Université Nord Américaine privée
Institut International Technologie

Génie Logiciel Avancée
Architecture logicielle évoluée : Framework Spring

02- Présentation du framework Spring

Mohamed ZAYANI

2025/2026

Plan

- Notion de framework
- Types de frameworks
- Introduction au Spring
- Caractéristiques principales de Spring
- Modules principaux de Spring
- Arrivée de Spring Boot
- Caractéristiques principales de Spring Boot

Notion de Framework

❑Définition

- Un **framework** est un ensemble d'outils et de composants logiciels organisés conformément à un plan d'architecture et des patterns et fournit de bibliothèques et de règles pour développer plus rapidement et plus facilement des applications.

- Contrairement à une librairie, qui est un ensemble de fonctions qu'on appelle quand on en a besoin, un framework définit **un cadre de travail** (squelette) dans lequel l'application va s'insérer.

- **Remarque:**
  - Avec une librairie : le développeur contrôle l'exécution (c'est lui qui appelle la librairie).
  - Avec un framework : c'est le framework qui contrôle l'exécution: principe d'Inversion of Control (IoC).

Notion de Framework

❑Objectifs

- **Simplifier** le travail des développeurs informatiques en leur offrant une architecture «prête à l'emploi» basée sur l'expérience des développements antérieurs.
- Faire **gagner du temps** (ne pas réinventer la roue).
- Constituer un conteneur qui gère **le cycle de vie** des objets manipulés par l'application.

→L'utilisation d'un framework permet:

- la réutilisation des codes
- la standardisation de la programmation (structure unique pour toutes les applications)
- L'amélioration de la productivité de développement: offre de bonnes pratiques intégrées (patrons de conception, sécurité, gestion des ressources).
- Une maintenance moins coûteuse

Types de frameworks

Il est possible de classer les frameworks selon le domaine d'utilisation:

1. **Frameworks « Web »:**
   - **Java** : Spring MVC, JSF (JavaServer Faces), Struts.
   - **Python** : Django, Flask.
   - **PHP** : Laravel, Symfony.
   - **JavaScript** : Angular: React, Vue.js, Next.js.
2. **Frameworks « Desktop »:**
   - **Java** : JavaFX, Swing (plus ancien).
   - **Python** : PyQt, Tkinter.
   - **C#/.NET** : Windows Presentation Foundation (WPF), WinForms.
3. **Frameworks « Mobile »:**
   - **Java/Kotlin** : Android SDK.
   - **Swift** : SwiftUI (pour iOS).
   - **Cross-platform**: Flutter (Dart), React Native (JS), Ionic.

Types de frameworks

4. Framework pour la persistance et la gestion des données

- **Java** : Hibernate, Spring Data (pour Java Persistence API)
- **.NET** : Entity Framework.
- **Python** : SQLAlchemy, Django ORM.

5. Frameworks pour tests unitaires

- **Java** : JUnit, TestNG, Mockito.
- **Java Script** : Jest, Mocha.
- **Python**: PyTest, Unittest.

6. Frameworks pour la plateforme Java:

- **Spring Framework** (complet : injection de dépendances, web, sécurité, data, cloud).
- **Hibernate** (ORM pour la persistance des données).
- **Struts** (ancien, pour le web, largement remplacé par Spring MVC).
- **JSF** (JavaServer Faces, soutenu par Oracle, mais moins populaire que Spring).

Introduction au Framework Spring

- Spring est un framework open-source pour Java créé par Rod Johnson en 2003.
- Spring est un framework qui offre les possibilités d'un serveur d'application JEE.
- Spring est considéré comme un conteneur « léger » puisqu'il nécessite pas un serveur pour charger ses modules ni des interfaces à implémenter (au contraire des conteneurs JEE lourds comme Jboss, WildFly, Jonas)
- Spring est rapidement devenu le framework le plus utilisé dans l'écosystème Java.

Objectifs principaux:

- ✓ Simplifier le développement d'applications Java complexes.
- ✓ Fournir une architecture cohérente, modulaire et extensible.
- ✓ Favoriser le faible couplage via l'Inversion de Contrôle (IoC) et l'Injection de Dépendances (DI).

Caractéristiques principales de Spring

1. Léger et modulaire:

- Il est possible d'utiliser uniquement des parties particulières (ex: Spring Data sans utiliser Spring MVC).

2. Inversion of Control (IoC) & Dependency Injection (DI):

- Le framework se charge de créer et gérer les objets (beans) à la place du développeur.
- Cela réduit le couplage et favorise la testabilité.

3. Programmation orientée aspect (AOP) :

- Permet de séparer les préoccupations transversales (logging, sécurité, transactions) du code métier

4. Intégration avec d'autres technologies:

- Fonctionne avec Hibernate, JPA, JMS, RabbitMQ, Kafka, etc.
- S'intègre facilement avec des API REST et SOAP

5. Sécurité:

- Avec Spring Security, gestion complète de l'authentification, autorisation, rôles, OAuth2, JWT.

6. Portabilité:

- Fonctionne aussi bien en local, sur un serveur d'application ou dans le cloud (Spring Cloud).

Modules principaux de Spring

Spring est composé de modules organisés autour d'un noyau (Core Container):

1. Core Container (Noyau):

- **Spring Core** : fournit IoC et DI.
- **Spring Beans** : gestion des objets (beans) et de leur cycle de vie.
- **Spring Context** : comme un "conteneur" qui gère les beans et permet de les récupérer.
- **Spring Expression Language (SpEL)** : langage d'expression pour manipuler des objets dans le conteneur.

2. Spring AOP (Aspect-Oriented Programming):

- Permet d'ajouter des comportements **transversaux** (transactions, logs, sécurité) sans modifier le code métier.

3. Spring Data Access / Integration:

- **Spring JDBC** : simplifie l'accès aux bases de données.
- **ORM** : intégration avec Hibernate, JPA.
- **Transactions** : gestion déclarative des transactions.

4. Spring Web:

- **Spring MVC** : framework web basé sur MVC (Model-View-Controller).
- **Support REST** (création d'API RESTful).
- **Intégration avec WebFlux** pour la programmation réactive.

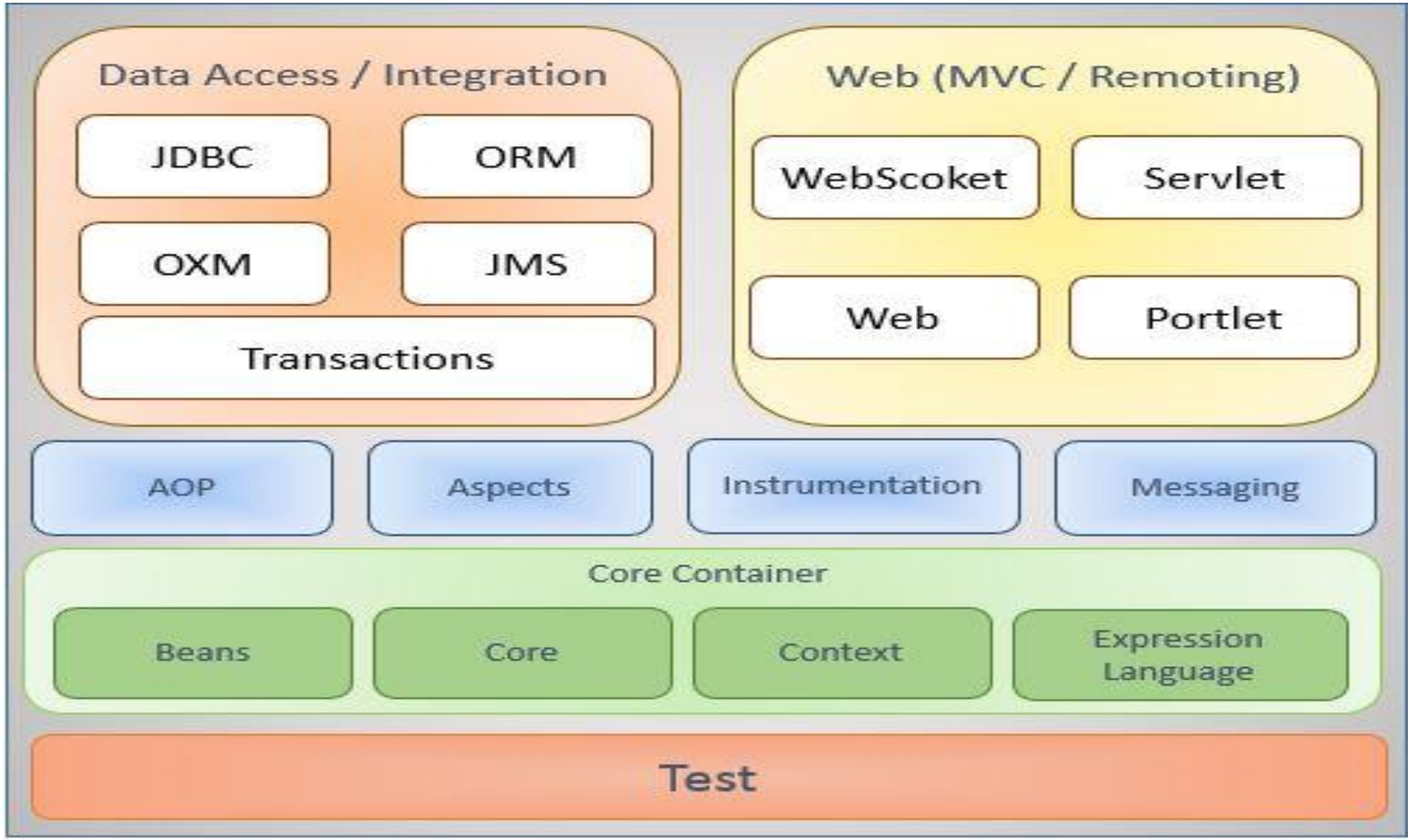
5. Spring Security:

- Gestion complète de la sécurité (authentification, autorisation, cryptage, OAuth2, JWT..

6. Spring Test:

- Support pour JUnit, Mockito et tests d'intégration.

Spring Runtime



L'évolution vers Spring Boot

- **Même si Spring Framework est puissant, sa configuration XML est un considérée lourde (complexité de configuration):**
  - Fichiers XML énormes pour déclarer les beans.
  - Configuration manuelle des datasources, du serveur web, des transactions.
  - Ajout manuel de toutes les dépendances dans le pom.xml (ou build.gradle).
  - Besoin d'un serveur externe (Tomcat, JBoss, Glassfish…) pour déployer l'application.
- **Spring Boot (apparu en 2014) est venu simplifier cela en introduisant :**
  - ✓ une onfiguration automatique (auto-configuration).
  - ✓ Un démarrage rapide avec un serveur intégré (Tomcat/Jetty embarqué).
  - ✓ Un principe de « Convention over configuration ».
  - ✓ Des Fichiers de configuration simples (application.properties ou application.yml).

- **Spring Boot ne remplace pas Spring, mais une extension qui le simplifie et l'automatise.**
- **Spring Boot est un modèle préconfiguré pour développer et exécuter des applications basées sur Spring.**

→ **Objectif : rendre Spring plus simple, plus rapide et plus productif.**

Caractéristiques principales de Spring Boot

1. Auto-configuration:

- Spring Boot détecte automatiquement les dépendances
- Exemple: si le développeur ajoute spring-boot-starter-data-jpa, il configure JPA, Hibernate et DataSource tout seul).

2. Starters:

- Au lieu d'ajouter manuellement des dizaines de dépendances, Spring Boot propose des "starters" (paquets de dépendances prédéfinis). Exemple:
  - spring-boot-starter-web → Tomcat + Spring MVC + JSON (Jackson).
  - spring-boot-starter-data-jpa → Hibernate + JPA + transactions.
  - spring-boot-starter-security → Spring Security préconfiguré.

3. Serveur embarqué:

- Plus besoin d'installer Tomcat ou Jetty séparément : Spring Boot utilise un serveur embarqué (Tomcat par défaut)

4. Convention over configuration:

- Spring Boot suit des valeurs par défaut intelligentes.
- Il est possible de les modifier dans application.properties ou application.yml si nécessaire.

5. Microservices Ready:

- Avec Spring Boot + Spring Cloud → développement simplifié d'applications distribuées (microservices).