

DDD : RESPONSABILITÉ DES COMPOSANTS ET ANNOTATIONS SPRING

TEAM CONCEPTEUR ITG / SIT / PACT01



BNP PARIBAS

The bank for a changing world

La Team

Pilotage Architecture Cyber sécurité et Technologies – ITG_SIT



Julien Goullon
Adriana Niteescu
Jimmy Sedo Savi
Romain Noel

Domain**D**riven **D**esign

Responsabilité des Composants
et Annotations **Spring**

Le DDD

**c'est mettre les besoins métier
au cœur de tout**

Qui ?

Quoi ?

Comment ?

Qui ? Quels sont les **composants** mis en jeu ?

Quoi ? Quelles sont les **responsabilités** des composants ?

Comment ? Quelle implémentation ? 

1 composant = 1 responsabilité
= ?

1 composant = 1 responsabilité

=

**1 seule raison
de changer**

**1 couplage
faible**

**1 code facile
à comprendre**

HOWTO

Architecture en



?

Exposition

Application

Domaine

Infrastructure

Exposition

La porte ouverte
sur le monde extérieur



Application

L'orchestration des use cases



Domaine

Exposition

Infrastructure

Application

Domaine



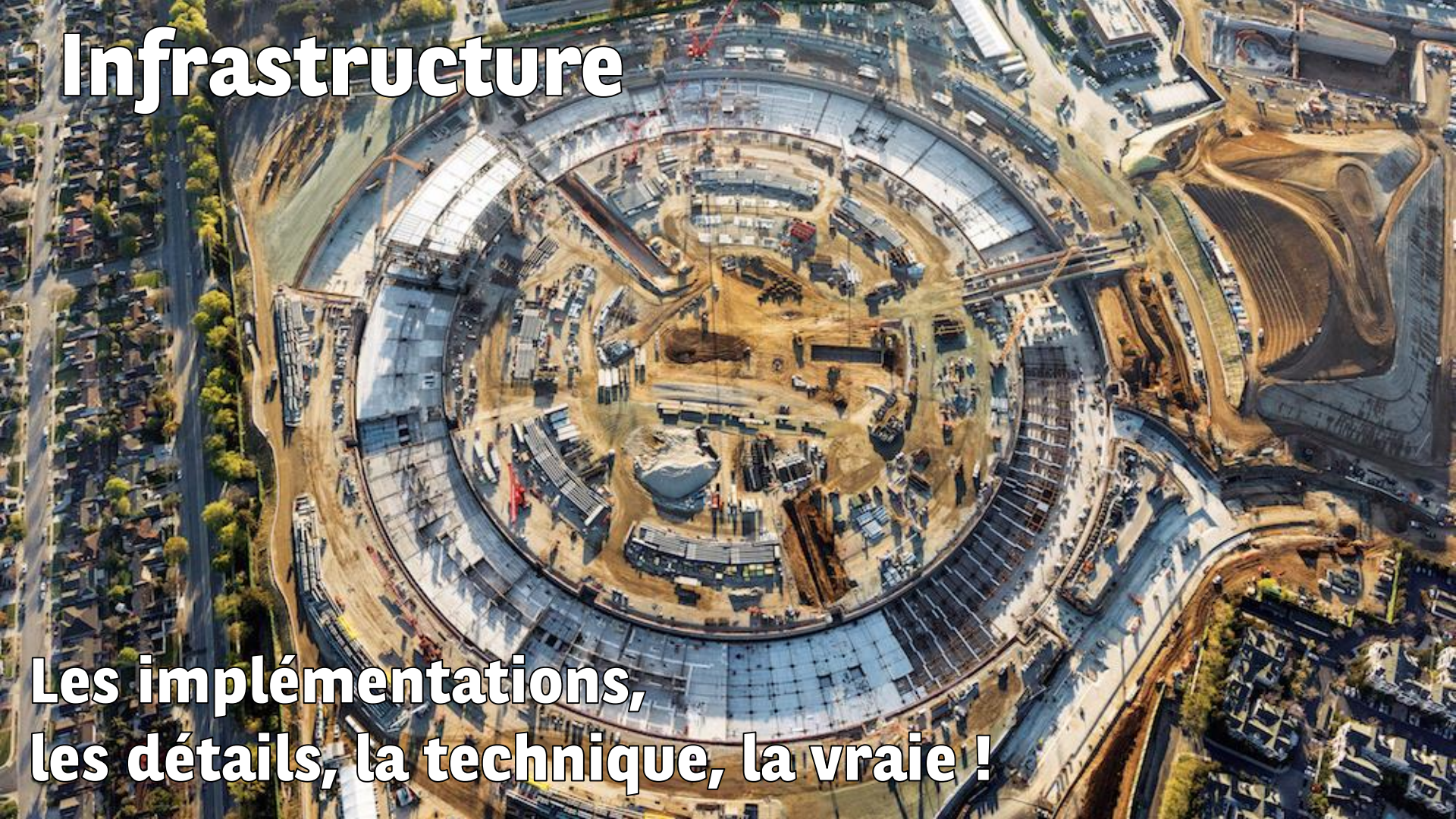
Framework

La logique métier
protégée du reste du monde

Domaine

Persistance

REST



Infrastructure

**Les implémentations,
les détails, la technique, la vraie !**

FACTORY
APPLICATION-SERVICE
CONTROLLER
ENTITY
DTO
DAO
BUILDER
VALUE-OBJECT
INTERFACE-SERVICE
DOMAIN-SERVICE

A close-up photograph of a woman with dark hair pulled back, wearing red lipstick and a black strap top. She is pointing her right index finger upwards and looking slightly to the side with a questioning or incredulous expression. The background is blurred with purple and blue tones.

NON MAIS ALLO QUOI ?!

Controller

Le réceptionniste

Responsabilité : Réceptionner les requêtes extérieures

Habitat naturel : couche Exposition

Déclinaisons : contrôleurs REST, batch steps, ...

DTO = Data Transfert Object **Le transporteur**

Responsabilité : Faire transiter les données depuis et vers l'extérieur

Habitat naturel : couche Exposition



Le domaine reste protégé du reste du monde

Un DTO n'est pas un objet du domaine

Application Service **Le chef d'orchestre**

Responsabilité: Implémenter des use cases,
Coordination spécifique au use case
Déclarer les préoccupations techniques

Habitat naturel : couche Application

Repository **Le collectionneur**

Responsabilité: Ensemble requêtable d'éléments d'un type donné

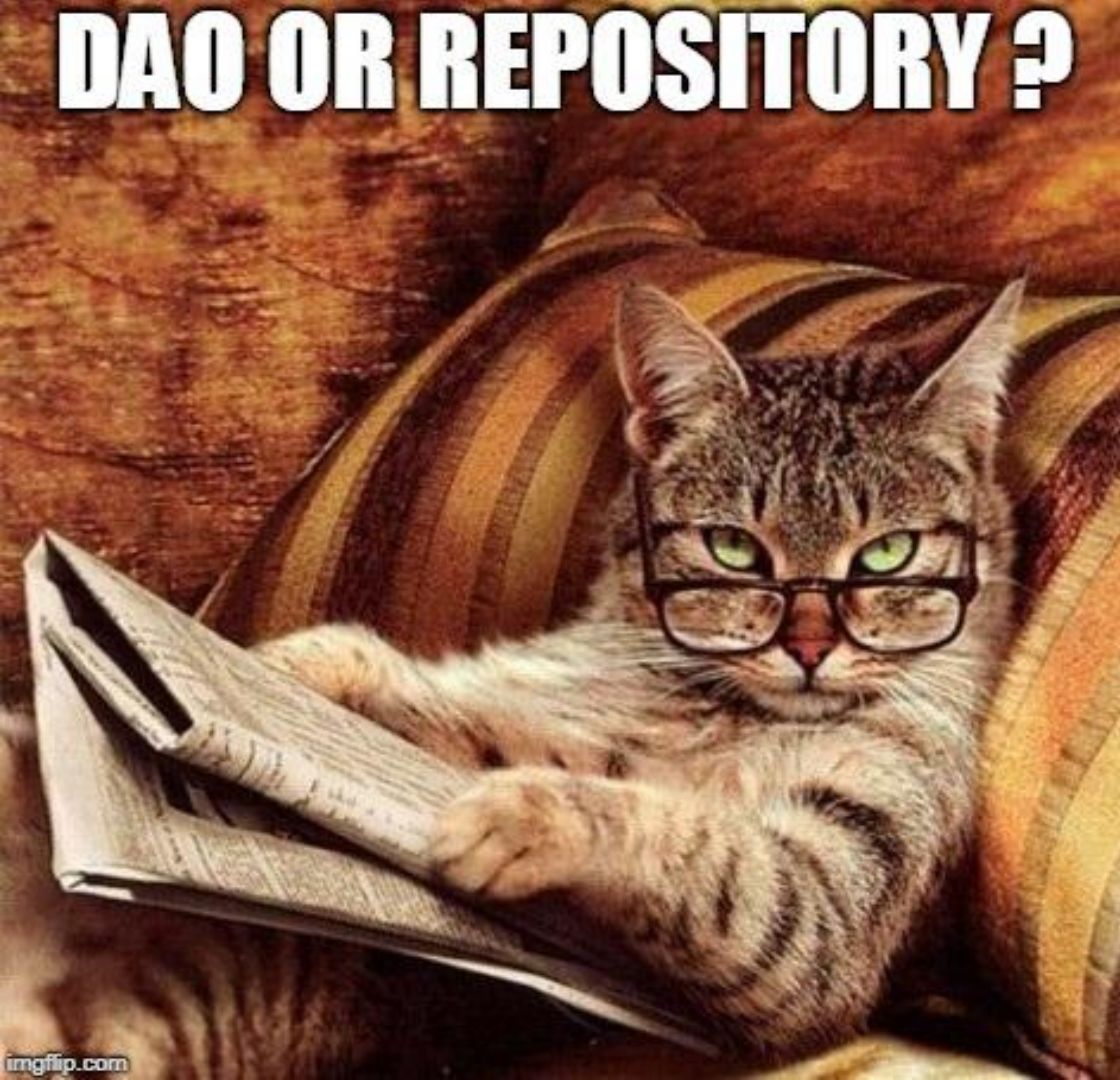
Habitat naturel : couche Domaine

Attention, seul l'interface est dans le domaine



L'implémentation habite dans l'infrastructure

DAO OR REPOSITORY ?



DAO

Composant technique
Sémantique technique

Repository

Élément du domaine
Sémantique métier

Interface Service **Le fournisseur**

Responsabilité: Offrir des services purement techniques pour répondre aux besoins métier

Habitat naturel : couche Domaine



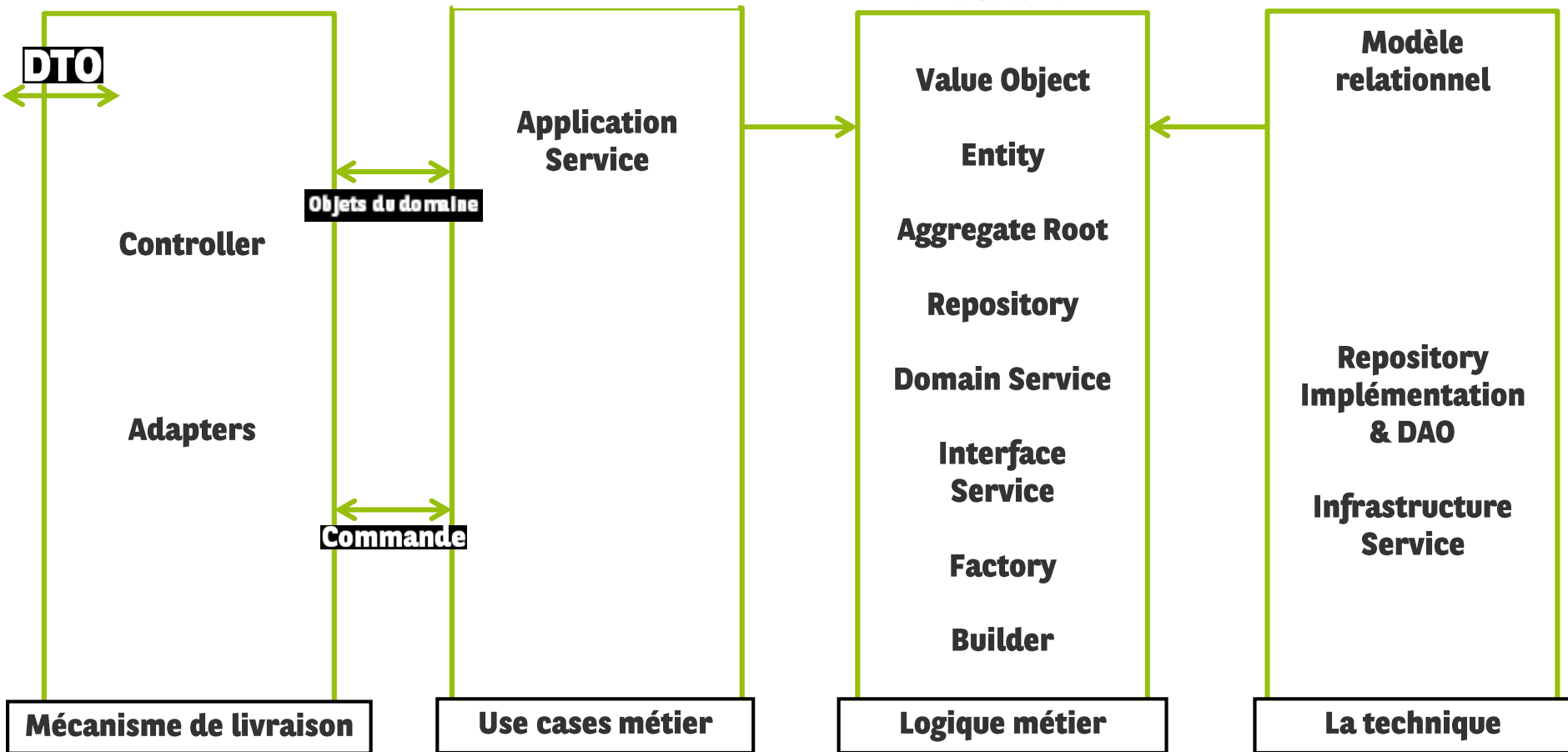
Ici aussi, seul l'interface est dans le domaine !

EXPOSITION

APPLICATION

DOMAIN

INFRASTRUCTURE



SHOW ME YOUR CODE

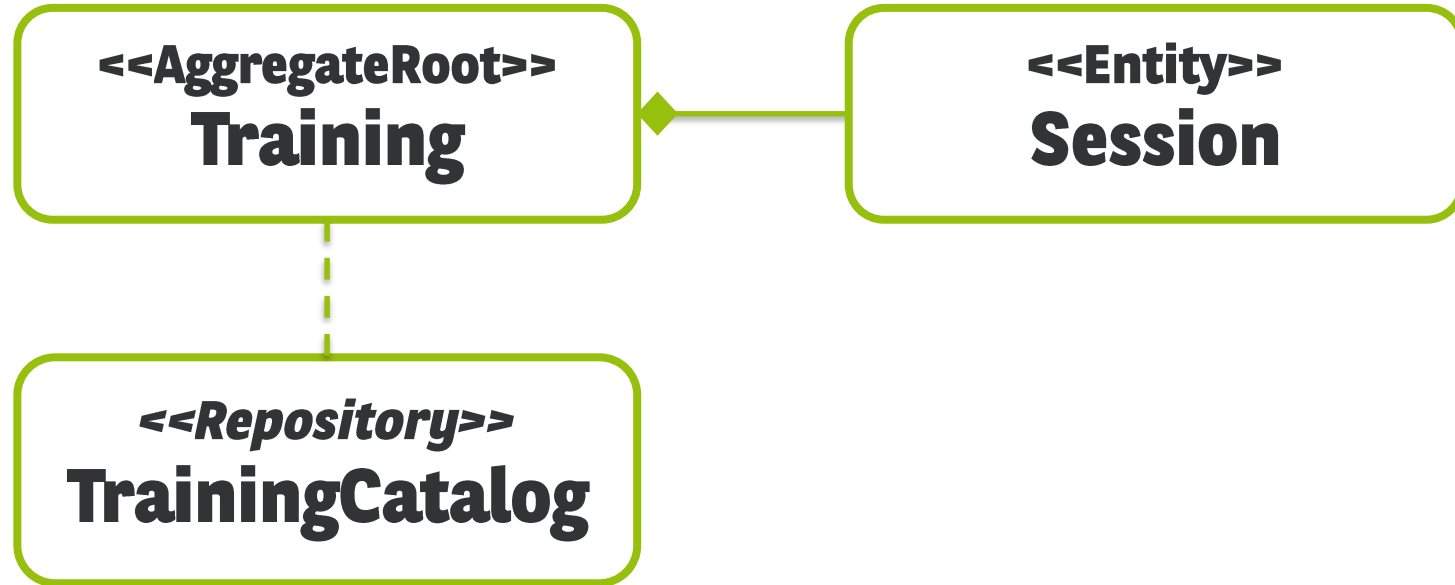
SHOW ME YOUR CODE



En tant que Collaborateur

Je veux m'inscrire à une session de formation
organisée par le dojo

Afin d'améliorer mes connaissances sur un
sujet particulier



***Nowadays
in a **Spring** galaxy, not so far away...***



Controller

@RestController

- + paramètres: @RequestParam / @PathVariable
- + validation: @Validated (javax.validation)

EXPOSITION

Application Service

@Service

@Transactional (en tant qu'initiateur)

@Secured

APPLICATION

Domaine



0 % framework
0 % leak technique
100 % métier

Clean DDD

Repository

@Repository

@Transactional (en tant que participant)

@Entity ≠ Entité(DDD)

**@Repository ≠
Repository(DDD)**

- ❖ Utilise les **@Entity** (modèle relationnel)
- ❖ Charge & persiste les données via **JpaRepository<@Entity>**
- ❖ Adapte les **@Entity** aux entités du domaine (modèle métier)

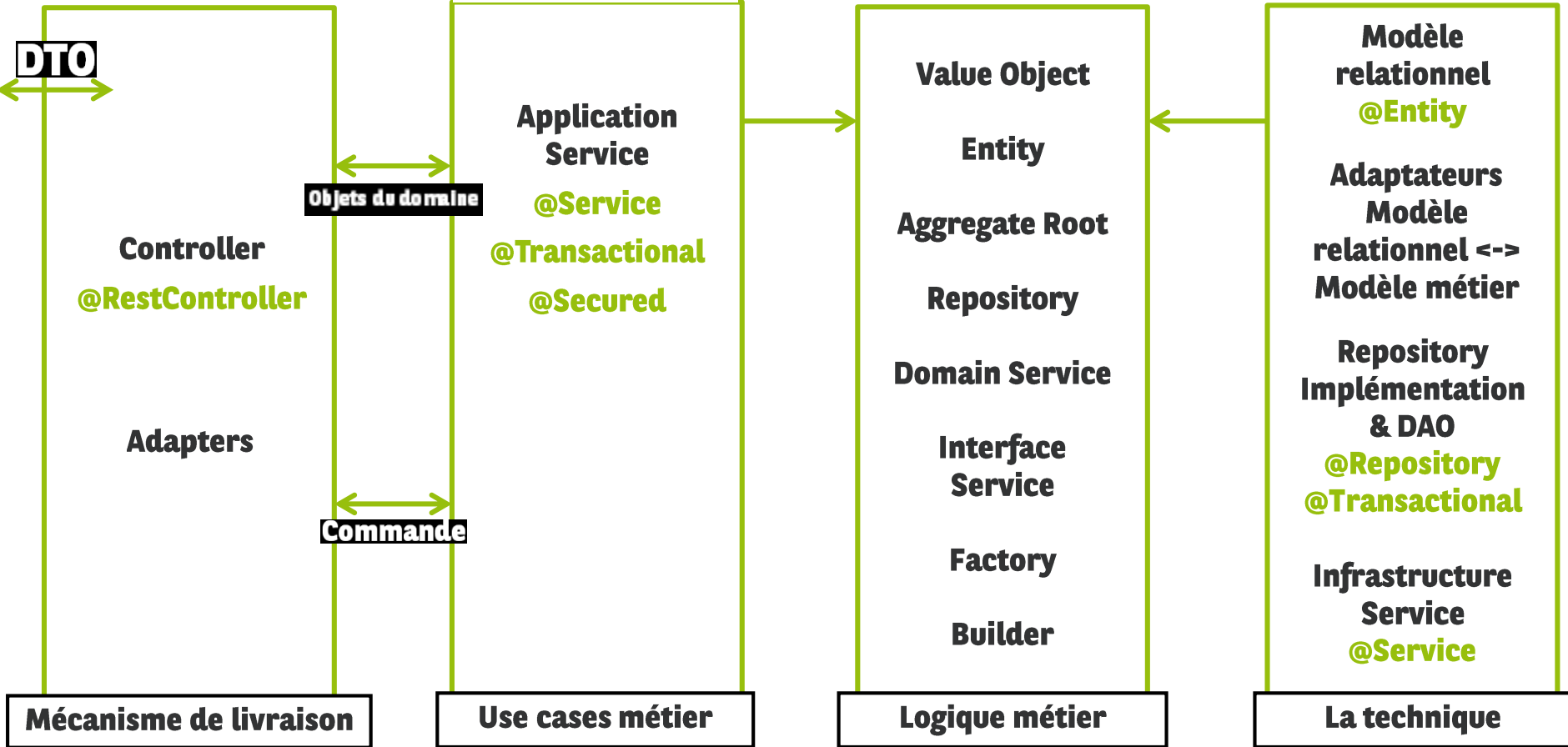
INFRASTRUCTURE

EXPOSITION

APPLICATION

DOMAIN

INFRASTRUCTURE



DÉMO



Nous avons utilisé



**Spring Core , Spring boot 2, Spring MVC,
Spring Data JPA, Spring dev-tools**

Nous aurions pu utiliser

ObjectMapper, OrikaMapper, MapStruct, bean-utils



DDD
+
Spring
=





That's all Folks!



**DON'T FEED
THE TROLL**



Les **@Entity** JPA dans le domaine

Domaine + @Entity = ?



**Où est la frontière des effets
de bords ?**

DO NOT CROSS

