

# MODULE Spring

## Programmation orientée aspect

# Spring - AOP

## Plan du module Spring - AOP

- Introduction
- 
-

# Spring - AOP

## Introduction

# AOP - Introduction

La « programmation orientée aspect » est un paradigme de programmation visant à améliorer la modularité d'un logiciel en découplant les services « transversaux » des services fonctionnels.

Les services transversaux sont ceux qui peuvent être utilisés par tous les services fonctionnels, mais sans rapport direct avec une fonctionnalité. On parle parfois de « services techniques ».

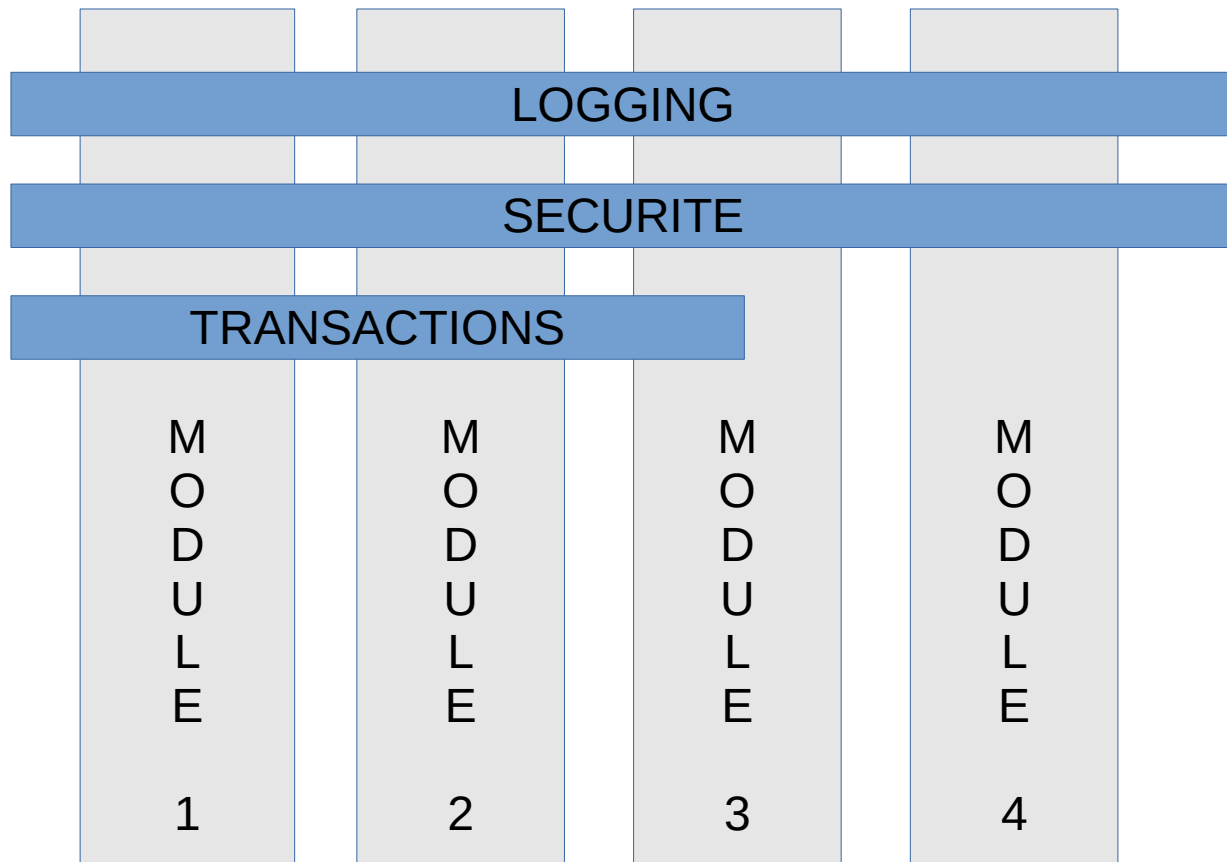
# AOP - Introduction

Exemples de services transversaux :

- L'enregistrement de logs
- La gestion des transactions
- La gestion de la sécurité
- ...

Il s'agit de services sans rapport direct avec les fonctionnalités du logiciel. Cependant, toutes les fonctionnalités ont (potentiellement) besoin de ces services.

# AOP - Introduction



# AOP - Introduction

## Concepts AOP

### 1. L'**aspect**

- Il s'agit d'un module réalisant un service transversal

### 2. L'**advice**

- Il s'agit d'un traitement simple réalisé par un service transversal

### 3. Le **joinpoint** (point de jonction)

- Il s'agit d'un point (un moment) durant l'exécution d'un programme, comme l'exécution d'une méthode ou le traitement d'une exception

# AOP - Introduction

## Concepts AOP

### 4. Le **pointcut**

- Une expression ou un prédicat qui correspond à (“matche”) un joinpoint

Un advice fait partie d'un aspect, est associé à un pointcut et s'exécute à chaque joinpoint correspondant à ce pointcut.



# AOP - Introduction

