

# Introduction à Java Enterprise Edition

Nicolas Delanoue

Université d'Angers - Polytech Angers



1 Java SE vs Java EE

2 Architectures

3 Développement

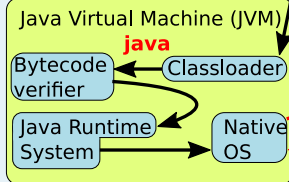
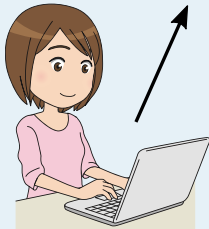
## Compilation et interprétation en Java SE

### Source code: **HelloWorld.java**

```
public class HelloWorld {  
    public static void main(String[] args){  
        System.out.println("Hello, world");  
    }  
}
```

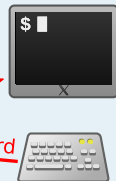
Java Compiler  
**javac**

Bytecode:  
**HelloWorld.class**



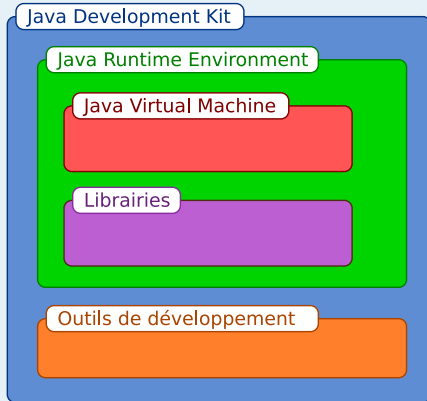
Standard output

Standard Input



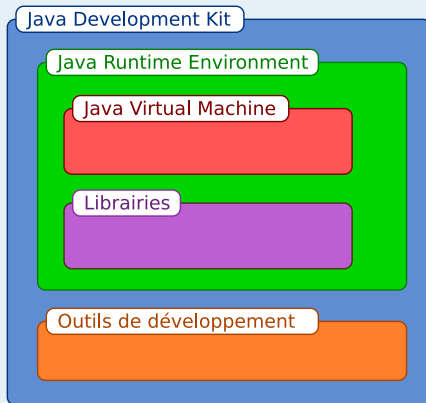
## Définition

Le Java Runtime Environment (JRE) une famille de logiciels qui permet l'exécution des programmes écrits en Java.



## Définition

Le Java Development Kit (JDK) contient de plus un ensemble d'outils avec lesquels le code Java peut être transformé en bytecode.



## Définition

Java Standard Edition est une spécification de la plate-forme Java d'Oracle, destinée typiquement aux applications pour poste de travail.

À chaque version de Java SE correspond notamment :

- les Java Specification Requests (JSR), i.e. les spécifications de la version considérée ;
- un Java Development Kit (JDK),
- un Java Runtime Environment (JRE).

## Remarque

Java Standard Edition est aussi appelée Java SE et anciennement Java 2 Platform, Standard Edition, ou encore J2SE.



## Remarque

Il existe plusieurs implémentations de Java SE.

- Oracle JDK,
- OpenJDK.

```
nico@pc:~$ javac -version
```

```
javac 11.0.9.1
```

```
nico@pc:~$ java -version
```

```
openjdk version "11.0.9.1" 2020-11-04
```

```
OpenJDK Runtime Environment (build 11.0.9.1)
```

```
OpenJDK 64-Bit Server VM (build 11.0.9.1)
```



## Définition

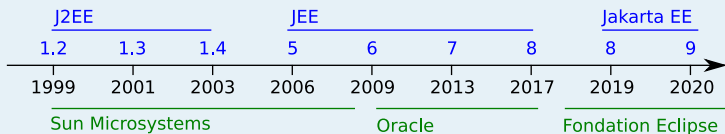
La plate-forme Jakarta Enterprise Edition étend Java SE en fournissant

- des API de mapping objet-relationnel,
- des architectures distribuées et multitiers,
- des services web,
- ...

## Remarques

- JEE propose finalement un ensemble de spécifications,
- JEE se fonde principalement sur des composants modulaires exécutés sur un serveur d'applications.
- JEE est une *spécification* destinée aux applications d'entreprises.

## Historique de Enterprise Edition



- une suite de tests (Java EE Compatibility Test Suite) pour vérifier la compatibilité ;
- une réalisation de référence (Java EE Reference Implementation).
- un catalogue de bonnes pratiques (Java EE BluePrints)

## Apports de Java Enterprise Edition

- Servlet : Composant représentant le C (Controller) du paradigme MVC
- Portlet : Conteneur Web (extension de l'API Servlet)
- JavaServer Pages (JSP) : Framework Web
- Java Standard Tag Library (JSTL) : bibliothèque de balises pour les JSP
- JavaServer Faces (JSF) : Java Server Face, Framework Web
- EJB : Composants distribués transactionnels
- JNDI : API de connexion à des annuaires, notamment LDAP
- JDBC : API de connexion à des bases de données
- JavaMail : API de gestion des mails
- JMX : Extension d'administration des applications
- JPA : API de gestion de la persistance des données
- JAXB : API de sérialisation par XML
- ...

## Classification des apports de Enterprise Edition

### Web

- Servlet
- Portlet
- JSP
- JSTL
- JSF

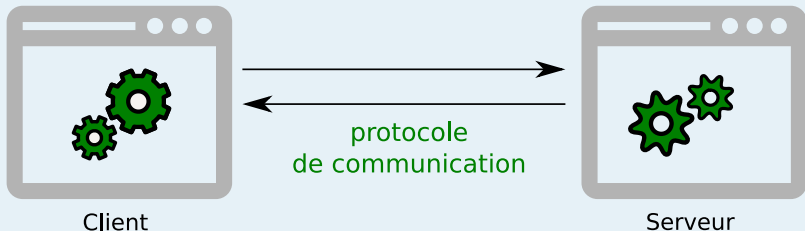
### Communication

- JNDI
- JMS
- JCA
- JavaMail
- JMX
- JAXM
- Java RIM
- ...

### Base de données

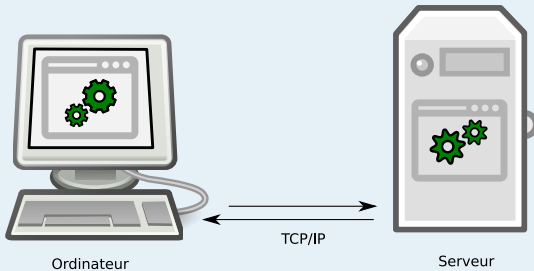
- JDBC
- JPA
- JTA

## Rappels : architecture client serveur



## Exemple d'installation

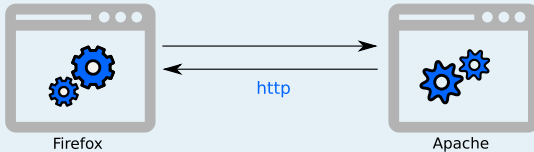
Concrètement, on peut avoir ce genre d'installation :



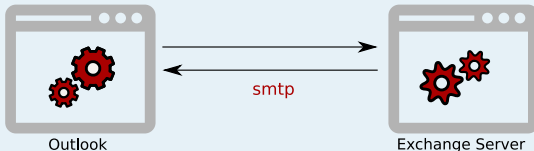
## Remarques

- Les couches TCP et IP nous permettent de nous abstraire du matériel.
- Les applications clientes et serveurs peuvent être installées sur la même machine.

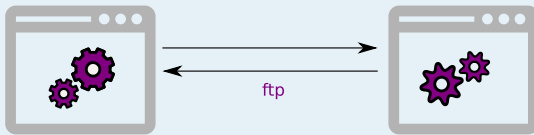
### Exemple navigation web



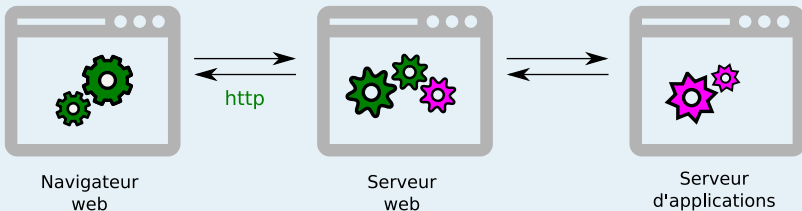
### Exemple envoi de mail



### Exemple transfert de fichiers

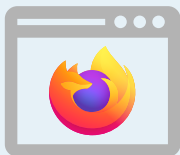


## Architecture basique et classique en Java EE





## Exemple d'implémentation d'une architecture Java EE



Firefox



Apache httpd



Tomcat

## Exemple d'implémentation d'une architecture Java EE



### Serveurs http

- apache
- nginx
- lighthttpd
- Internet Information Services
- ...

### Serveurs d'application

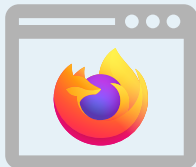
- Tomcat (Fondation Apache)
- Glassfish (Oracle)
- WildFly / JBoss (Red Hat)
- WebSphere (IBM)
- ...

[https://koor.fr/Java/TutorialJEE/jee\\_application\\_servers.wp](https://koor.fr/Java/TutorialJEE/jee_application_servers.wp)

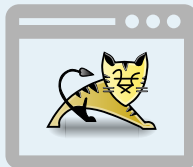
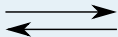
## Remarque

Les serveurs d'applications peuvent aussi servir de serveur http.

Exemple où Tomcat sert de serveur d'applications et de serveur http



Firefox

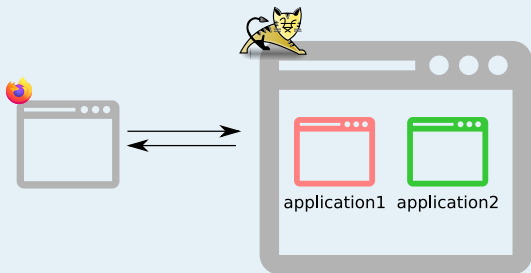


Tomcat

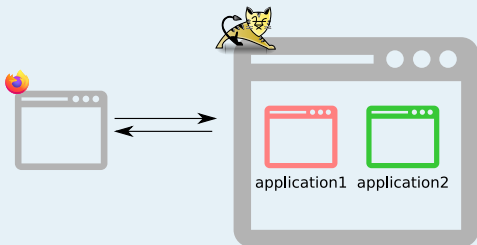
## Définition

Un serveur d'applications est un logiciel d'infrastructure offrant un contexte d'exécution pour des composants applicatifs.

## Exemple



**FIGURE** – Un serveur d'applications Tomcat sur lequel deux applications sont déployées.



## Remarques

- Généralement, le serveur d'applications fournit aussi un ensemble de composants accessibles au développeur de logiciels via des API standards définies pour la plate-forme elle-même.
- Finalement, il fournit des fonctionnalités proches de celles offertes par un système d'exploitation.

## Exemples de serveurs d'application JEE

Tomcat, Glassfish, WildFly, WebSphere, ...

## Remarques

- Glassfish est l'implémentation de référence pour la spécification Java EE (pas forcément le plus efficace),
- Tomcat n'implémente qu'une partie de la spécification (e.g. pas les EJB).

## Comparatifs :

[https://koor.fr/Java/TutorialJEE/jee\\_application\\_servers.wp](https://koor.fr/Java/TutorialJEE/jee_application_servers.wp)

## Motivations pour les serveurs d'applications

- simplifier le développement et séparer clairement les différentes couches,
- obtenir des gains de productivité en facilitant l'intégration avec l'existant,
- optimiser la maintenance et la sécurité,
- permettre la liberté de choix,
- proposer un accès transparents aux données,
- gérer la charge avec du clustering, équilibrage de charges, basculement automatique, ...

## Principal avantage

Le développeur se concentre principalement sur la logique de l'application : i.e. le "Business Logic" ou la couche métier.

## Clustering

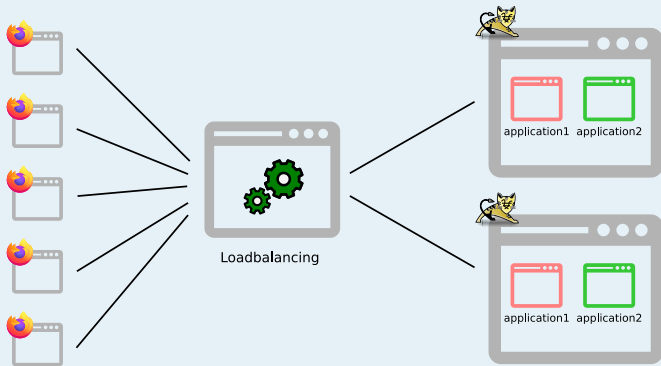
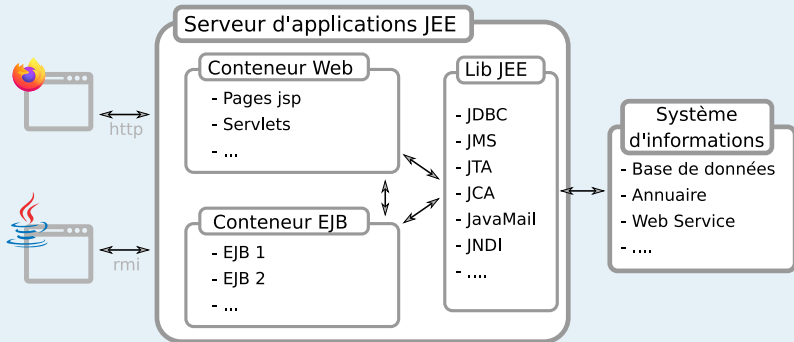


FIGURE – Equilibrage de charges.



## Architecture plus détaillée



## Remarque

Tomcat ne contient pas de conteneur EJB.

Une application Java EE assemble des composants

composants clients : HTML, applets

composants Web : servlets et JSP

composants business : EJB

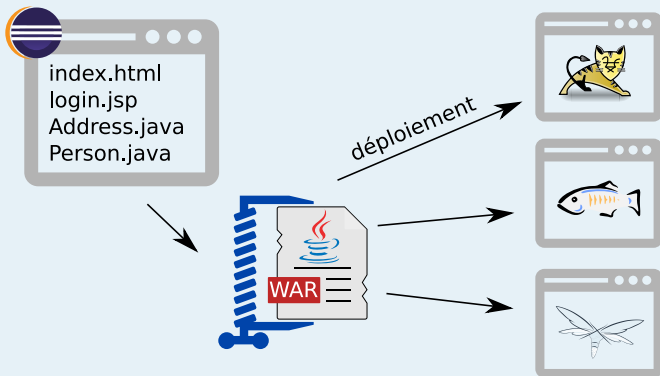
Les composants sont

- écrits en Java compilé en bytecode,
- assemblés dans l'application Java EE,
- déployés dans un serveur Java EE.

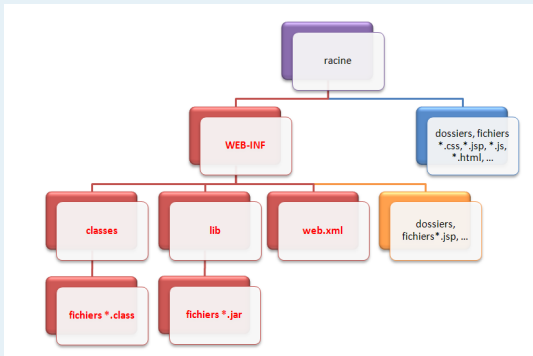
## Définition

Une application web est archivée pour déploiement dans un fichier war (Web Application Archive).

## Développement et déploiement d'une application



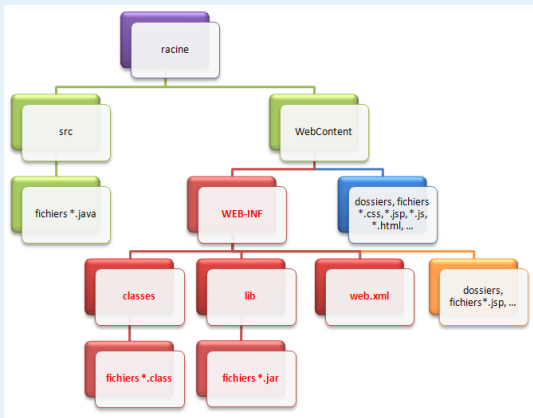
## Struture d'une application Java EE, i.e. contenu du fichier war



## Code couleur

- rouge : obligatoire pour une application JEE
- bleu : accessibles directement par le client via leurs URL,
- rouge/orange : ne sont pas accessibles directement par le client.

## Struture d'un projet Web (en phase de développement)



### Remarque

Grâce à l'environnement de développement d'Eclipse, la construction (build) et le déploiement (run) se feront facilement.

## Arborescence de Tomcat

- `./bin` fichier binaire,  
(pour lancer Tomcat par exemple)
- `./conf` fichiers de configuration (sécurité, déploiement, ...),
  - `./lib` une partie les librairies Java EE !
- `./log` fichiers de journalisation,
- `./webapp` les applications  
(il y en a par défaut : docs, exemples, ...)
- `./work` Répertoire de travail de Tomcat  
(c'est ici par exemple que les jsp sont compilées)

## Résumé

- Intérêts de Java et Java EE
  - solution portable
  - solution sécurisée
  - solution standardisée
- Application Web : assemblage de composants (bibliothèque), de contenu et de fichiers de configuration
- Environnement d'exécution d'une application web : Web container

## Organisation temporelle

Un thème par semaine avec, à chaque fois :

- apport des connaissances pendant une séance d'1h20,
- travaux pratiques pendant 3 séances d'1h20.

## Programme

- 1 Installation poste de travail (IDE & Serveur d'applications).
- 2 Servlets et JSP.
- 3 Premier MVC sans Framework avec JSP, EL , Javabeans (POJO).
- 4 JPA - ORM Hibernate. ...
- 5 Web service SOAP CXF ou/et REST.
- 6 Evaluation.