

# TD2 architecture logicielle

Dr. Ghazala Hcini

31/10/2025

## 1 Objectif du TD

Concevoir et documenter une architecture logicielle pour une plateforme E-commerce en utilisant le modèle "4+1 vues" de Kruchten. L'accent doit être mis sur l'organisation, les performances, la scalabilité et le découplage des composants, des qualités essentielles dans le secteur du commerce en ligne.

## 2 Conception Architecturale d'une Plateforme E-commerce (Modèle 4+1 Vues)

Contexte : Vous êtes l'architecte logiciel chargé de concevoir la plateforme de commerce électronique "ShopFast". Le système doit gérer un catalogue, les comptes clients, les commandes, et l'inventaire. Le facteur critique est la capacité du système à absorber des pics de charge importants (scalabilité et performance).

- Approche Requise : La conception doit être structurée en utilisant le modèle "4+1 vues" de Kruchten.

### 2.1 Vue des Cas d'Utilisation (Pré-requis)

- Modules Métier Clés : Catalogue, Panier & Commandes, Authentification & Utilisateurs, Paiement (via service externe), et Inventaire (via service externe).
- Scénario Critique (Performance) : "Consultation Massive du Catalogue" (nécessite une réponse rapide et une haute disponibilité).
- Scénario Critique (Fiabilité) : "Validation et Paiement de Commande" (nécessite une séquence fiable entre la réservation de stock et la confirmation du paiement).

## 2.2 Tâches de Conception Architecturale (Les 4 Vues)

L'objectif est de détailler et de justifier les choix architecturaux pour chacune des 4 vues principales :

### 1. Vue Logique:

Décrivez l'organisation interne du système en composants logiques, en mettant l'accent sur les principes de qualité.

- Tâche 1.1 : Découplage par Interface.

- Modélisez l'interaction entre le module de haut niveau Panier & Commandes (l'appelant) et le module de bas niveau Paiement (l'appelée).
- Expliquez et justifiez pourquoi l'utilisation d'une Interface est fondamentale à cet endroit. Comment cela permet-il de changer l'implémentation (ex: passer de PayPal à Stripe) sans impacter le code de l'appelant?

- Tâche 1.2 : Cohésion et Couplage.

- Proposez une organisation pour le module Panier & Commandes qui maximise la forte cohésion et minimise le couplage faible avec les autres modules.

### 2. Vue des Processus:

Décrivez les interactions entre les processus et les threads pour garantir les contraintes de performance et de fiabilité.

- Tâche 2.1 : Inversion de Dépendance (DIP).

- Expliquez l'intérêt d'appliquer l'Inversion de Dépendance (DIP) entre le module de haut niveau ValidationCommande et un module de bas niveau ServiceInventaire.
- Montrez, à l'aide des schémas de packaging , comment cela permet de ne packager, builder et déployer que le module de bas niveau (ServiceInventaire) s'il est modifié, sans toucher au module de haut niveau (ValidationCommande).

- Tâche 2.2 : Scalabilité et Performance.

- Pour le Scénario Critique "Consultation Massive du Catalogue", décrivez le comportement temporel attendu et les mesures architecturales qui doivent être prises (sans parler encore de matériel) pour garantir une performance optimale et une faible latence.

### 3. Vue de Réalisation:

Décrivez l'organisation des composants dans l'environnement de développement (paquetages, bibliothèques).

- Tâche 3.1 : Choix de Granularité.

- Justifiez l'adoption d'une Architecture Service-Based (3 à 12 services) plutôt qu'une Architecture Microservices (plusieurs centaines de services) pour "ShopFast", en tenant compte de la gestion de projet et de l'objectif d'éviter les interconnexions inutiles.
  - Comment ce choix impacte-t-il la définition d'un Service (granularité) dans votre système?
  - Tâche 3.2 : Modularité et Réutilisation.
    - Proposez un découpage logique en paquetages/bibliothèques (Module.A, Module.B, etc.) et expliquez comment cette structure facilite la réutilisation de certaines parties de conception et l'évolution future du système.
4. Vue de Déploiement:
- Représentez le système dans son environnement physique d'exécution et gérez les contraintes non fonctionnelles.
- Tâche 4.1 : Stratégie de Mise à l'Échelle.
    - Pour garantir la Scalabilité du service face aux pics de charge (Noël, promotions), choisissez et justifiez entre le Horizontal Scaling (scale out : ajouter des nœuds supplémentaires) et le Vertical Scaling (scale up : renforcer le matériel).
    - Dessinez un schéma simple de déploiement qui montre l'application de cette stratégie pour le module Catalogue .
  - Tâche 4.2 : Interopérabilité et Évolutivité.
    - Décrivez comment l'élasticité du système (capacité à s'adapter automatiquement aux demandes) pourrait être gérée pour les services d'hébergement du site, en citant les aspects cruciaux (vitesse et précision).