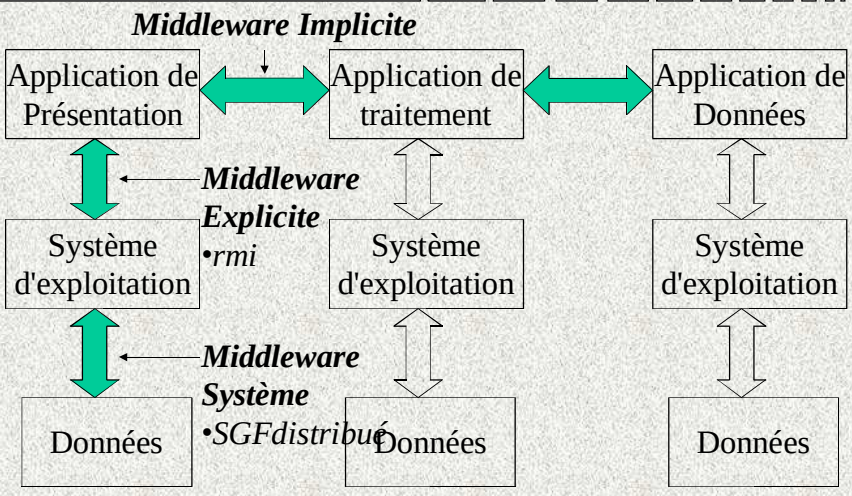
Les Middlewares : De l'approche implicite à l'architecture J2EE

Répartition d'une application



Du C/S au middleware Explicite

- CS :
  - Le client et le serveur sont développés en collaboration
- Objet distant :
  - Client et serveur sont liés par une interface
  - La couche réseau est masqué au client et au serveur
- ==> Notion de code applicatif/code non applicatif

Code applicatif/code non applicatif

- Le **code applicatif** est le code propre à l'application développée (aussi appelé **code métier** ou **BusinessCode**)
  - Ex : Banque ==> compte, retrait, dépôt
- Le code non-applicatif est le code non spécifique à l'application développée
  - Ex :
    - Accès réseau, Accès à la base de données, Debug, Log…

==> Pourquoi ne pas automatiser systématiquement le code non-applicatif

Principes

- Le code non-applicatif :
  - Ne doit pas apparaître dans le code applicatif
  - Il est accessible par la notion de services
    - Service de persistance (Base de données)
    - Service de présentation (html/http)
    - Service de log, Service d'authentification
    - Service de cycle de vie
  - Un service c'est
    - une interface = ensemble de méthodes
    - une ou plusieurs implantations
  - La programmation objet permet de masquer entièrement le comportement d'un objet
  - L'association code applicatif/non-applicatif se fait de manière déclarative
    - résolue au run-time ==> souple, adaptable

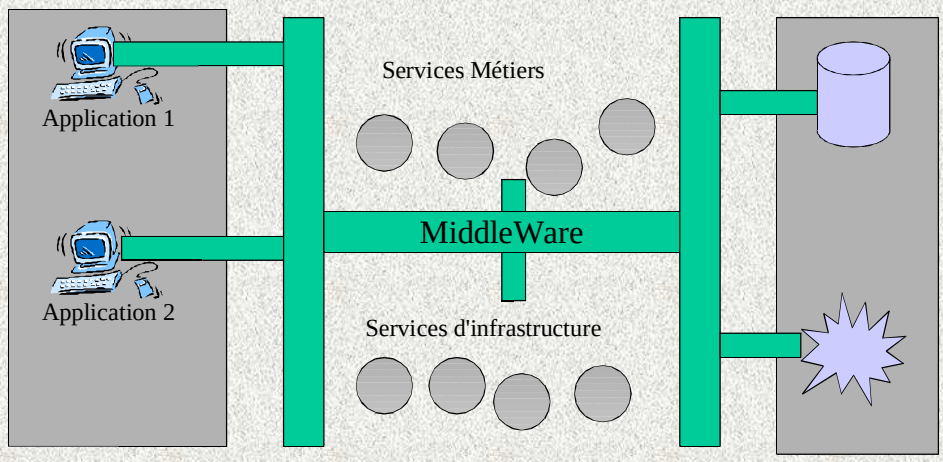
Exemple : Logger

```
public class test{
  public void uneMethode(){
    System.out.println("entrée dans la méthode");
    i++;
    System.out.println("sortie de la méthode");
  }
}
```

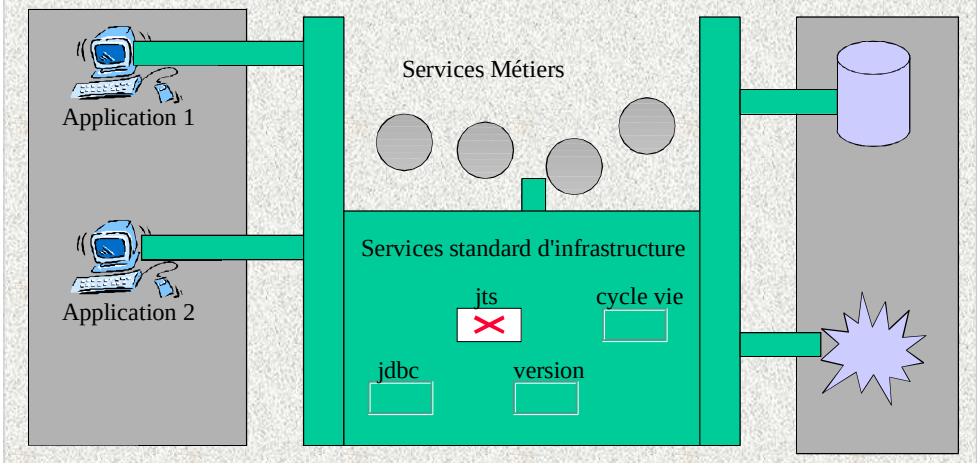
==> Quels sont les inconvénients de ce code ?
==> Quels sont les avantages de passer par un service ?

Les Conteneurs

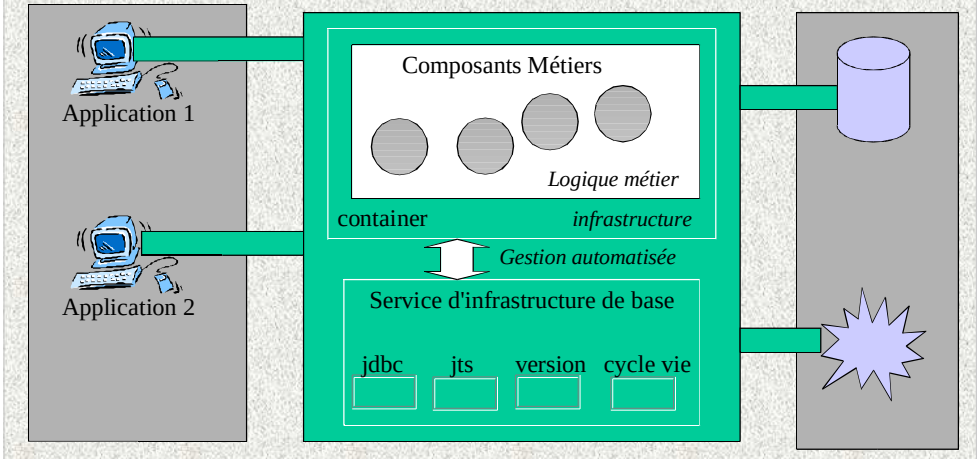
Architectures à Objet Distribués



Serveur de composants de base



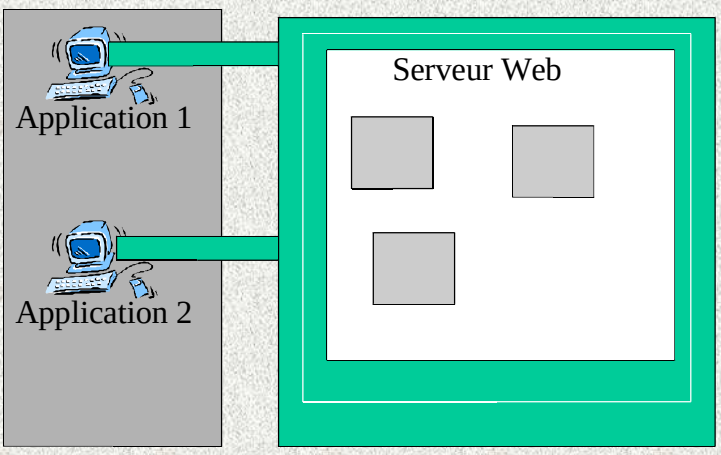
Serveur de composants intégré



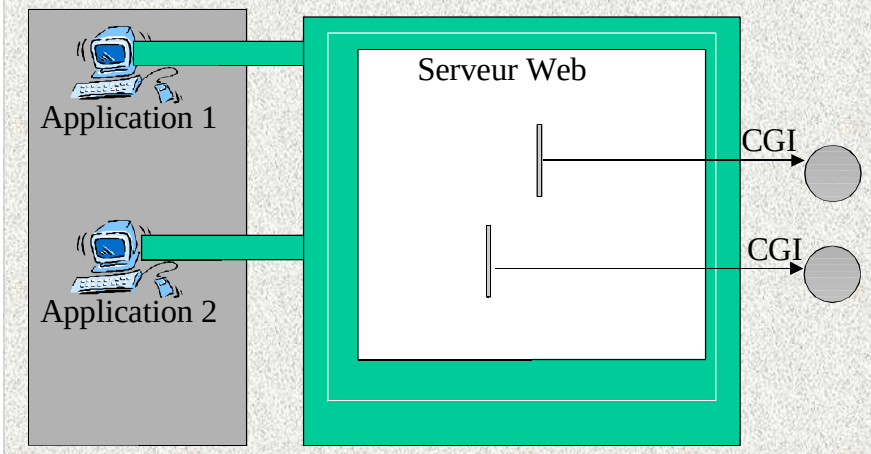
Services du container d'objets métiers

- Services internes
  - Gestion de la charge du serveur
    - (cycle de vie, accès client, passivation...)
  - Service de nommage
  - Gestion des accès aux objets métiers
- Services externes
  - Gestion du mapping sur BD relationnelle
  - Gestion des transactions
  - Gestion des échanges de messages

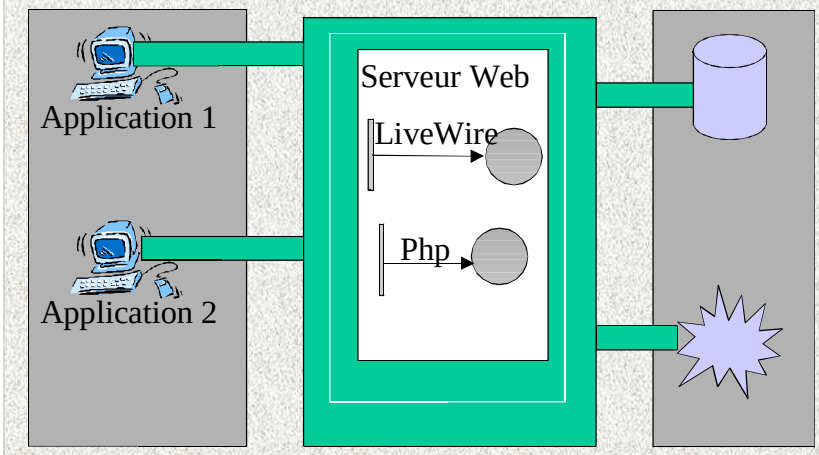
Serveur Web dynamique



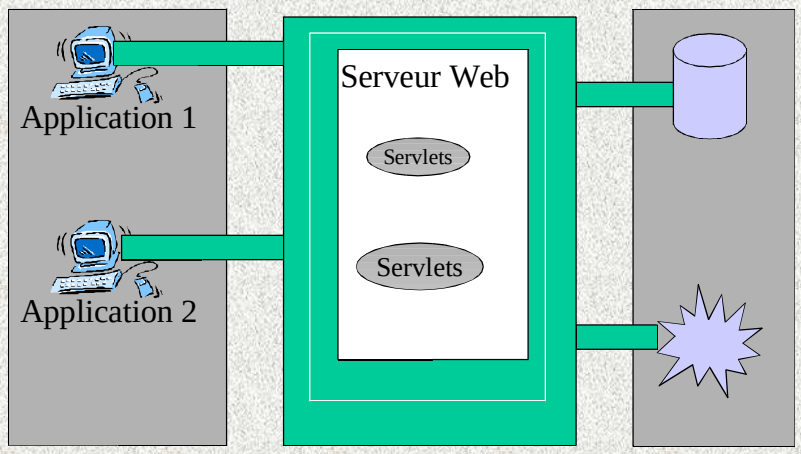
Serveur Web dynamique



Serveur Web dynamique



Serveur Web dynamique (Servlets / JSP)



Services du container de pages Web

- Servlet / JSP
- Services internes
  - Gestion de la charge du serveur
    - cycle de vie
  - Gestion des autorisations d'accès
- Services externes
  - API Java

Un conteneur c'est :

- Une boîte qui automatise
  - La communication avec des services non-fonctionnels
  - La gestion des applications
  - Le cycle de vie d'une application pour son client

==> Qui réalise une interception entre le « client » et le « service » afin de réaliser des tâches

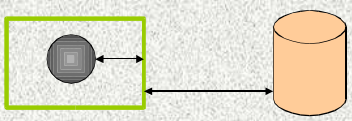
- Economie de code,
- Economie de moyen,
- Simplification pour le programmeur,
- et l'hébergeur

Interception de code 1/3

- Explicite
  - Le développeur inclut son propre code d'accès au service

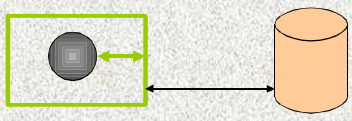
Interception de code 2/3

- Implicite : Le conteneur joue un rôle de proxy
  - Le conteneur fournit une API similaire
  - Le client croit voir une base de données alors qu'il voit le conteneur
  - ==> Avantages



Interception 3/3

- Automatique : Le conteneur automatise la vision du service
  - Le conteneur réalise les opérations standards du client
  - Le client ne voit rien, il est automatiquement peuplé de données
  - ==> Exemple : Base de données, Transaction...

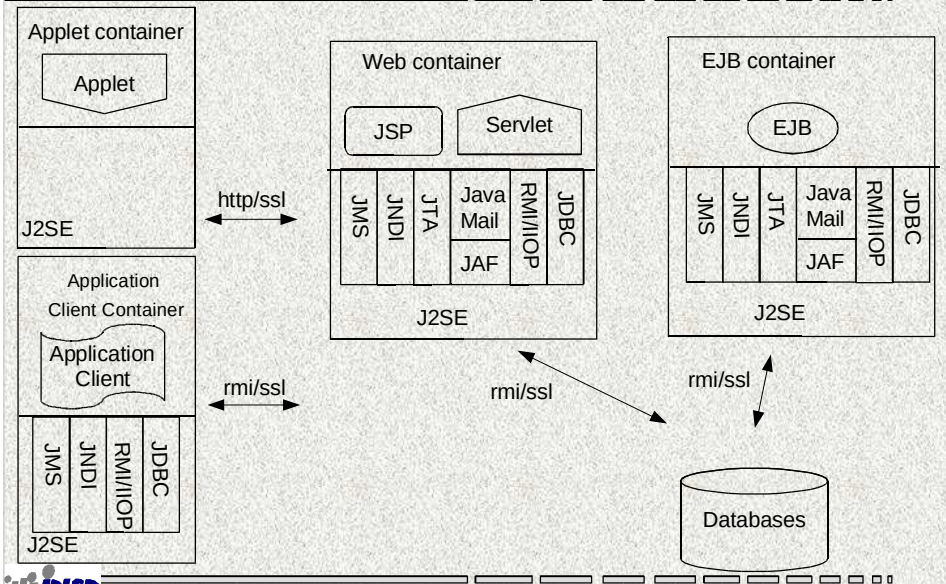


Les Serveurs d'applications

Un serveur d'application

- Application qui cherche à simplifier la programmation, et l'administration de grands systèmes
- Hébergeant :
  - Des containers
    - Pages Web
    - Composants Métier (EJB)
  - Des services
    - Nommage
    - Base de données/Mapping sur Base
    - Moniteurs transactionnels
    - Déploiement ...
  - Des API sur les services
    - JDBC/JTS/JMS…
- Deux grandes familles de serveurs d'applications
  - Les interfaces utilisateurs
  - Les applications distribuées

The J2EE Architecture



Les offres de serveur d'applications

- Serveur d'application J2EE
  - Weblogic BEA,WebSphere IBM, Iplanet Sun, Oracle
  - Jonas (ObjectWeb)
  - Jboss
- Autres serveurs d'applications
  - Microsoft .net
  - Zope (Python)
  - OpenACS
  - Serveur CORBA (OrbixWeb)

Le développement d'applications sur les SA

- Développement du code applicatif :
  - Phase de développement « classique »
  - L'appel à des services externes se fait soit :
    - de manière explicite dans le code
    - de manière implicite
- Packaging du code
  - Le code est regroupé dans une archive (jar, tar, rpm)
    - Du code d'exécution
    - Des indications de dépendances
    - Des indications d'interaction avec les services
- Déploiement du code code
  - Le code est déployé sur une machine d'exploitation
    - Si il y a des bugs, l'ensemble du code est réinstallé

Répartition d'une application