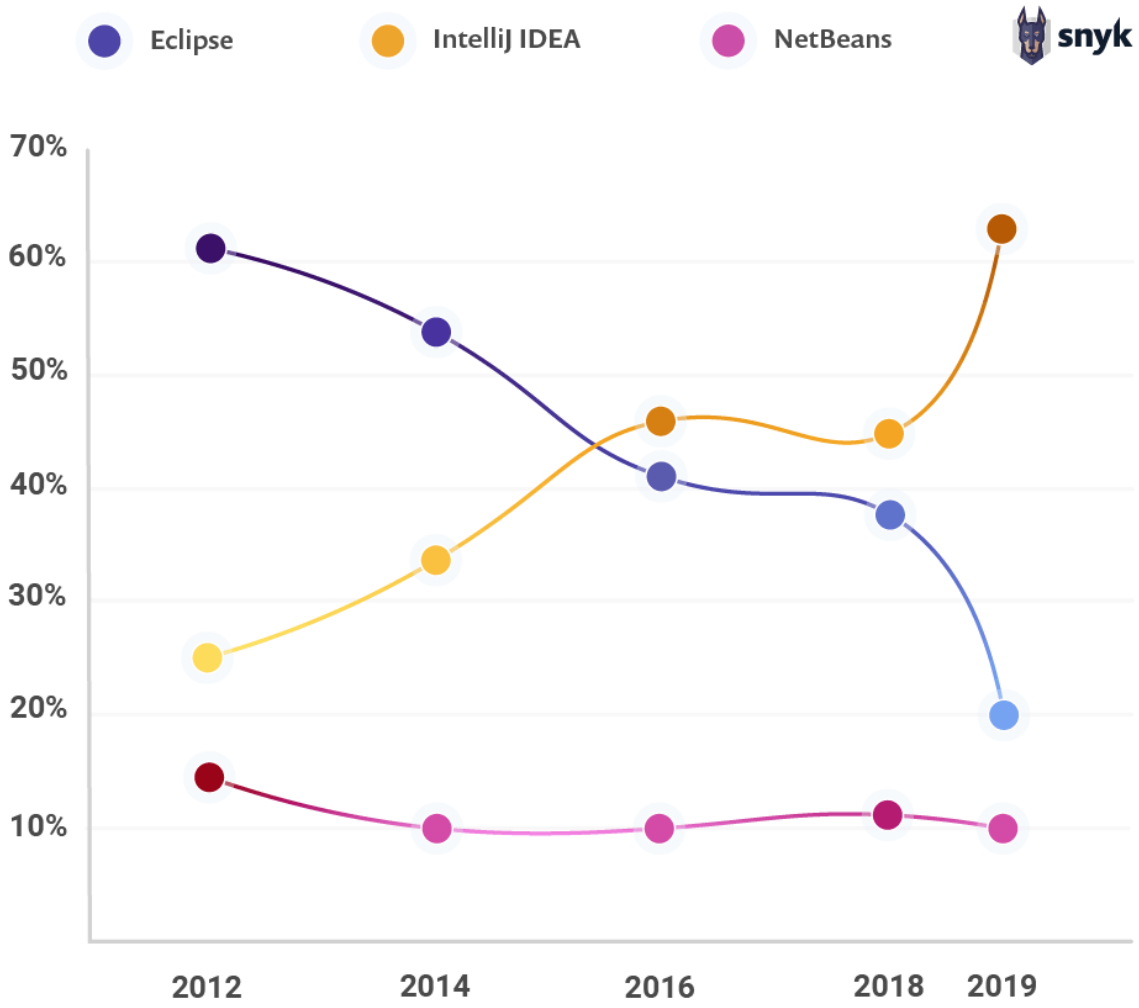


- IntelliJ est passé de 45,8% en 2017 à 55,4% aujourd'hui, remportant clairement la bataille IDE dans l'espace Java cette année.
- NetBeans, est tombé à 5,1% cette année, soit moins de la moitié des 12,4% de 2017.
- Et Eclipse semble avoir quelque peu stoppé le saignement et n'a chuté que de 2% au cours de cette année, pour atteindre 38% du marché.

2. Environnements intégrés



2. Environnements intégrés



3. Les langages du web

Coté client:

- Pour le Web et dans les architectures N tiers, le client ne fait que recevoir des pages web, les afficher à l'utilisateur et transmettre ses actions au serveur.
- Les langages utilisés pour mettre en forme les données et les afficher à l'utilisateur sont le HTML, le CSS et éventuellement le Javascript → ils sont tous interprétés par le navigateur, directement sur la machine client. D'ailleurs, le client est uniquement capable de comprendre ces quelques langages, rien de plus !

3. Les langages du web

Coté serveur:

- le serveur aussi dispose de technologies bien à lui, que lui seul est capable de comprendre, ayant pour objectif final de générer les pages web à envoyer au client, avec tous les traitements que cela peut impliquer au passage
  - analyse des données reçues via HTTP,
  - transformation des données,
  - enregistrement des données dans une base de données ou des fichiers,
  - intégration des données dans le design…
- il existe plusieurs technologies capables de traiter les informations sur le serveur. Java EE est l'une d'entre elles, mais il en existe d'autres : PHP, .NET, Django et Ruby on Rails … Toutes offrent sensiblement les mêmes possibilités, mais toutes utilisent un langage et un environnement bien à elles !

Meilleurs langages de programmation pour le développement Web en 2020

1. Python
2. Java
3. Javascript
4. PHP
5. Go
6. Ruby
7. C
8. Swift
9. Rust
10. Kotlin

Python

Python est devenu l'un des langages de programmation les plus populaires actuellement et il ne montre aucun signe de disparition. Ce langage est également connu comme le meilleur langage pour la création d'applications web basées sur l'IA et l'apprentissage machine.

De plus, l'essor de la science des données a amélioré le développement de Python en tant que langage de programmation. Python bat maintenant JavaScript en tant que langage d'enseignement dans les instituts.

Sites web utilisant Python : Facebook, Microsoft, Dropbox, Mozilla, Netflix, Youtube et d'autres projets Google utilisent partiellement Python.

Python

Les avantages de Python :

- Facile à utiliser et agréable à apprendre
- Prise en charge de plusieurs plates-formes et systèmes
- Permet un développement rapide en utilisant moins de code
- L'open source avec une vaste communauté
- A toutes les bibliothèques que vous pouvez imaginer
- Permet d'adapter facilement les demandes les plus complexes

Les inconvénients de Python :

- Python n'est pas natif de l'environnement mobile
- On ne peut pas utiliser Python pour construire un jeu en 3D à haute résolution graphique
- Python n'est pas recommandé pour les tâches gourmandes en mémoire.
- Python n'est pas une bonne option pour les travaux multiprocesseurs/multi-cœurs

Java

Java est considéré comme le langage le plus stable et il a survécu à l'apogée de l'industrie de la programmation depuis 20 ans. Qu'est-ce qui fait le succès de Java ? C'est d'écrire une fois et de fonctionner partout, grâce à sa polyvalence et son ubiquité. En outre, Java a une bonne réputation pour sa grande compatibilité entre les plates-formes. La machine virtuelle Java (JVM) lui permet de fonctionner sur une grande variété de dispositifs et de plateformes. La plupart des entreprises ont construit leur application back-end en utilisant Java.

Sites web utilisant Java : ebay.com, linkedin.com, aws.amazon.com, aliexpress.com, bitbucket.org, ebay.co.uk

Java

Les avantages de Java :

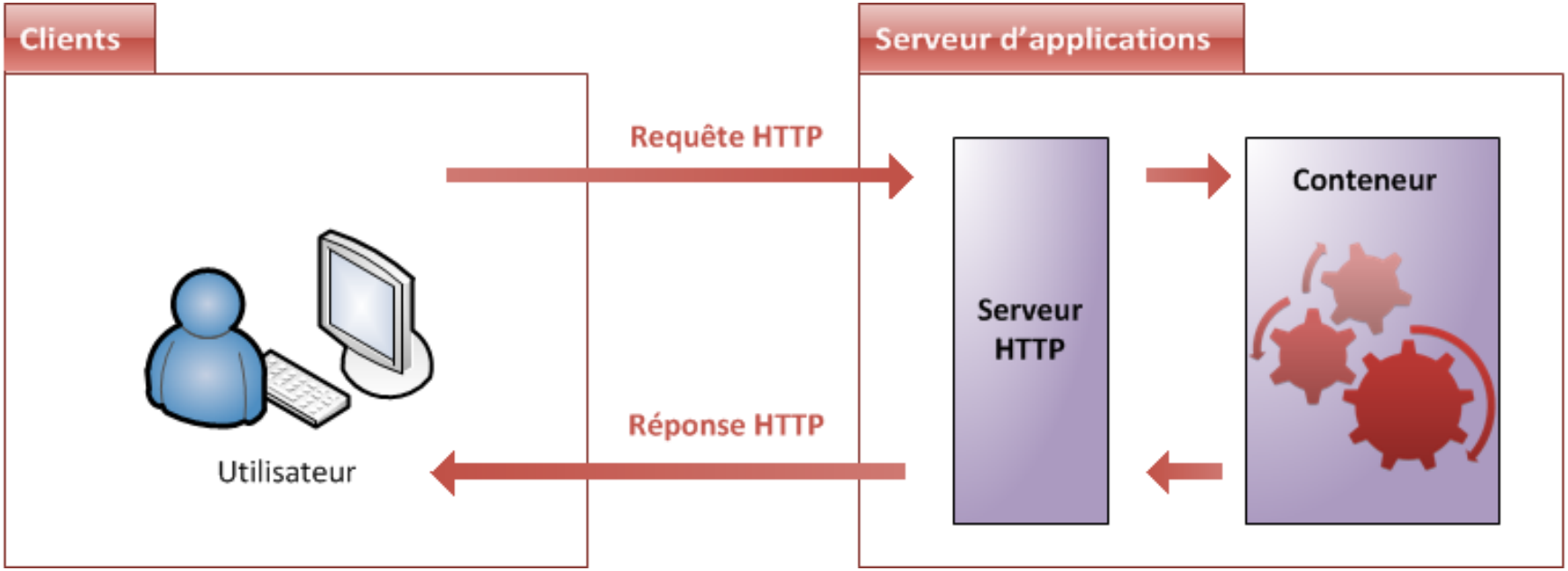
- Un bon début d'études pour penser comme un développeur
- Un langage de haut niveau avec une courbe d'apprentissage douce et une syntaxe simple
- Stabilité et grande communauté
- Norme pour l'informatique d'entreprise
- Multithreading
- Gestion automatique de la mémoire
- Indépendance vis-à-vis des plateformes (écrire une fois, exécuter partout)
- Pénurie de risques en matière de sécurité

Les inconvénients de Java :

- Beaucoup de nouveau vocabulaire à apprendre
- Mauvaise performance
- Un code complexe

4. Serveurs d'applications

Le composant, qui va se charger d'exécuter le code du client en plus de faire le travail du serveur HTTP, se nomme le serveur d'applications. Un tel serveur inclut un serveur HTTP, et y ajoute la gestion d'objets de diverses natures au travers d'un conteneur,



4. Serveurs d'applications

Solutions commerciales comme:

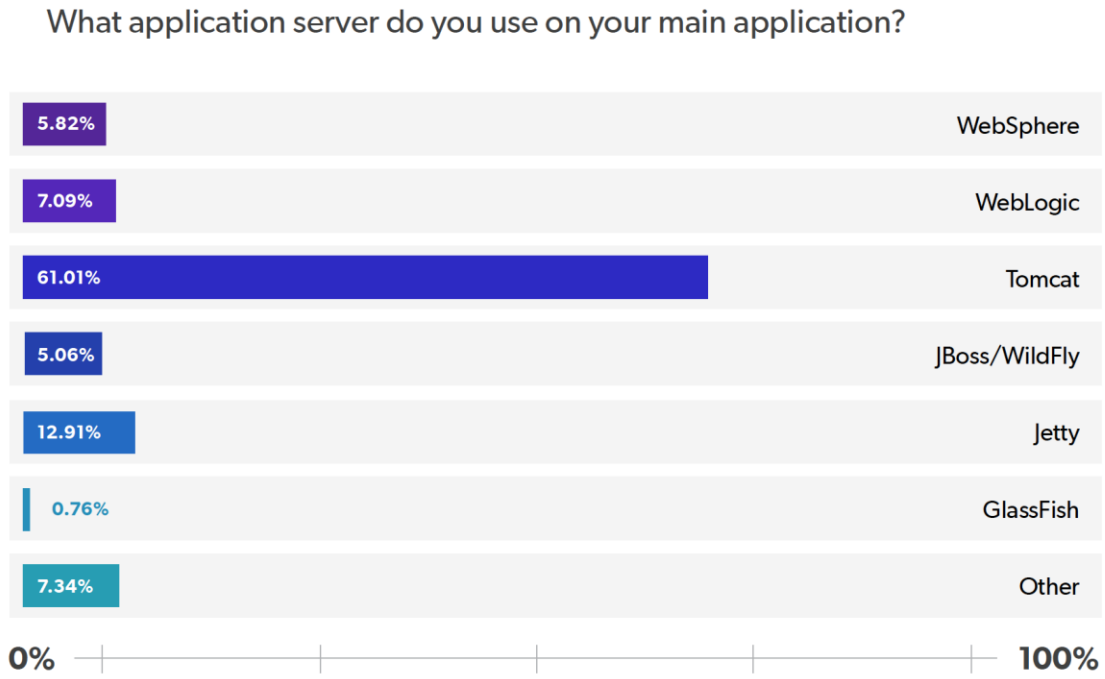
- Weblogic server
- WebSphere Application Server
- VisualAge for Java
- iPortal Application Server
- iPlanet Application Server
- Oracle9iAS
- eXtend Application Server
- EAServer

4. Serveurs d'applications

Solutions open-source comme:

- Tomcat
- JBoss
- Enhydra
- Apache Projects
- Xerces
- Xalan
- SOAP
- Jakarta
- …

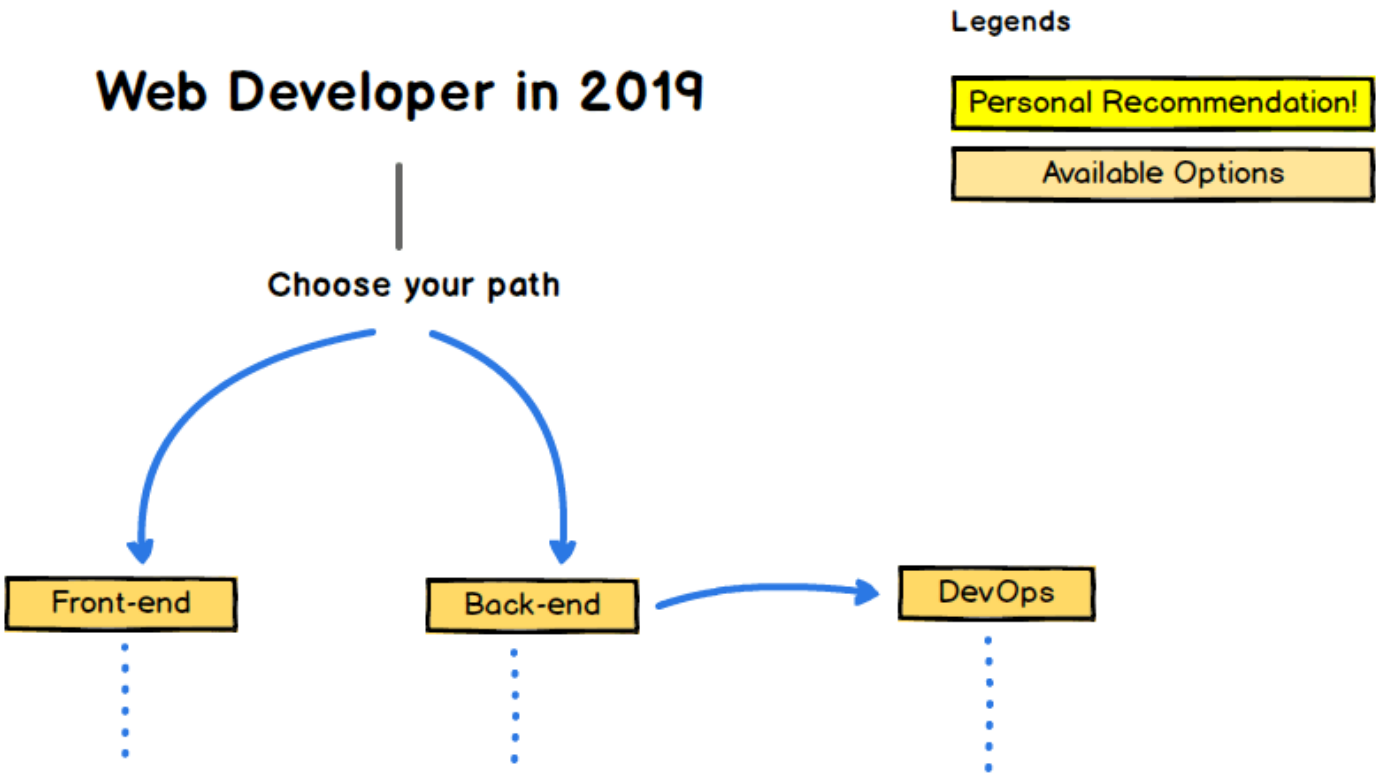
4. Serveurs d'applications



- Il s'agit en fait d'une nouvelle question de sondage,
- En termes simples, Tomcat possède le marché, avec plus d'adoption que tout le monde combiné, soit +61 %.
- Les autres serveurs semblent être utilisés par environ 7% du marché, dans une répartition relativement égale.

Frontend and Backend development

Devenir un développeur web



Your Guide to become a Moder Web Developer

5. Internet et le Web

Un site web est un ensemble constitué de pages web (elles-mêmes faites de fichiers HTML, CSS, Javascript, etc.). Lorsqu'on développe puis publie un site web, on met en réalité en ligne du contenu sur internet.

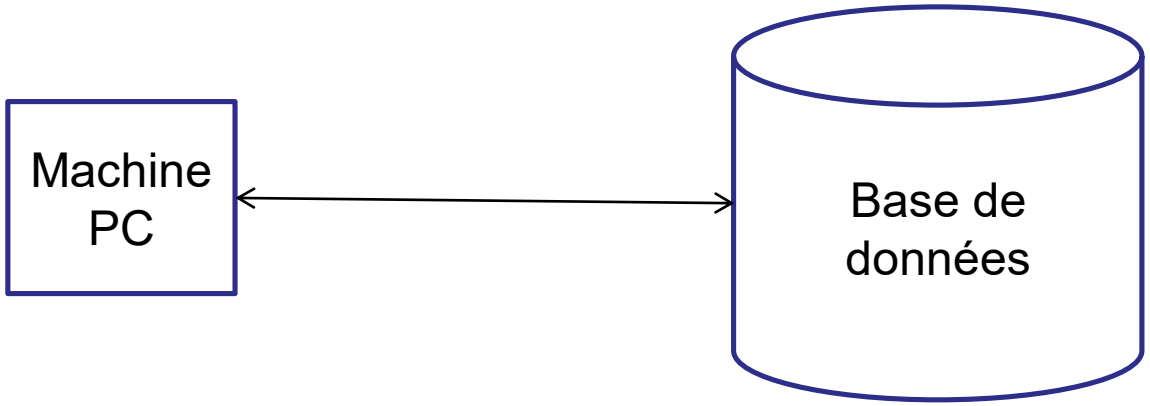
Il ne faut pas confondre l'internet et le web :

- l'internet est le réseau, le support physique de l'information. Pour faire simple, c'est un ensemble de machines, de câbles et d'éléments réseau en tout genre éparpillés sur la surface du globe ;
- le web constitue seulement un des services accessible sur l'internet. Vous connaissez et utilisez d'autres services, comme le courrier électronique ou encore la messagerie instantanée…

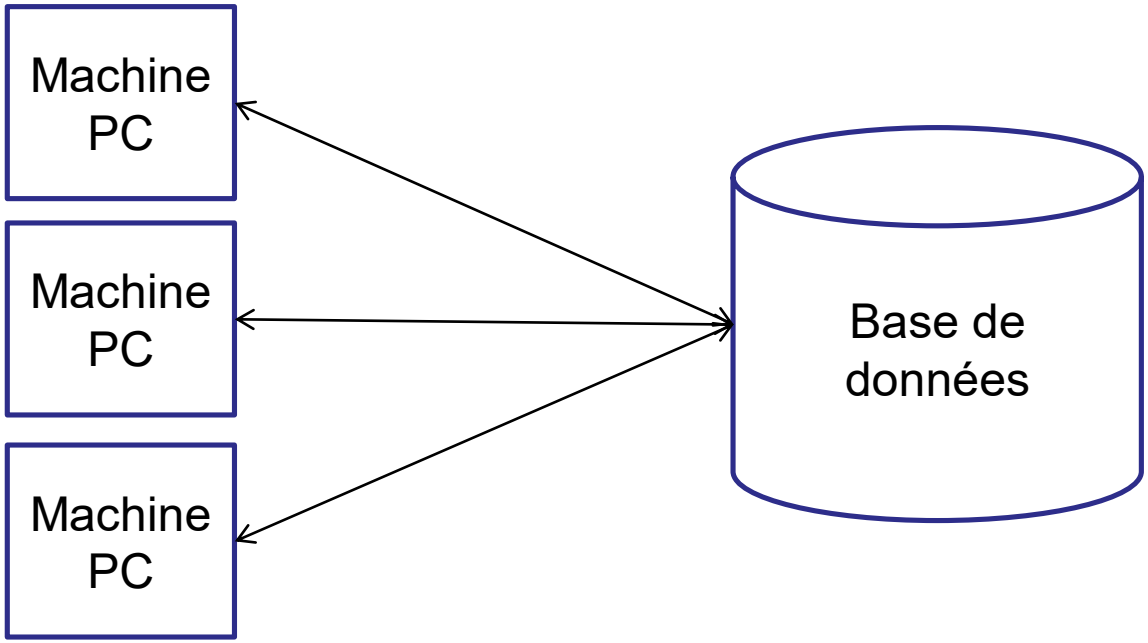
6. Evolution des architectures client-serveur

- Architecture des systèmes centraux
  - Tout est centralisé, pas de réseau
- Architecture client-serveur
  - Un serveur et des milliers de clients
- Navigateurs Web
  - Information sur serveur, consulté par des millions de clients.

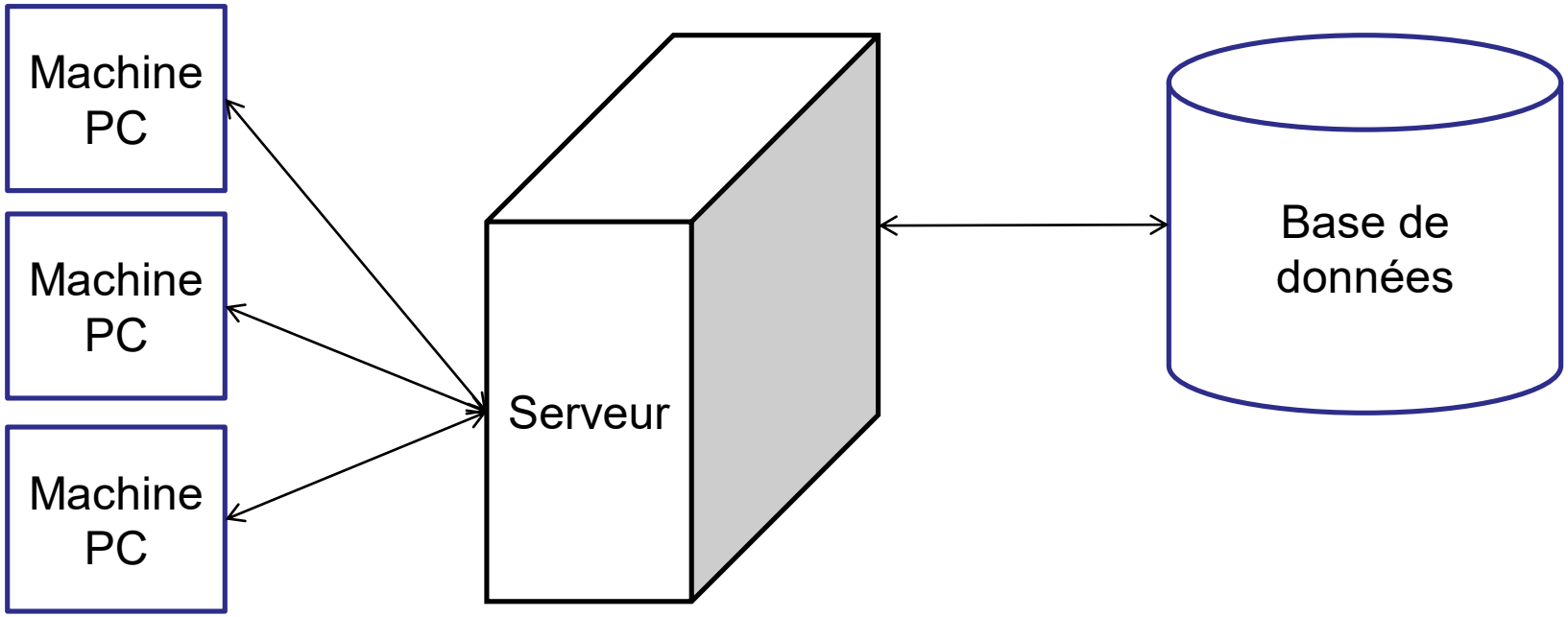
Architecture 1 tier (des systèmes centraux)



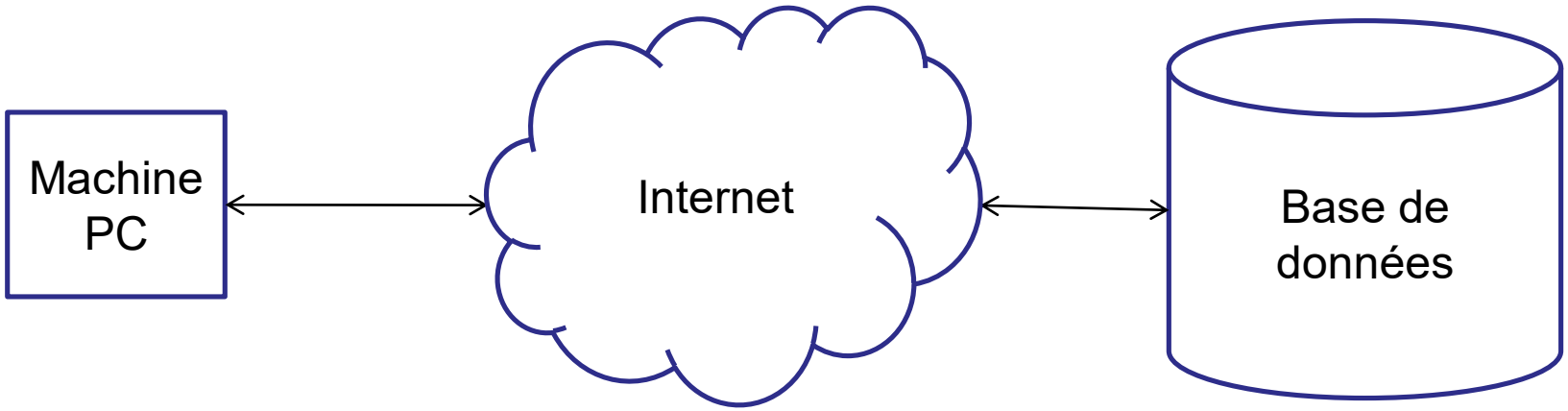
Architecture 2 tiers (client-serveur)



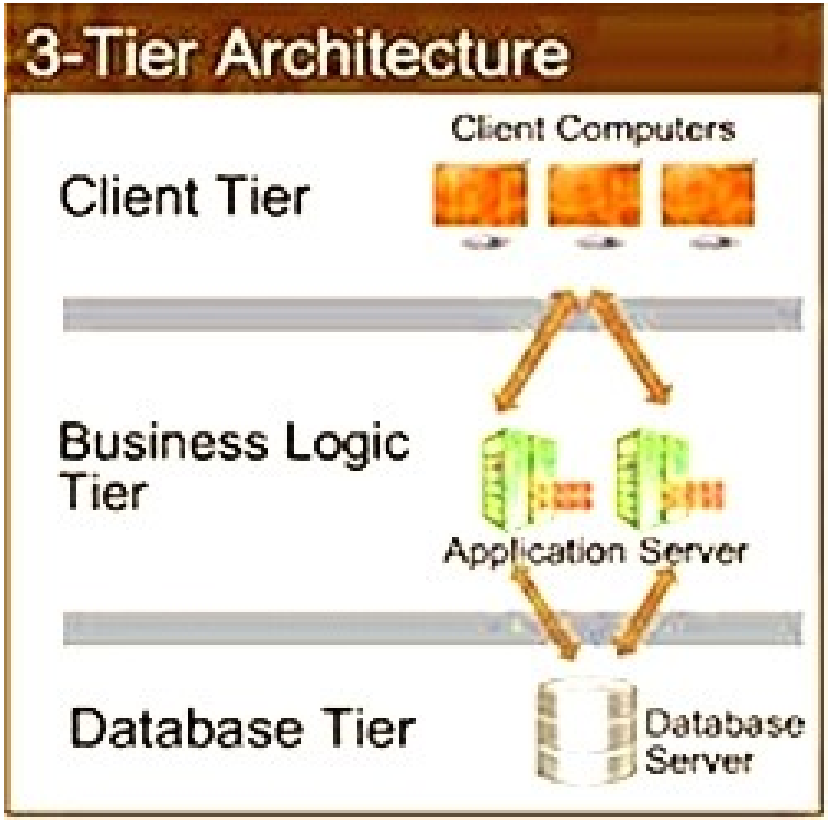
Architecture 3 tiers (RCP)

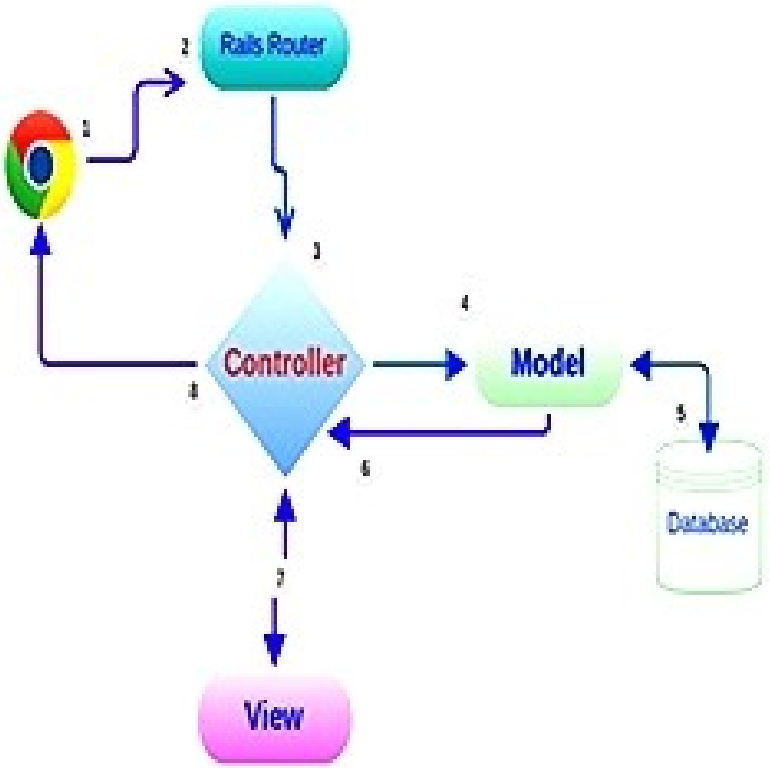


Architecture 3 tiers (Internet)



Architecture 3 tiers



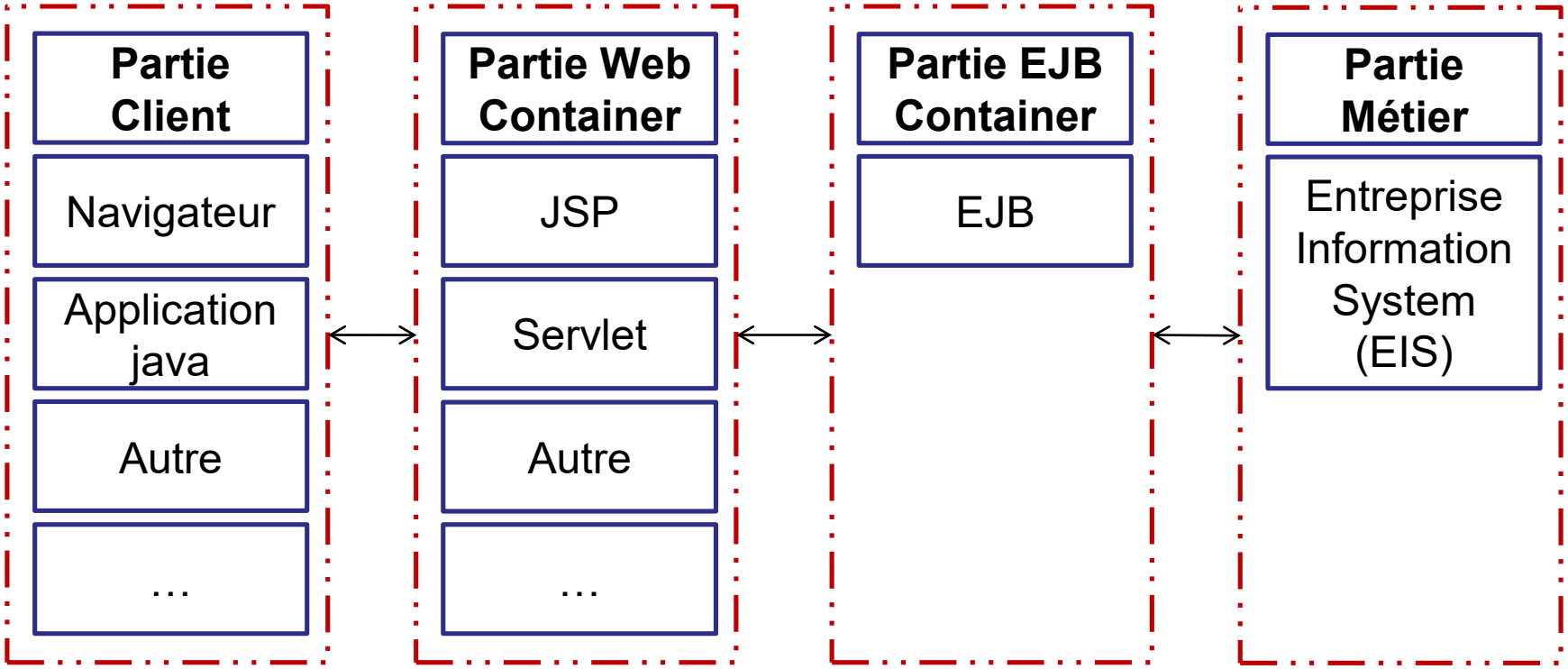


7. J2EE

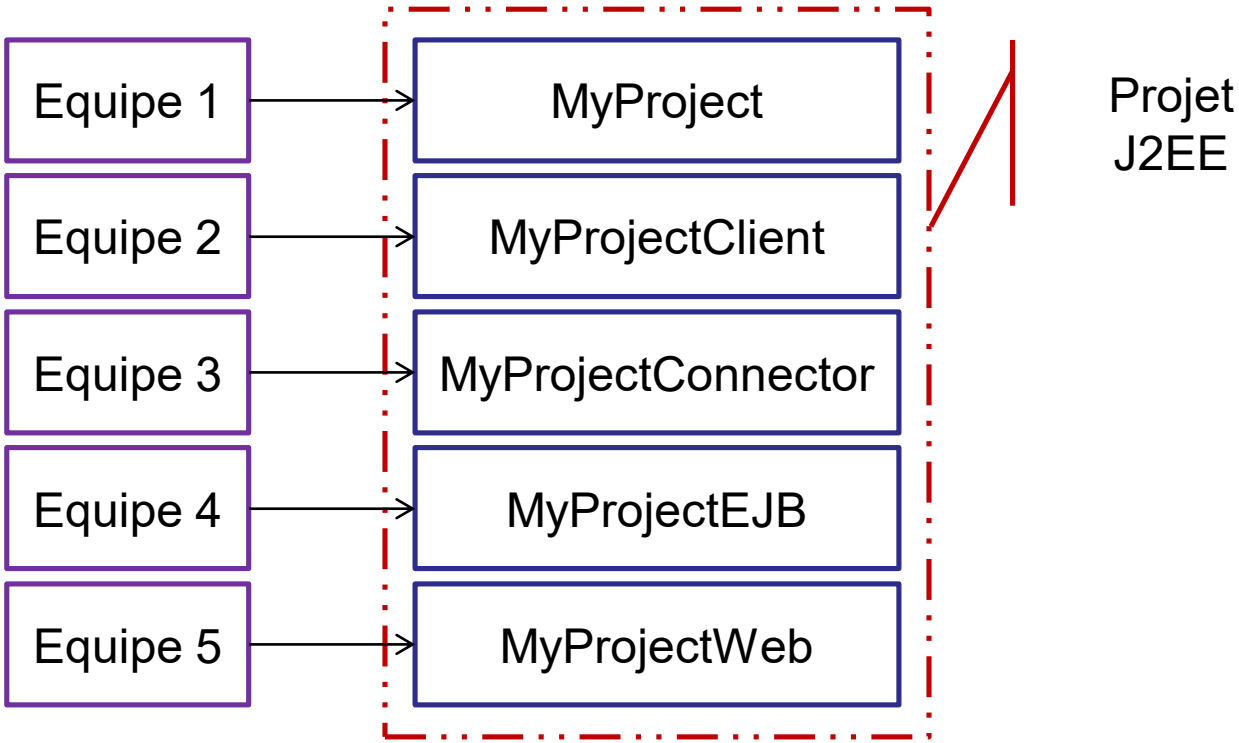
- Java fait bien évidemment référence à un langage, mais également à une plate-forme : son nom complet est « Java SE » pour Java Standard Edition, et était anciennement raccourci « J2SE ». Celle-ci est constituée de nombreuses bibliothèques, ou API (ex. java.lang, java.io, java.math, java.util, …).
- Java EE signifie Java Enterprise Edition, et était anciennement raccourci en « J2EE ». C'est une extension de la plate-forme standard. Autrement dit, la plate-forme Java EE est construite sur le langage Java et la plate-forme Java SE, et elle y ajoute un **grand nombre de bibliothèques** remplissant tout un tas de fonctionnalités que la plate-forme standard ne remplit pas d'origine.
- L'objectif majeur de Java EE est entre autres, de faciliter le développement d'applications web robustes et distribuées, déployées et exécutées sur un serveur d'applications.

Architecture J2EE

- Un projet J2EE est développé par plusieurs équipes. Chaque équipe développe une partie appelée « Module » comme indiqué sur la figure suivante:



Architecture J2EE



8. Objets répartis et architecture client-serveur

Les objets répartis ou distribués permettent de distribuer une application.

Un objet distribué peut être appelé par:

- un client qui appartient au processus contenant l'objet;
- un client de l'extérieur du processus contenant l'objet;
- un client distant sur le réseau.

9. Systèmes transactionnels

Sont des systèmes qui assurent la réalisation de transactions.

Une transaction est définie comme une unité de code **indivisible**.

Exemple:

Dans un système de e-commerce, une transaction de paiement en ligne doit débiter le compte de l'acheteur, et créditer le compte du vendeur.

Si une opération de ces deux opérations n'est pas faite, le tout sera annulé.

10. Exigence d'un projet informatique

- **Exigence fonctionnelles:**
  - Une application est créée pour répondre, tout d'abord, aux besoins fonctionnels des entreprises,
- **Exigences Techniques:**
  - Les performances
    - Temps de réponse
    - Haute disponibilité et tolérance aux pannes
    - Eviter les problèmes de montée en charge
  - La maintenance:
    - Une application doit évoluer dans le temps
    - Doit être fermée à la modification et ouverte à l'extension
  - Sécurité
  - Portabilité
  - Distribution
  - Capacité de communiquer avec d'autres applications distances
  - Capacité de fournir le service à différents type de clients (Desk TOP, Mobile, SMS, http…)
  - …
  - Coût du logiciel

11. Constat

- Il est très difficile de développer un système logiciel qui respecte ces exigences sans utiliser l'expérience des autres:
  - Serveur d'application JEE:
    - Tomcat, GlassFish
    - JBOSS, Web Sphere
    - …
  - Framework pour l'inversion de contrôle:
    - Spring (Conteneur léger)
    - EJB (Conteneur lourd)
  - Middlewares:
    - RMI, CORBA: application distribuées
    - JAXWS pour les Web services SOAP
    - JAXRS pour les Web services RESTful
    - JMS: Communication asynchrone entre les applications
    - …