

# Comprendre le Domain-Driven Design (DDD) Construire des applications ancrées dans le métier

Le DDD n'est pas une architecture, mais une façon de penser et modéliser les règles métier au cœur de votre application.

Voici les clés pour bien démarrer !



# Pourquoi le DDD ?

## Quand la complexité métier devient un casse-tête



Services qui deviennent illisibles



Règles métiers dispersées



Code métier dépendant des technos



Et surtout, une logique métier difficile à maintenir



# Les piliers du DDD

## **Avant de coder, comprendre les concepts**


- ✓ Un langage commun (Ubiquitous Language)
- ✓ Une modélisation métier alignée avec les experts fonctionnels
- ✓ Des Bounded Contexts pour découper clairement le métier
- ✓ Des entités, agrégats, value objects, services, repositories...



# Le modèle du domaine

## Un cœur métier autonome et pur

- ✓ Les règles métier
- ✓ Les entités (avec identité)
- ✓ Les agrégats (invariants)
- ✓ Les Value Objects (immuables)

 Aucune dépendance à Spring, JPA, Web, etc.



# Services du domaine

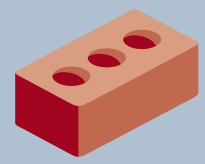
## Pour la logique qui ne rentre pas dans une entité

Quand une règle métier implique plusieurs objets ou n'a pas d'état :

👉 Utilisez un ***DomainService***

🎯 Exemple :  
*ValidationDePaiementService*,  
*CalculateurDePrixLivraison*





# Infrastructure et Application

## Comment connecter votre domaine au reste du monde

L'application orchestre les use cases

L'infrastructure s'occupe des technos (JPA, API, MQ...)

👉 Le domaine reste “aveugle” à tout ça

➡ On inverse les dépendances avec des interfaces





# Exemple concret

## Supprimer un utilisateur sans commande active

- ✓ UserService (dans le domaine)
- ✓ Dépend d'un OrderChecker (interface métier)
- ✓ Implémenté en infra via JPA
- ➡ Le métier exprime l'intention, l'infra exécute



# Checklist express

## DDD

### Pose-toi ces questions pour bien démarrer


- ✓ As-tu un langage commun avec le métier ?
- ✓ Tes règles métier sont-elles testables sans Spring ?
- ✓ Les entités manipulent-elles leurs invariants ?
- ✓ Le domaine connaît-il l'infrastructure ? (spoiler : non)







**? Et vous, vous  
utilisez déjà le DDD ?**

**Partagez vos retours  
d'expérience** 

 Vous avez une structure  
différente ?

 Vous avez rencontré des  
difficultés ?

 Dites-moi en commentaire  
comment vous modélisez votre  
domaine et vos cas métier !

