

# OPEN WORLD FORUM

*Développer un projet  
fonctionnellement complexe*

# *En face de vous*

- Jean-François Lépine
- Consultant technique chez Alter Way
- @Halleck45
- <http://blog.lepine.pro>



# *Jeu concours : qui a déjà entendu ...*

*“Techniquement et fonctionnellement, c'est super !  
On est même largement en avance...”*

*“Génial ! C'est une chance que vous soyez  
télépathe ; c'est exactement ce à quoi je pensais,  
il n'y a rien à reprendre.*

*“Ah oui, je me souviens bien : c'est en page 28 de  
l'annexe 2 des spécifications fonctionnelles  
détaillées”*

# *Il était une fois...*

- “J'ai besoin d'une voiture pour des petits trajets, aller au travail et transporter ma famille”

# *Il était une fois...*

- “J'ai besoin d'une voiture pour des petits trajets, aller au travail et transporter ma famille”



- Le client n'a pas eu ce qu'il voulait



- Et en plus il a perdu de l'argent

*Communiquez*

# *Une langue commune*

- Éviter les ambiguïtés
- Est élaborée en commun
- Est parlée par tous
- Glossaire du projet



UBIQUITOUS  
LANGUAGE

# Une grammaire commune

Fonctionnalité: *acheter un chiot*

*En tant que client du magasin*

*Je peux acheter un chiot*

*Afin de avoir un compagnon  
tout mignon (ou pas ^^)*





# Gherkin

Fonctionnalité: *acheter un chiot*

*(...)*

Scénario: *acheter un chiot trop jeune*

*Etant donné qu'un chiot est trop jeune pour être vendu*

*Quand j'essaye de l'acheter*

*Alors je suis informé que c'est impossible*

# Développement piloté par le comportement

- Exercice agile quotidien
- Les scénarios sont établis en commun
- Tests automatisés ! (Behat, Cucumber, Jbehave...)
- Suivi en temps réel de l'avancée des scénarios

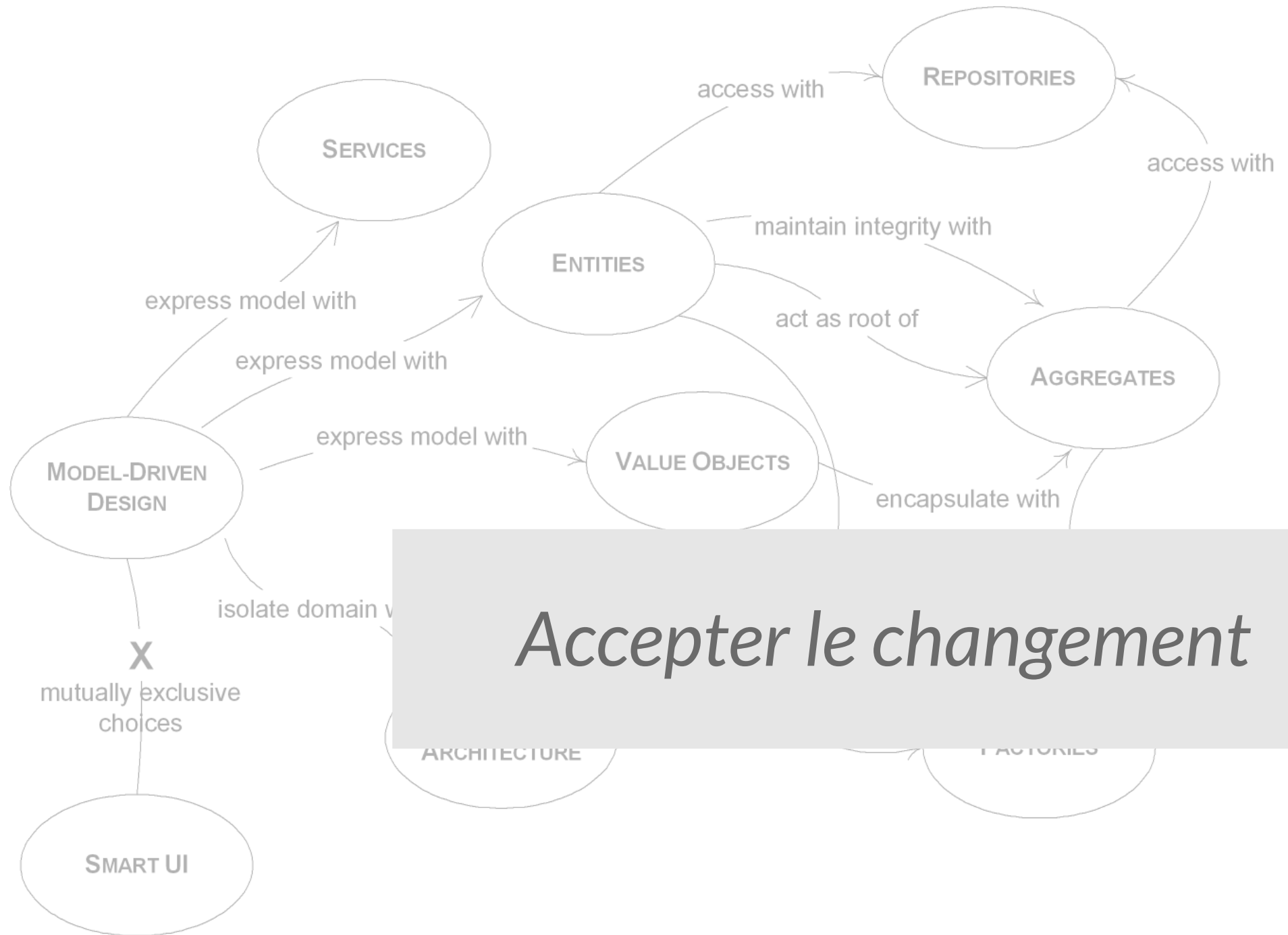
**My features**

Show: **Everything** Success Failures Pending

Sort by: **Name** Failures Pending

[+ New feature](#)

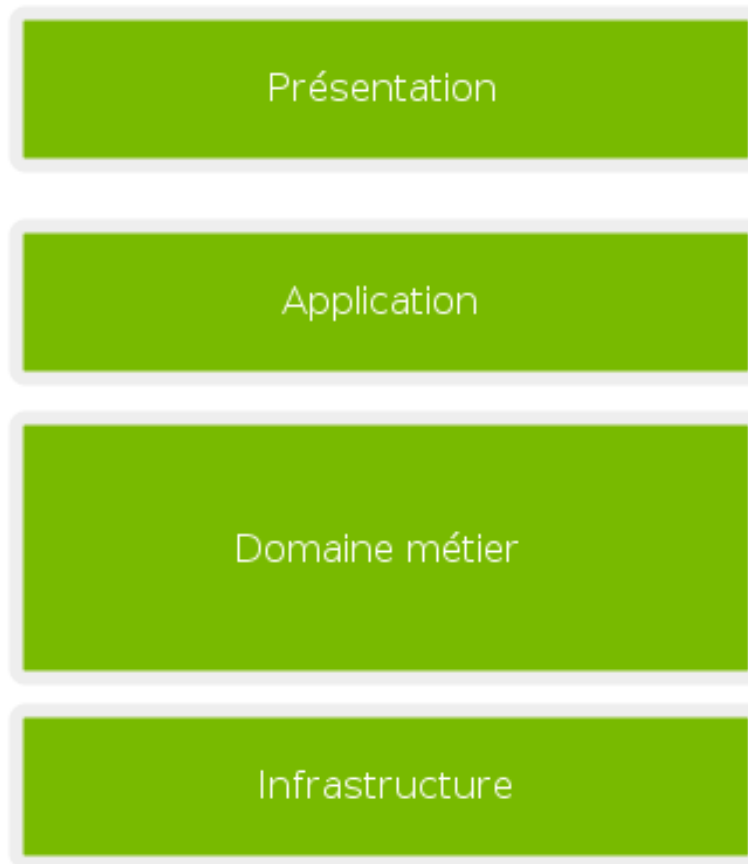
<p><b>❗ All fails</b> ( 0 / 0 / 4 / 4 ) In order to anything As as anything I should be able to do anything</p>	<p><b>✓ All is correct</b> ( 4 / 0 / 0 / 4 ) In order to anything As as anything I should be able to do anything</p>	<p><b>❗ Incomplete</b> ( 1 / 3 / 0 / 4 ) In order to anything As as anything I should be able to do anything</p>
<p><b>❗ It fails sometimes</b> ( 2 / 0 / 2 / 4 ) In order to anything As as anything I should be able to do anything</p>	<p><b>❗ Never tested</b> ( 0 / 0 / 0 / 0 ) In order to anything As as anything I should be able to do anything</p>	



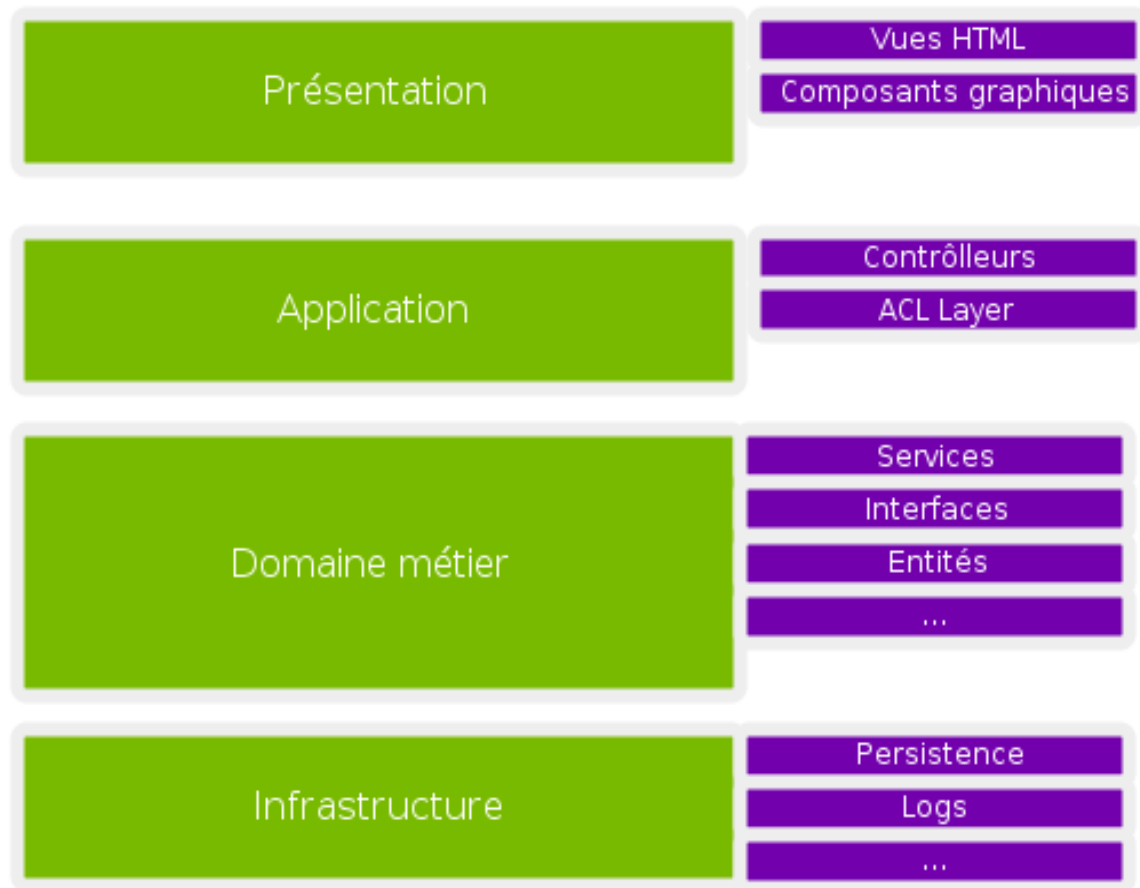
# Domain Driven Design

- Le **code source est le reflet du besoin fonctionnel**
- Le code source **adopte le vocabulaire fonctionnel**
- Le code source facilite :
  - *Le changement*
  - *La gestion des règles métiers*
  - *La maintenabilité*
  - *La testabilité*
- Appliquer à la lettre les principes de la programmation orientée objets

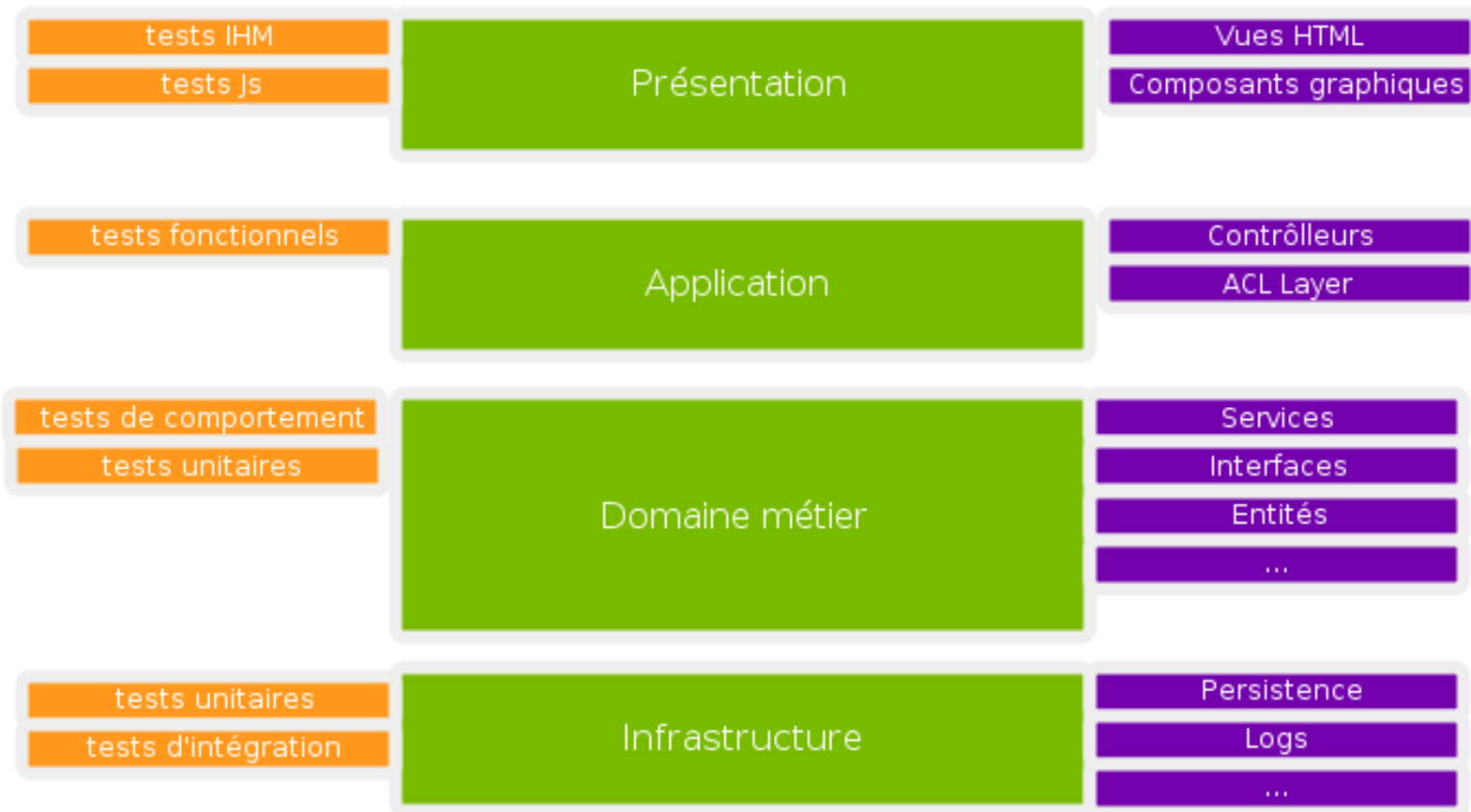
# *Isolation*



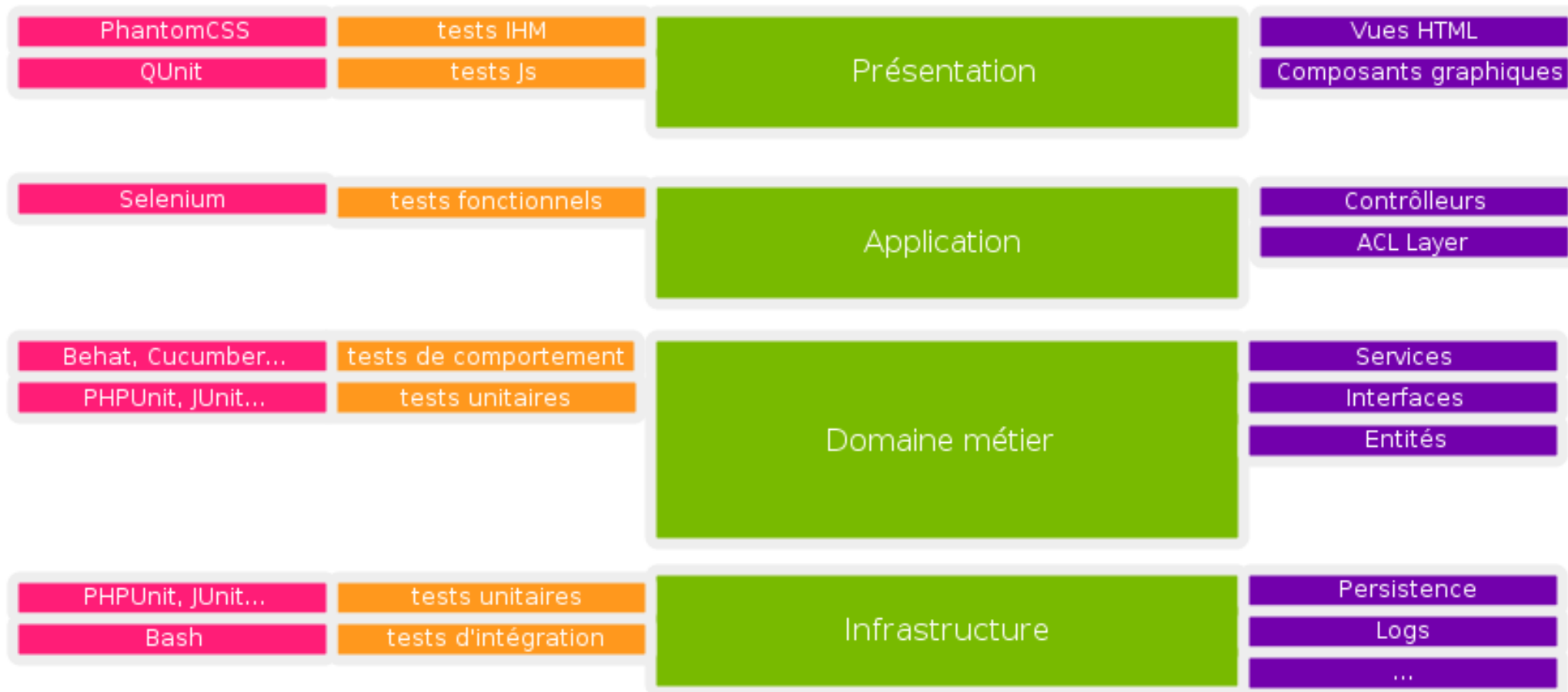
# Spécialisation



# Testabilité



# Testabilité





# *Keep It Simple and Stupid !*

- Un objet simple par règle métier
- La complexité se fait par l'assemblage de composants simples et réutilisables



# *Complexité vs performance*

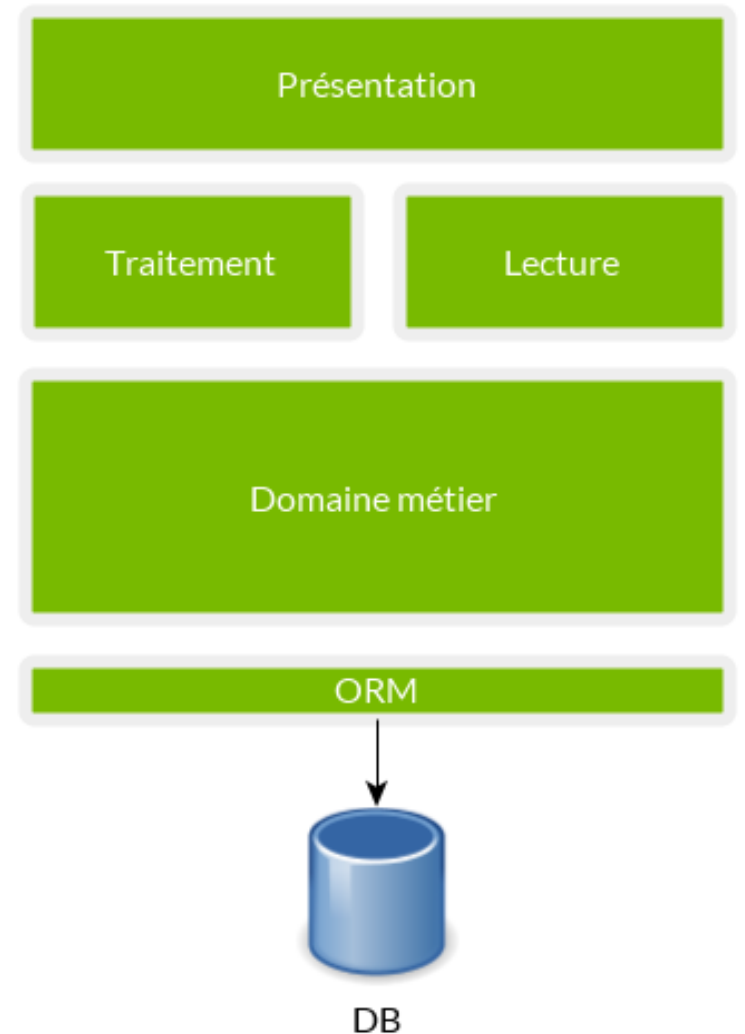
# Web = Schizophrène



- Effectuer un traitement métier
  - *Complexe*
  - *Lent*
  - *Intégrité et cohérence*
  - *A intégrer dans un système global*
- Fournir une information
  - *Immédiat*
  - *Spécifique*

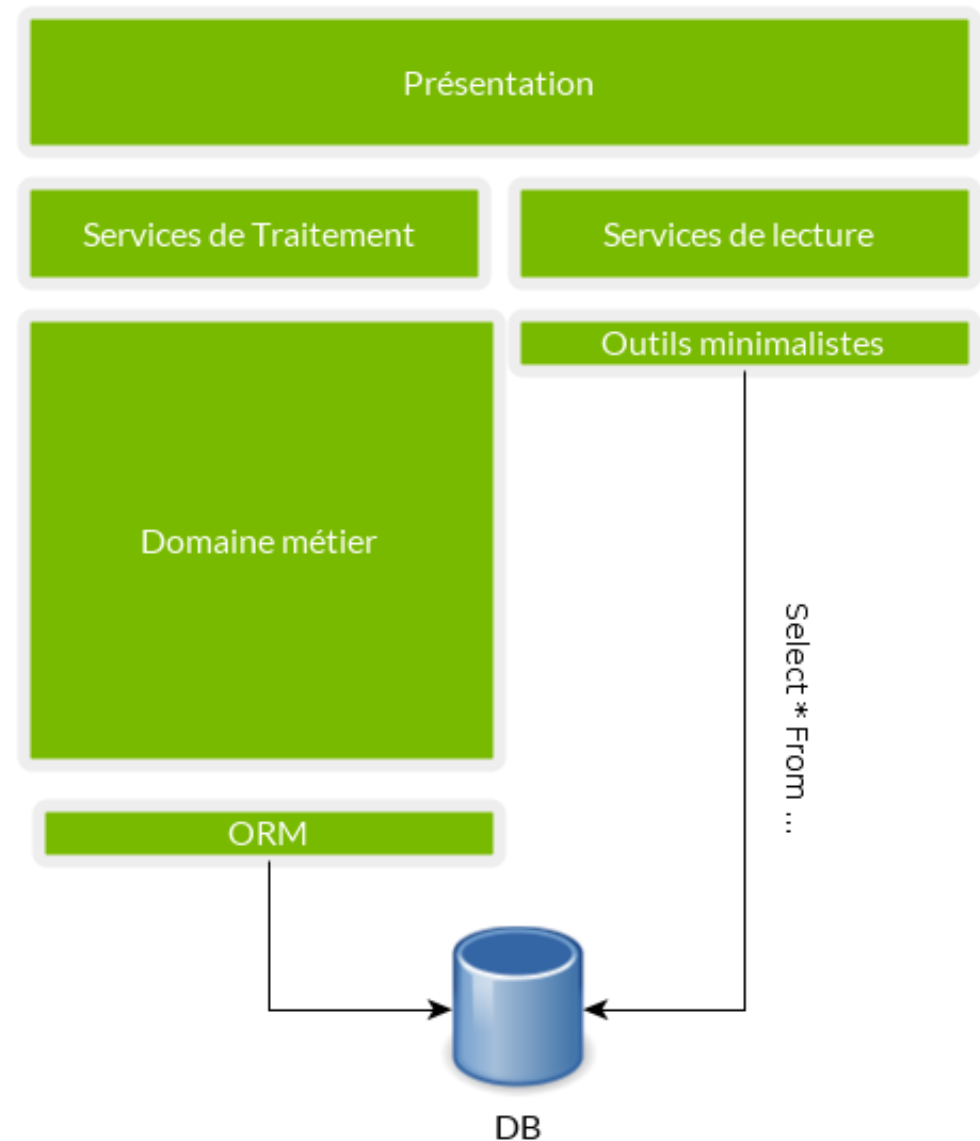
# Commands / Query

- Séparer la logique de traitement (Commands) de la logique de lecture (Query)
- Command Query Responsibility Segregation (CQRS)



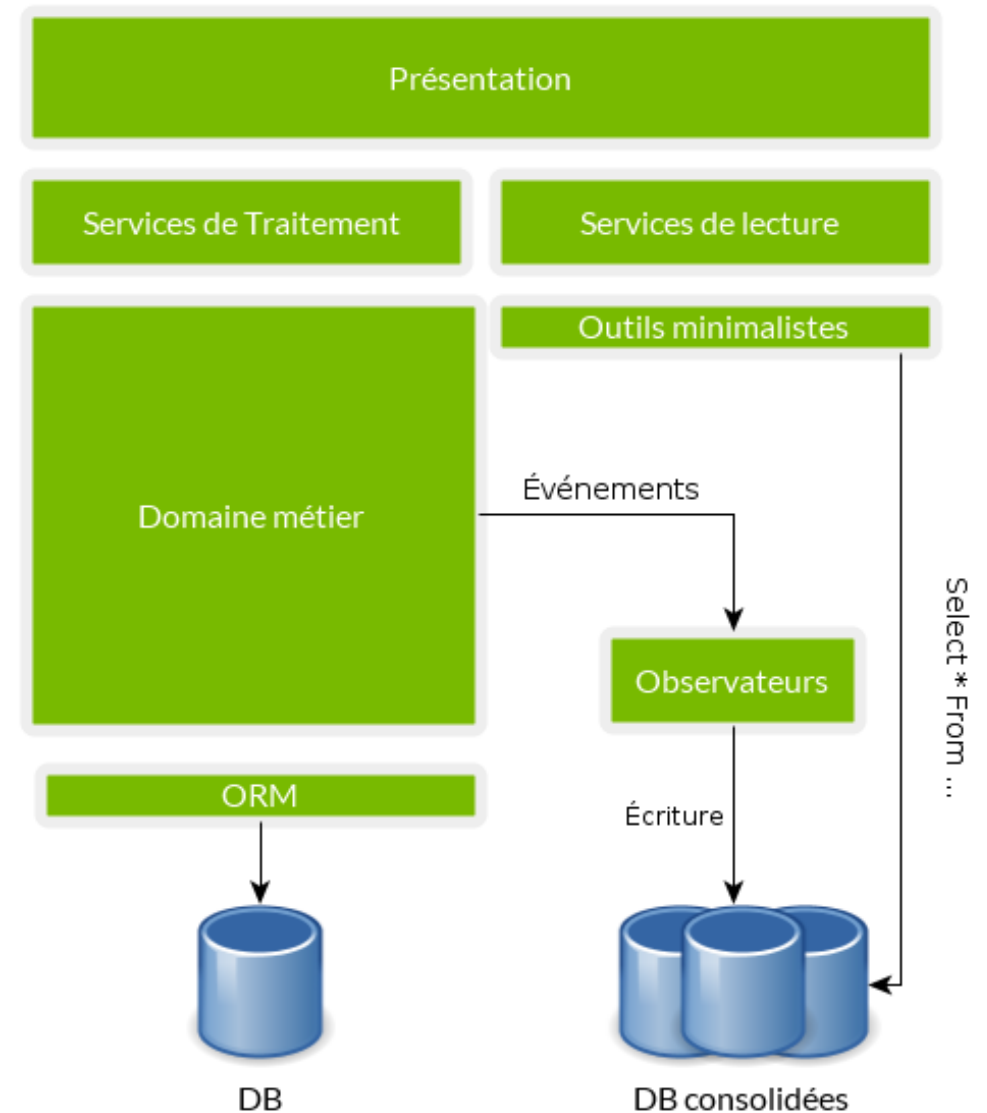
# Performance

- Alléger la lecture



# Consolidation

- Les données sont consolidées
- La consolidation est asynchrone
  - *Event sourcing*



# *Conclusion*

# Bilan

- Les solutions pour gérer de la complexité métier dans un projet existent :
  - *Communication (agile)*
  - *Développement piloté par le comportement*
  - *Domain Driven Design*
  - *Command query responsibility segregation*
- Certaines pratiques peuvent mises en place facilement, même si elles ne sont pas des recettes miracles



# Questions ?



- Twitter : @Halleck45
- <http://communiquiez.lepine.pro>
- Merci !