

# L'utilisation de frameworks pour le développement avec Java EE

## Module 1 - Présentation



# Objectifs

- Se familiariser avec l'architecture multicouches
- Comprendre l'intérêt des frameworks
- Découvrir la notion d'intégration continue

# Problématique

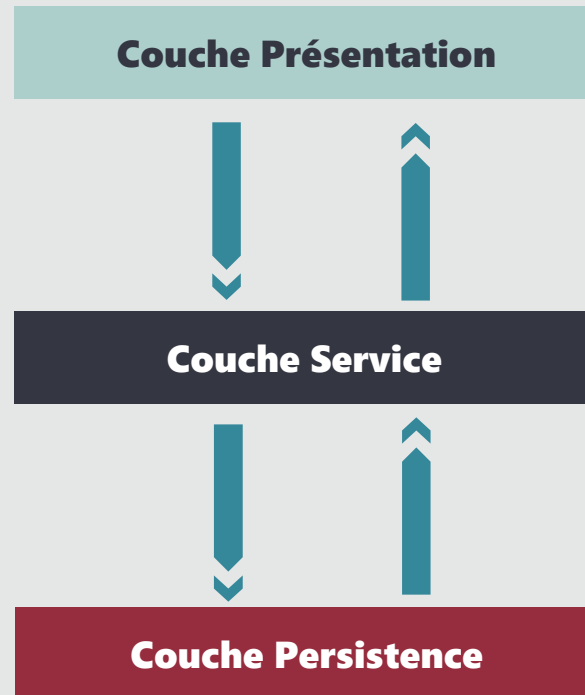
- L'industrialisation du développement d'applications Java distribuées demande une rigueur aujourd'hui incontournable
  - Une architecture multicouches (N-tiers)
  - L'utilisation de frameworks basés sur des design patterns
  - Une automatisation du process de déploiement

# Une architecture multicouches

- Classiquement, découpage en trois couches :
  - La couche présentation (tiers web)
  - La couche métier (tiers business ou tiers service)
  - La couche de stockage des données (tiers persistance)

Présentation

# Une architecture multicouches



# La couche présentation

- Permet la communication avec l'utilisateur
  - Client lourd « non adapté »
    - Application spécifique installée sur le poste client
  - Client léger adapté
    - Utilisation d'un navigateur
    - Affichage habituellement en HTML
    - Mise en page via du CSS
    - Gestion des événements avec JavaScript

# La couche service

- Correspond à la couche fonctionnelle
  - Services applicatifs
  - Services métier
- Les données transitent obligatoirement par cette couche
  - Traitement des données avant persistance
  - Formatage des données avant affichage

# La couche persistance

- Couche permettant l'accès aux données
  - En base de données relationnelle (SQL : MySQL, PostgreSQL, Microsoft SQL Server, Oracle Database...)
  - En base de données non relationnelle (NoSQL : MongoDB, Cassandra...)
  - En fichier texte formaté ou non (XML...)
  - En données gérées par une autre application (Excel, application de comptabilité...)



# Les frameworks

- Qu'est-ce qu'un framework ?
  - Un "cadre de travail"
  - Ensemble d'outils et de composants logiciels fournissant un "squelette" de programme
- But
  - Simplifier la tâche du développeur
  - Fournir un cadre réutilisable et standardisé

# Les frameworks

- Avantages

- Réduction du temps de développement (composants déjà écrits et réutilisables)
- Uniformisation et cohérence du codage
- Robustesse, évolutivité
- Pérennisation de la solution en entreprise
- Grande communauté

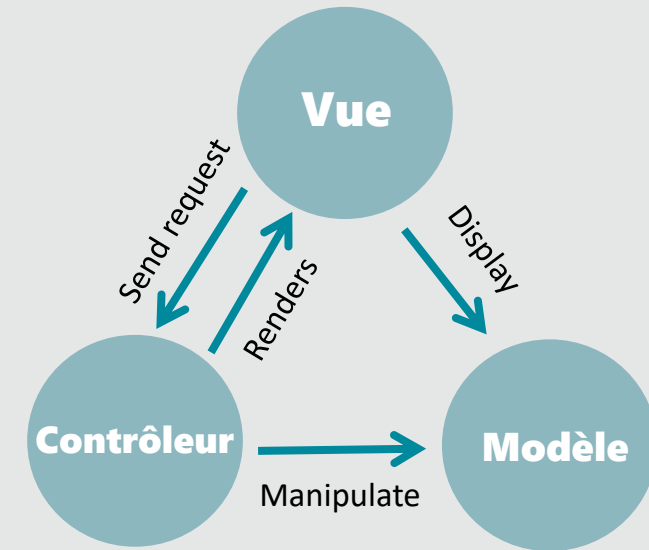
- Inconvénients

- Apprentissage du framework
- Connaissance préalable des design patterns utilisés

# Les frameworks de développement web

- Le design pattern MVC

- Comporte trois types de modules : les modèles, les vues et les contrôleurs
- Modèle
  - Élément qui contient les données ainsi que de la logique en rapport avec les données comme la validation
- Vue
  - C'est l'interface graphique
  - Contient les composants d'affichage des données ainsi que des composants permettant d'effectuer des actions
- Contrôleur
  - Module qui traite les actions de l'utilisateur, modifie les données du modèle et de la vue



# Les frameworks de développement web

- Struts 2
  - Apache
  - Basé sur les actions
- Spring MVC
  - Basé sur les actions
  - Très utilisé en entreprise
- JavaServer Faces
  - Java EE
  - Basé sur les composants

# Les frameworks de persistance

- Le Mapping Objet-Relationnel (ORM)
  - Technique de programmation permettant de manipuler des données stockées dans un Système de Gestion de Base de Données Relationnelles (SGBD) au sein d'objets Java
- Les principaux frameworks
  - Hibernate
  - EclipseLink
  - Java Persistence API
    - Interface requérant une implémentation
  - Spring Data

# Le framework Spring

- Créé en 2004 par Rod Johnson
- C'est un framework qui propose un socle pour le développement d'applications
  - Java SE ou Java EE
- Repose sur la notion de conteneur léger
  - Prise en charge du cycle de vie d'objets
- Utilise le concept d'inversion de contrôle

# Intégration continue

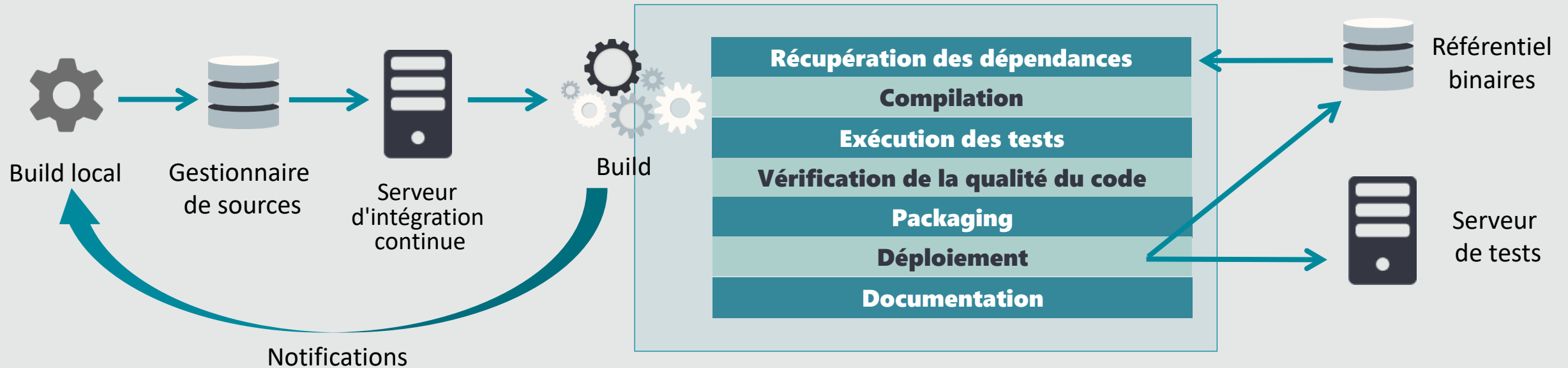
- Définition

- Ensemble de pratiques utilisées en génie logiciel consistant à vérifier à chaque modification de code source que le résultat des modifications ne produit pas de régression dans l'application développée (Wikipédia)

- Avantages

- Améliorer la productivité des développeurs
- Trouver et corriger plus rapidement les bogues
- Livrer plus rapidement des mises à jour

# Présentation Intégration continue





Présentation

# Intégration continue



# Intégration continue

- Les outils
  - Build local
    - Maven, Gradle, Ivy
  - Gestionnaire de sources
    - CVS, SVN, Git
  - Serveur d'intégration continue
    - Jenkins, GitLab
  - Build
    - Maven, Gradle, Ivy
  - Référentiels binaires
    - Nexus, Jfrog
  - Serveur de tests
    - Tomcat, WildFly, ...