

# Les bases du TDD par l'exemple



Xavier NOPRE

Raise Partner - déc. 2019

Qui suis-je ?



# Xavier NOPRE

*Artisan-développeur & agiliste*



Indépendant : développement, formation, accompagnement, ...

Actuellement : CTO / Tech lead @ vrtice



xnopre@gmail.com

@xnopre



Et vous ?

*(Ex)* Développeur(euse) ?

Et vous ?

Tests unitaires ?

TDD ?

# TDD : la base pour faire du développement agile ?

Agilité ?



Développement itératif & incrémental



Refactoring

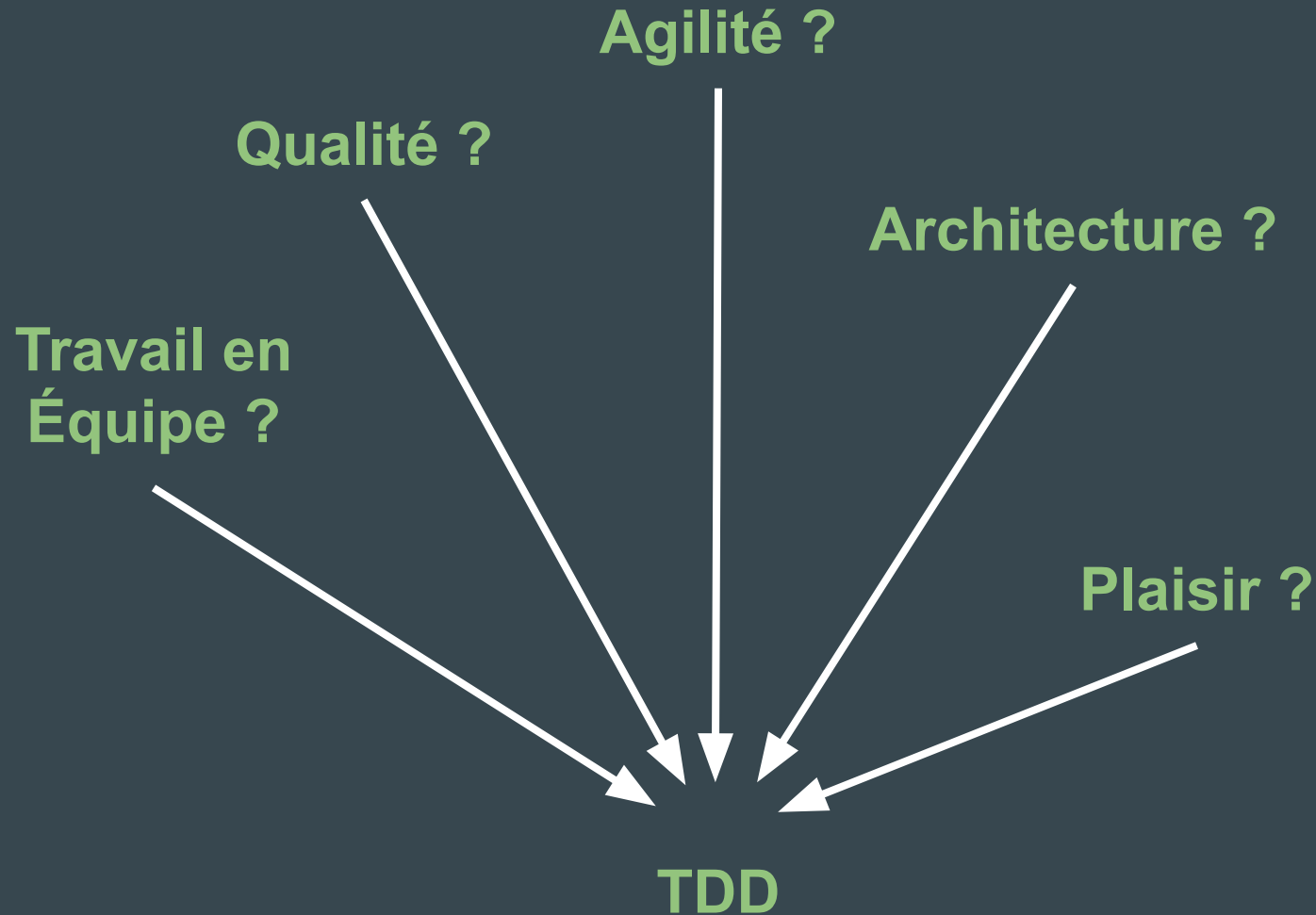


Tests unitaires



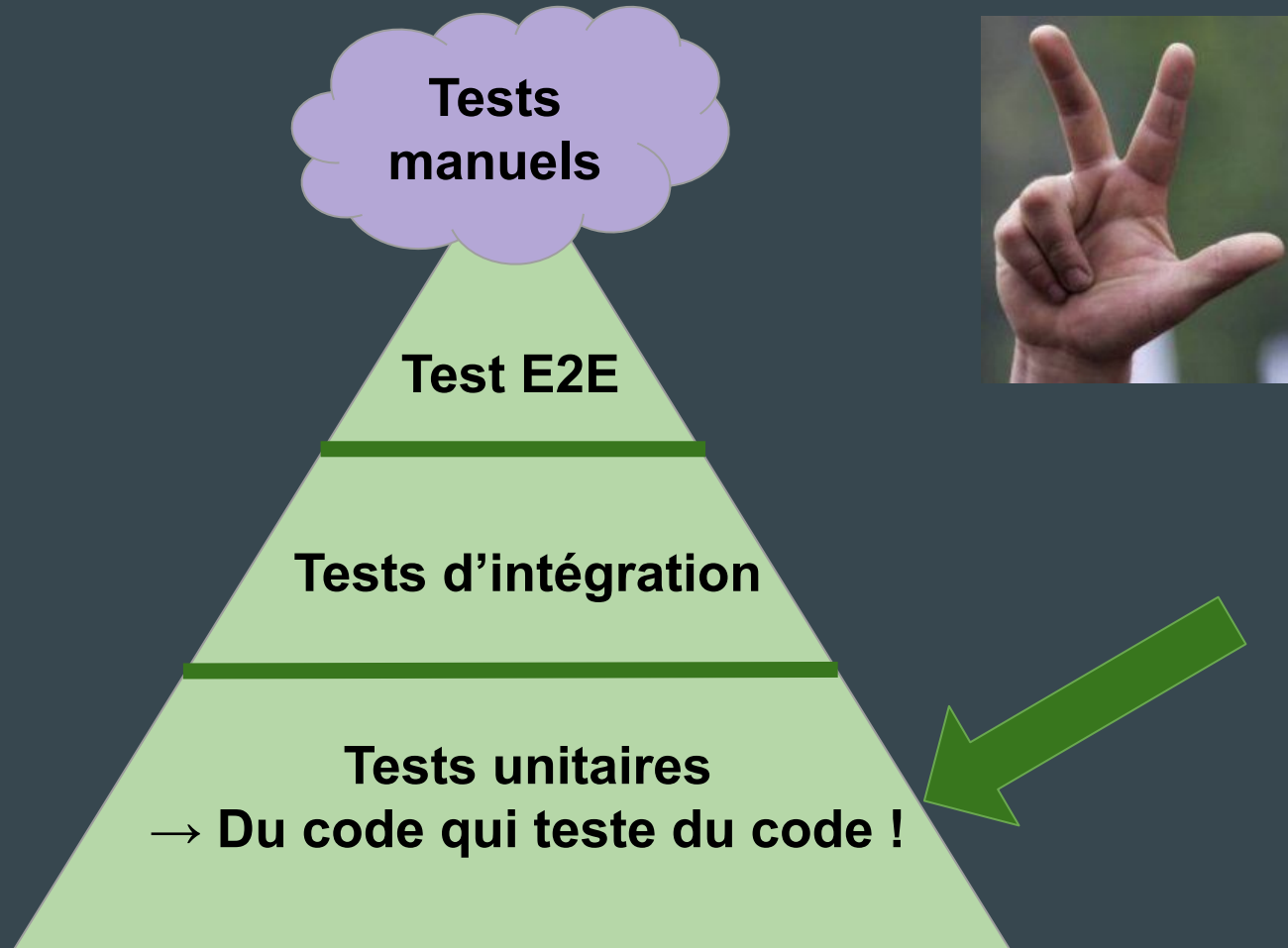
TDD

# TDD : la base pour faire du développement agile ?



# Tests unitaires

# Tests unitaires : c'est quoi ?



Pyramide des tests de Mike Cohn



# Tests unitaires : principes

- **Indépendants** (les uns des autres)
- **Répétables** (seuls ou en groupes)
- **Automatisables** → intégration continue
- **Rapides** (quelques secondes pour une classe de tests)
- **Lisibles** (code soigné autant que pour la prod)
- **Maintenables**

# Tests unitaires : règles

- 1 test = 1 cas (pas de scénarios)
- Se focaliser sur les situations courantes  
(puis traiter les cas d'erreurs)
- Structure : préparation - action - vérification(s)
- Maintenir les tests (ne jamais les supprimer)

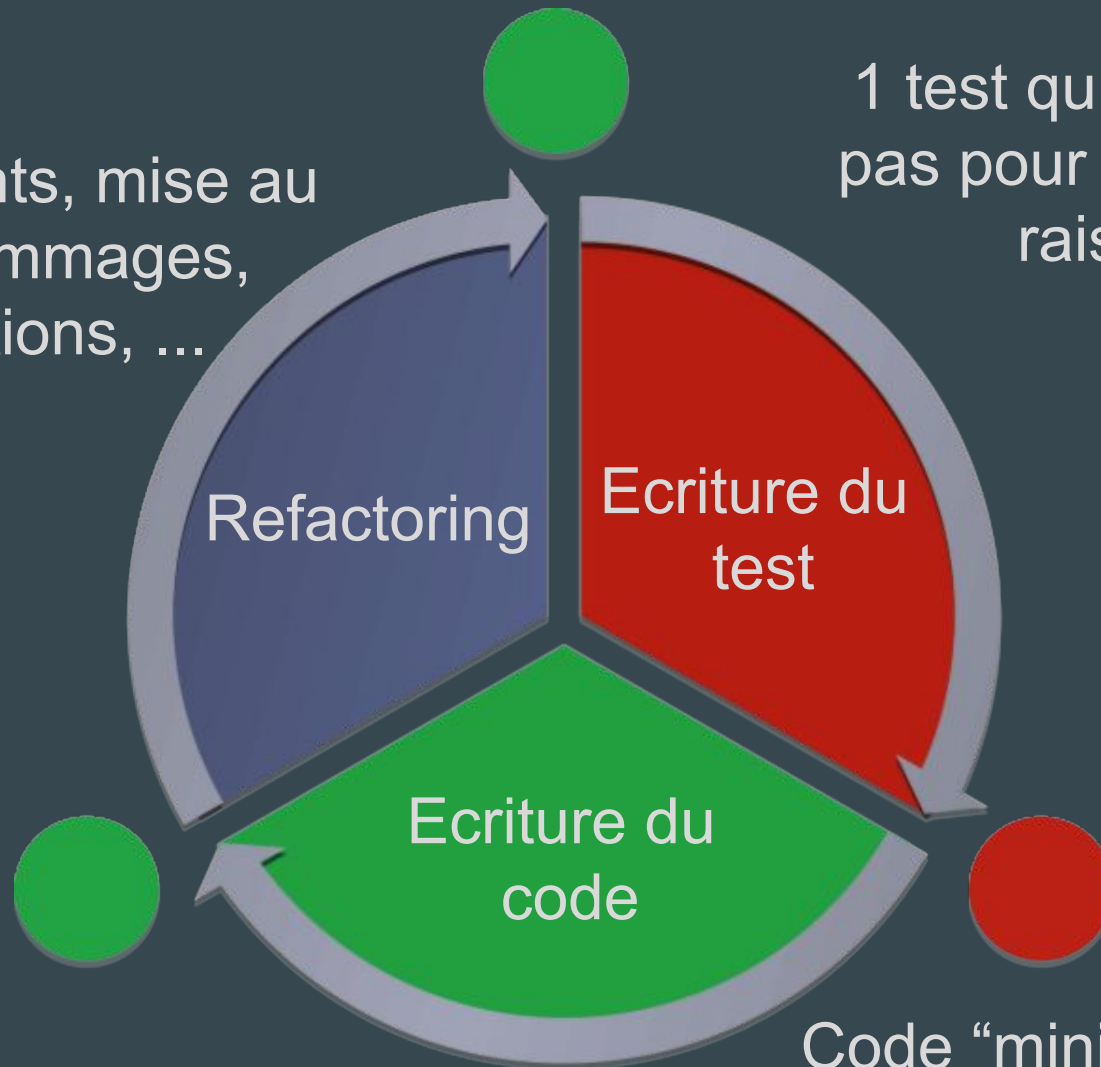
TDD

# TDD : c'est quoi ?

- TDD = “Test Driven Development”  
= “Développement piloté par les tests”
- C’est une **technique**, une façon de faire (≠ tests unitaires)
- Ecrire les tests d’abord... mais pas que...  
→ Etat d’esprit ...

# TDD : comment ?

Remaniements, mise au propre, nommages, factorisations, ...



1 test qui ne passe pas pour de bonnes raisons

Refactoring

Ecriture du test

Ecriture du code

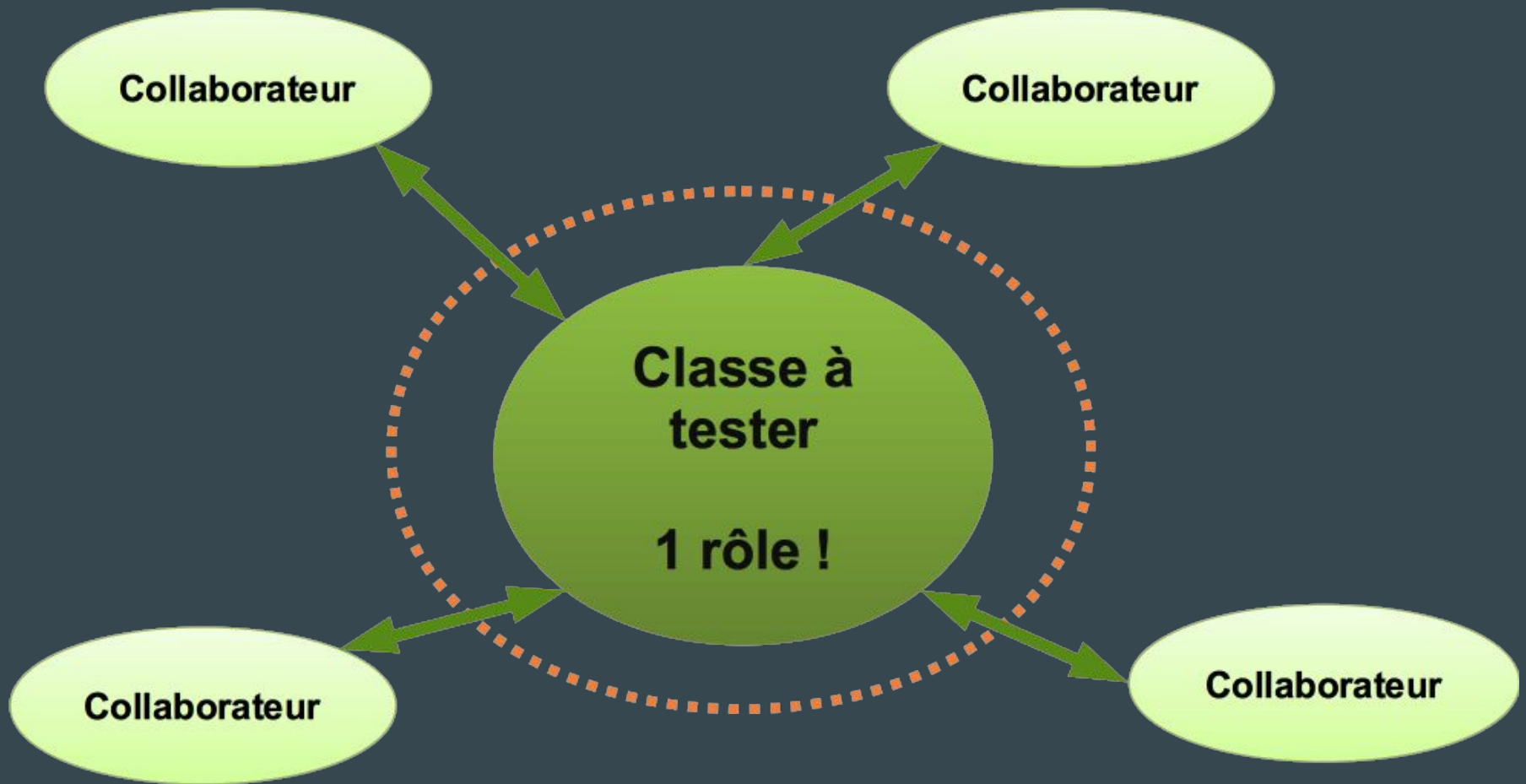
Code "minimum" pour faire passer ce test

# TDD : Focus

- “Quoi ?” (vs “comment ?”)
- Approche par l’usage

**Mocks**

# Mocks : principe 1/2

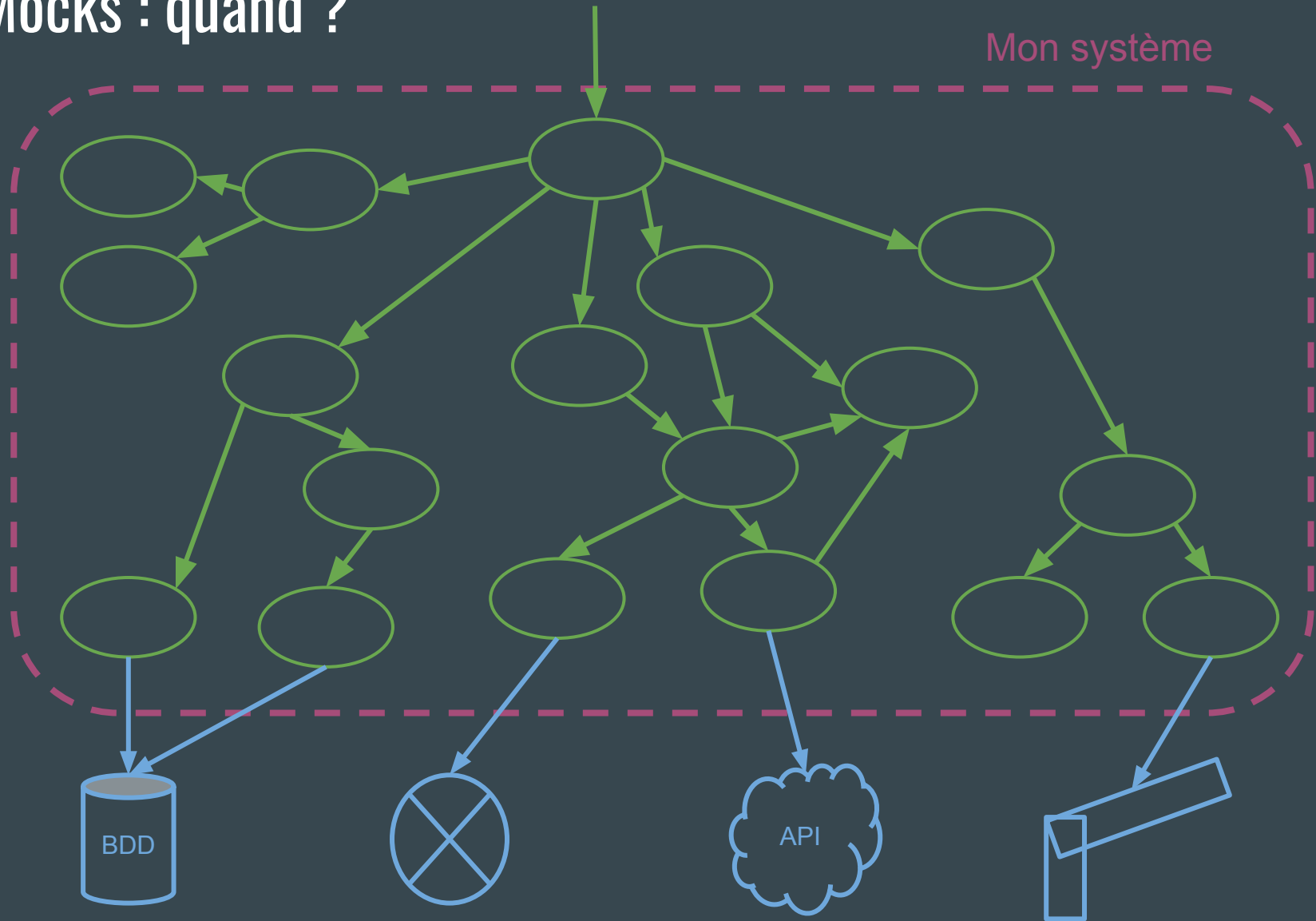




# Mocks : principe 2/2

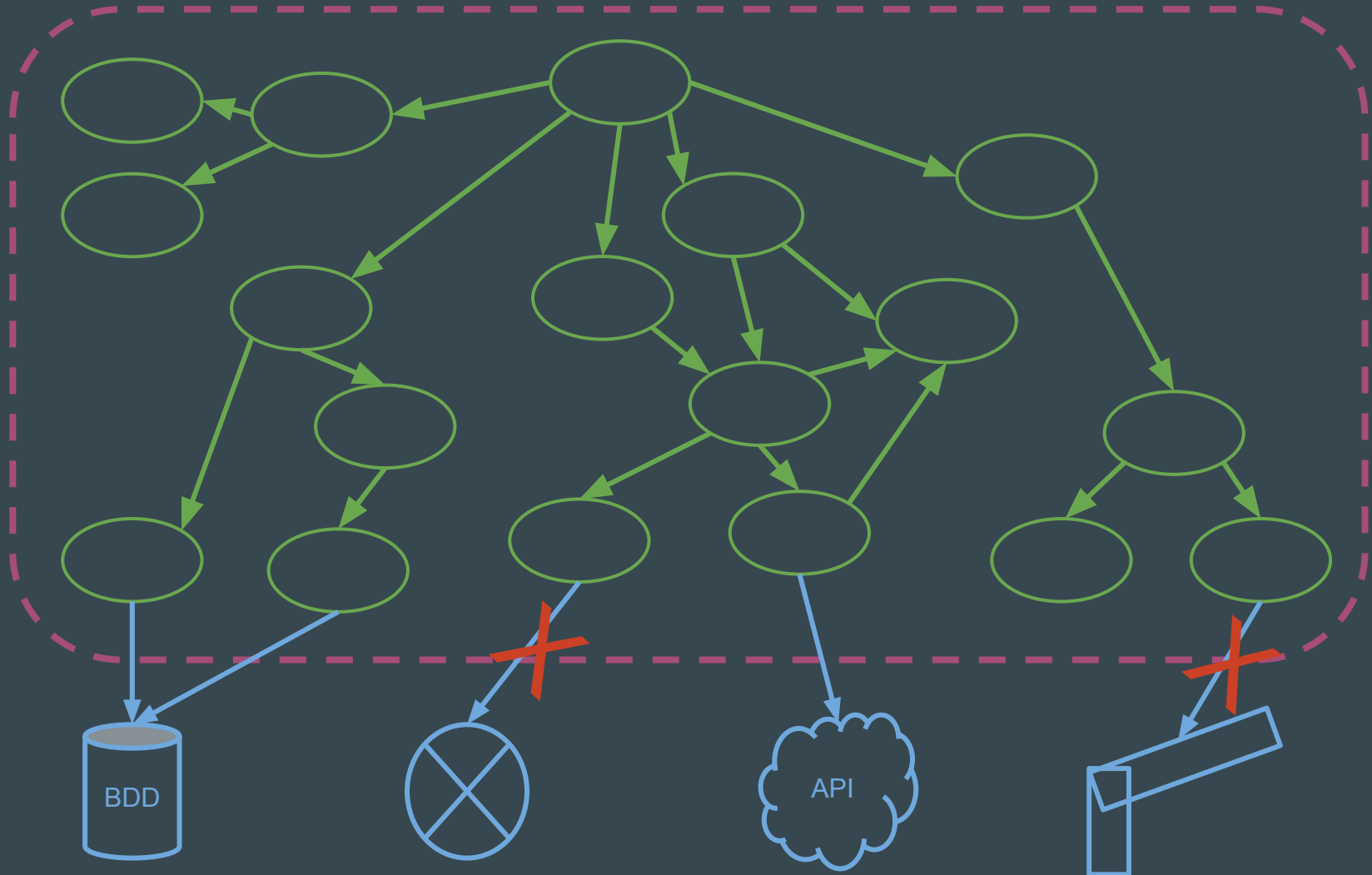
- “1 classe = 1 rôle” → présence de “collaborateurs”
  - Remplacer les collaborateurs par des “faux”
    - Pouvoir donner du comportement à ces “faux”
    - Pouvoir vérifier les interactions avec ces “faux”
- Tester 1 classe en maîtrisant son contexte par rapport aux “collaborateurs”

# Mocks : quand ?



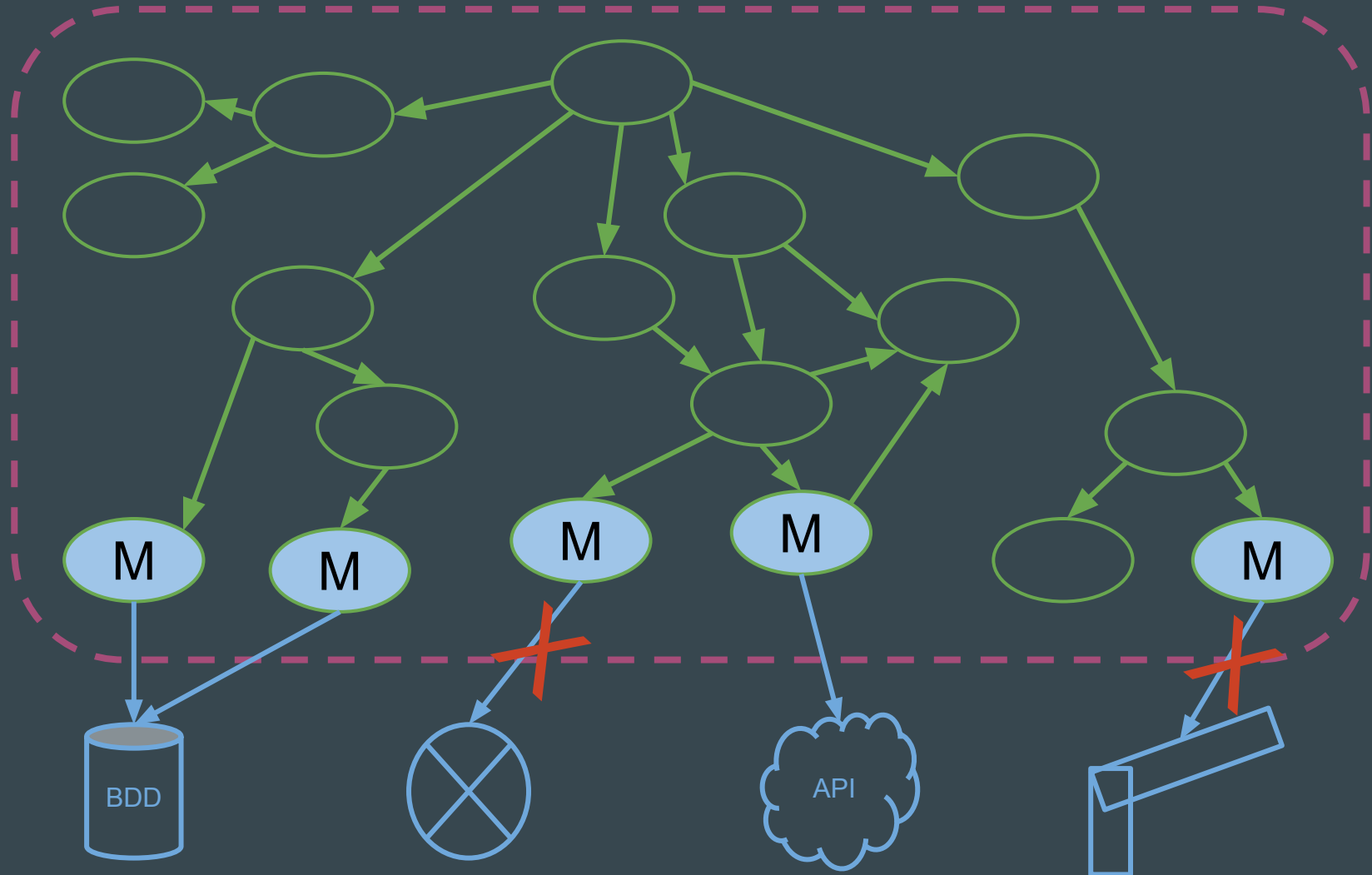
# Mocks : quand ?

Mon système



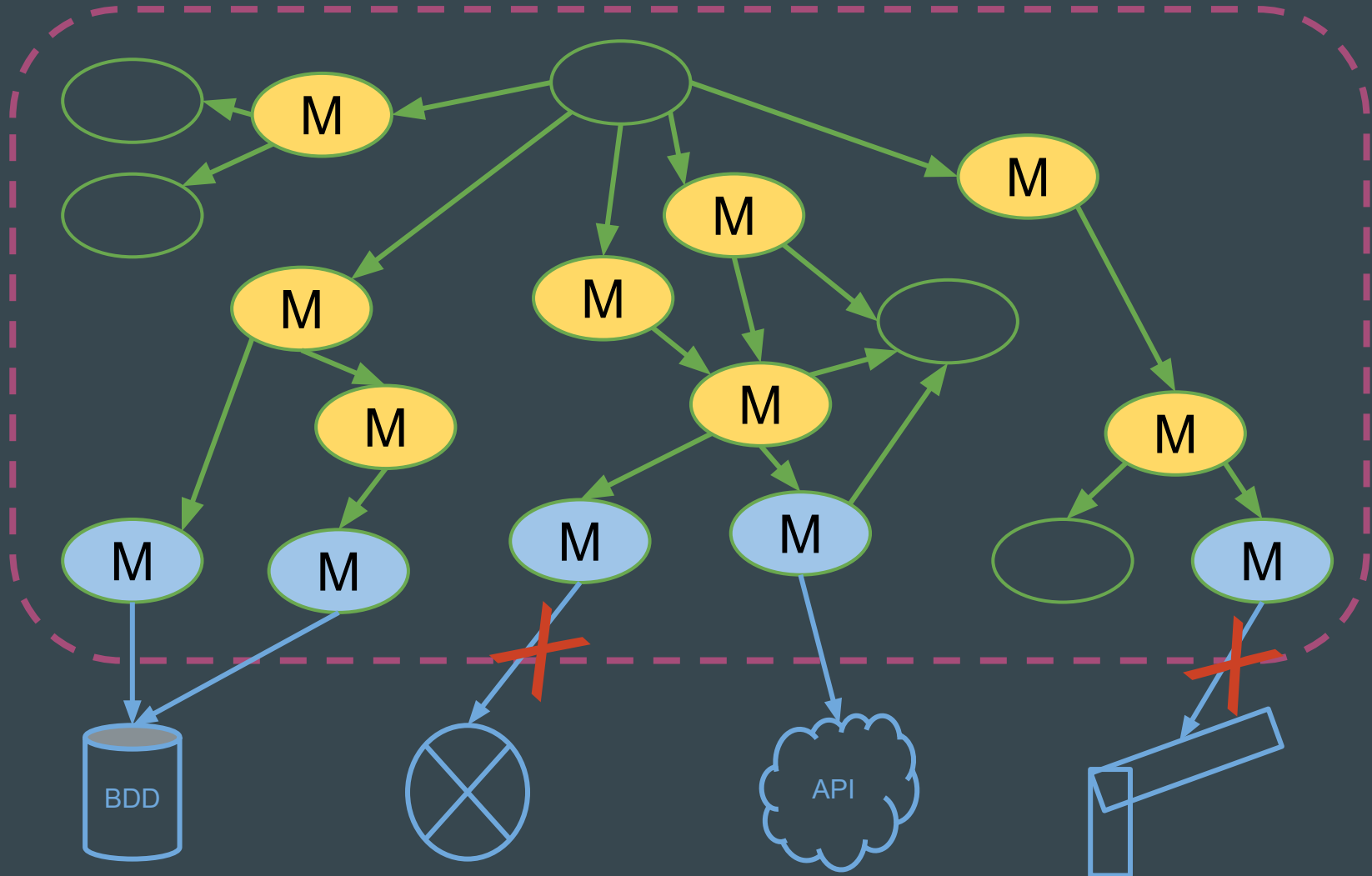
# Mocks : quand ?

Mon système



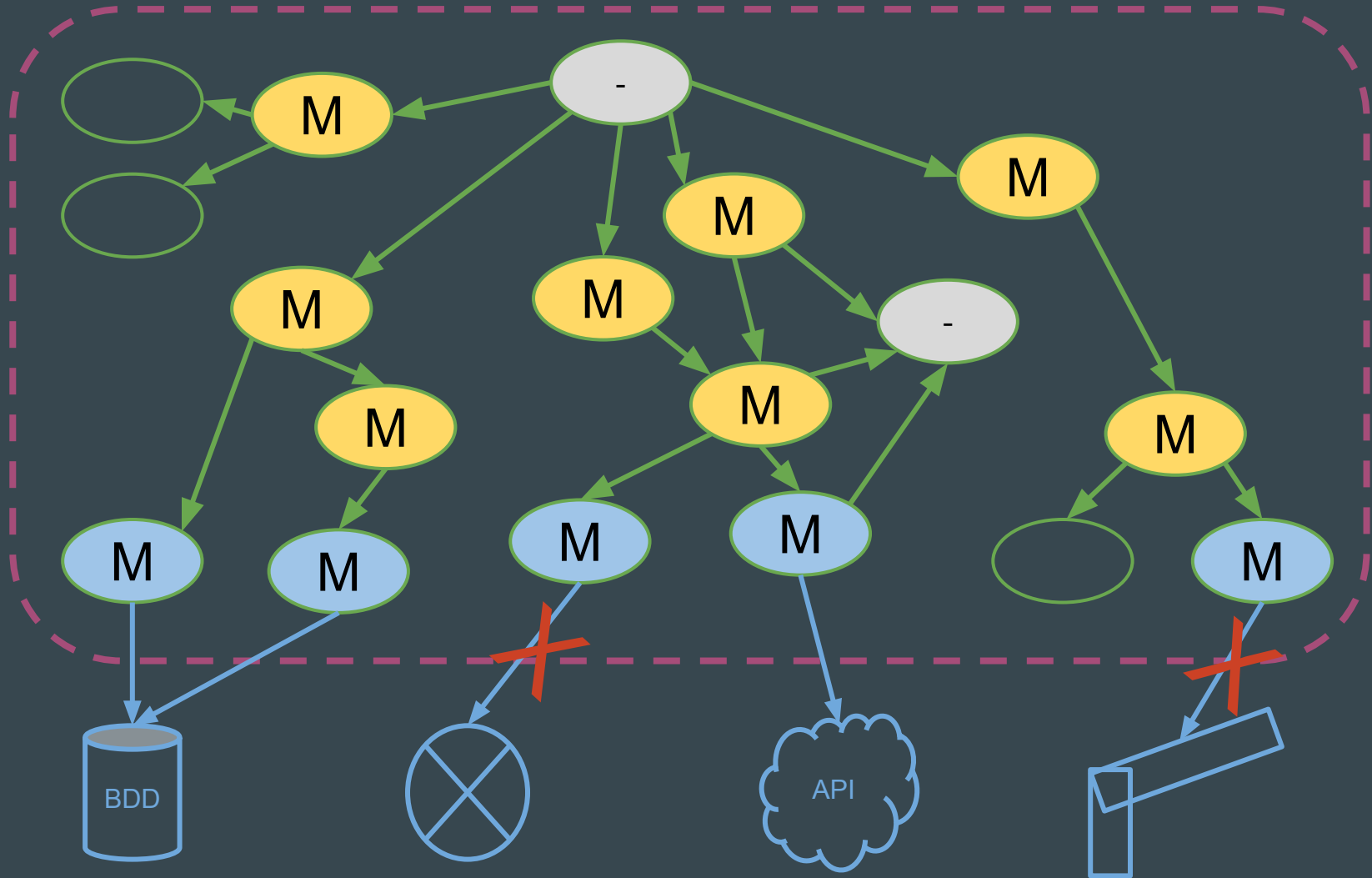
# Mocks : quand ?

Mon système



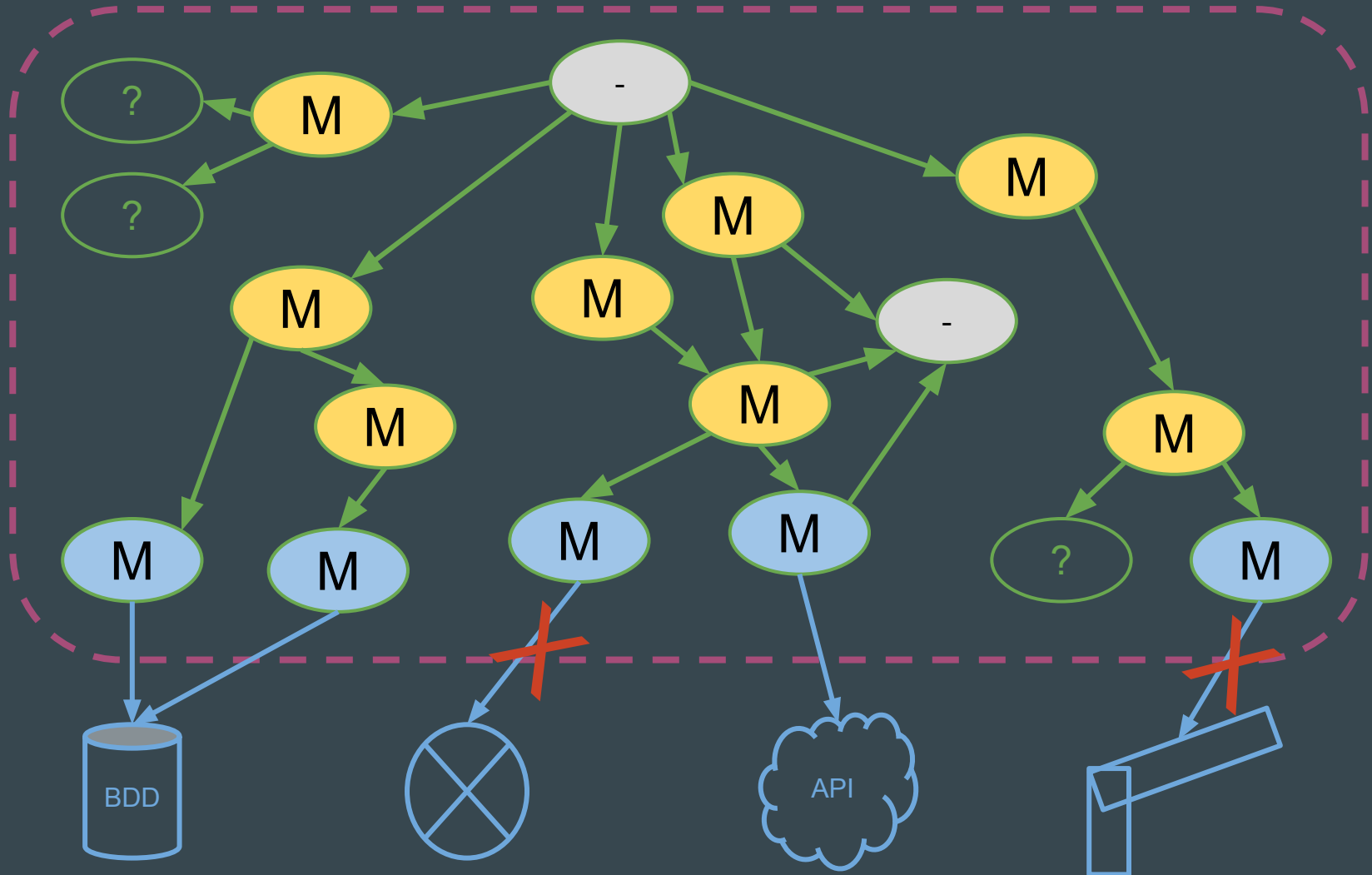
# Mocks : quand ?

Mon système



# Mocks : quand ?

Mon système





**“J’veux voir !”**



# Démo TDD

## Calcul de prix :

- HT : 3 articles à 1,21 € + 0 %  $\Rightarrow$  “3.63 €”
- TTC : 3 articles à 1,21 € + 5 %  $\Rightarrow$  “3.81 €”

# Démo TDD

## Calcul de prix :

- HT : 3 articles à 1,21 € + 0 %  $\Rightarrow$  “3.63 €”
- TTC : 3 articles à 1,21 € + 5 %  $\Rightarrow$  “3.81 €”
- TTC : 3 articles à 1,21 € +20 %  $\Rightarrow$  “4.36 €”

# Démo TDD

## Calcul de prix :

- HT : 3 articles à 1,21 € + 0 %  $\Rightarrow$  “3.63 €”
- TTC : 3 articles à 1,21 € + 5 %  $\Rightarrow$  “3.81 €”
- TTC : 3 articles à 1,21 € +20 %  $\Rightarrow$  “4.36 €”

$\rightarrow$  “TDC” = “Correction Pilotée par les Tests”

# Démo TDD

## Calcul de prix :

- HT : 3 articles à 1,21 € + 0 %  $\Rightarrow$  “3.63 €”
- TTC : 3 articles à 1,21 € + 5 %  $\Rightarrow$  “3.81 €”
- TTC : 3 articles à 1,21 € +20 %  $\Rightarrow$  “4.36 €”

$\rightarrow$  “TDC” = “Correction Pilotée par les Tests”

## Réductions :

- $> 1000$  €  $\rightarrow$  Remise 3% :
  - $5 \times 345,00$  € + taxe 10%  $\rightarrow$  “1840.58 €”

# Démo TDD

## Calcul de prix :

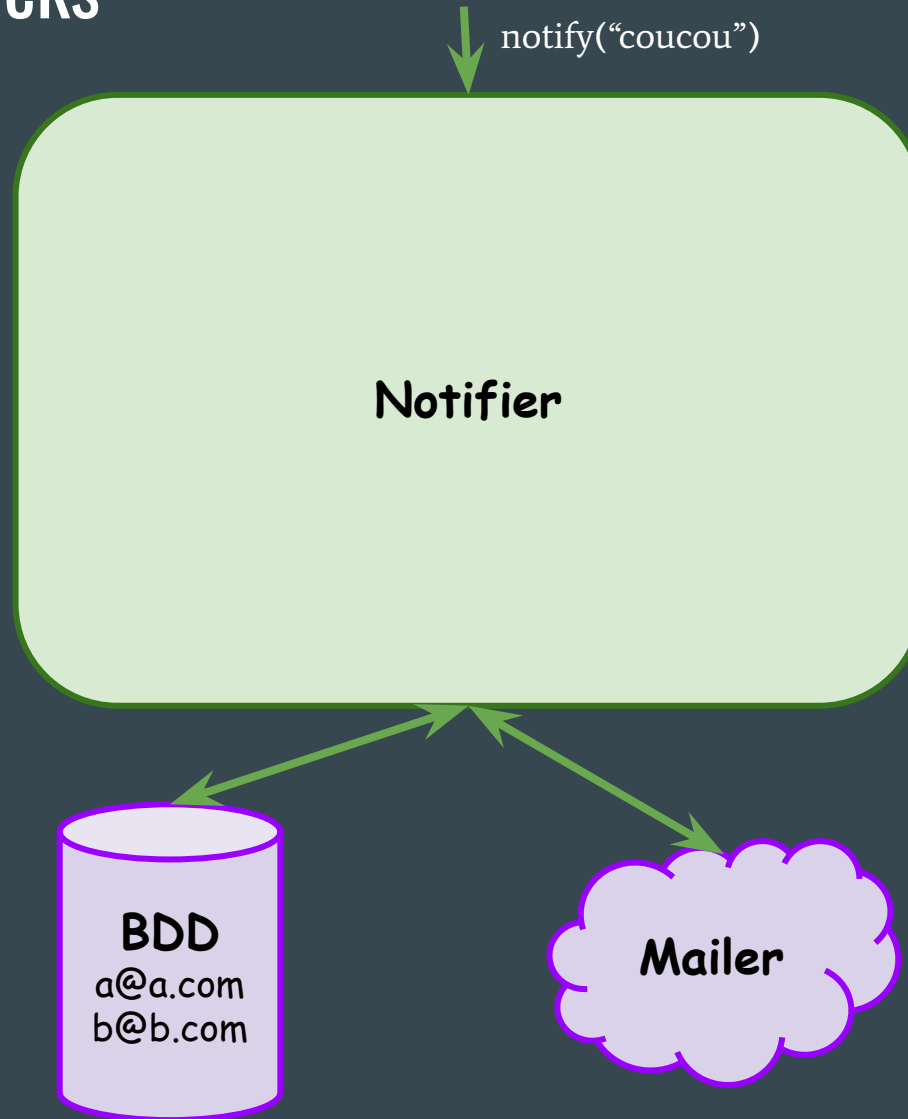
- HT : 3 articles à 1,21 € + 0 %  $\Rightarrow$  “3.63 €”
- TTC : 3 articles à 1,21 € + 5 %  $\Rightarrow$  “3.81 €”
- **TTC : 3 articles à 1,21 € + 20 %  $\Rightarrow$  “4.36 €”**

$\rightarrow$  “TDC” = “**Correction Pilotée par les Tests**”

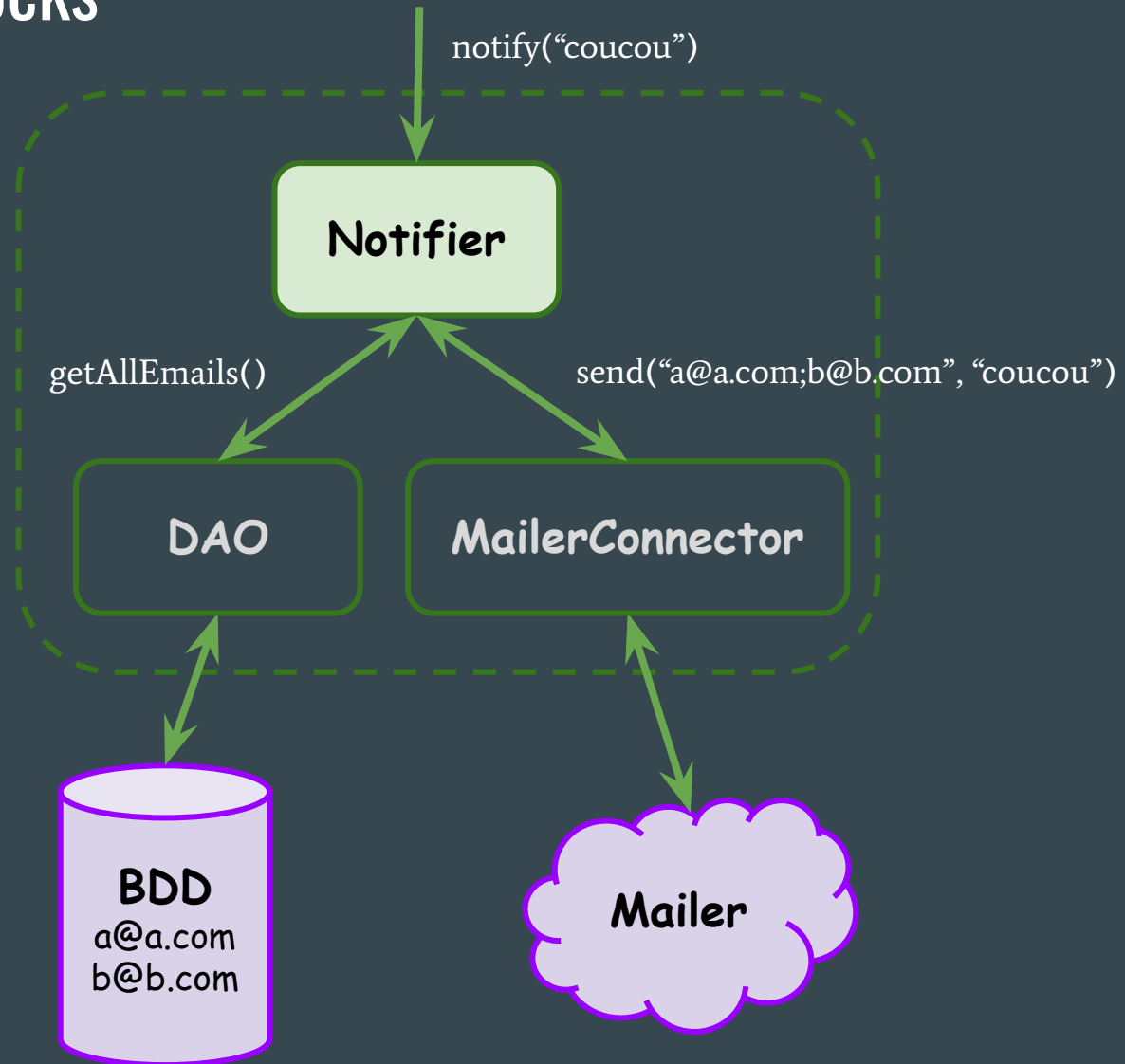
## Réductions :

- > 1000 €  $\rightarrow$  Remise 3% :
  - 5 x 345,00 € + taxe 10%  $\rightarrow$  “1840.58 €”
- > 5000 €  $\rightarrow$  Remise 5% :
  - 5 x 1299,00 € + taxe 10%  $\rightarrow$  “6787.28 €”

# Démo TDD : mocks



# Démo TDD : mocks



# Epilogue



# Mais alors ? C'est quoi le problème ?

- Apprentissage long et difficile
- Investissement (personnel et collectif)
- Impression de surcoût pour l'entourage
- Difficulté pour s'y mettre, savoir par où commencer... !

# Erreurs

- Commencer sur du code existant
- Ne pas suivre rigoureusement le cycle du TDD
- Écrire trop de tests à la fois
- Écrire des tests trop gros ou trop complexes (scénarios)
  - architecture et mocks
- Pas de démarche collective de l'équipe, ou pas d'entretien des tests, ou pas soutien du management
- Manque de vision globale de l'architecture
- Manque de compétence (langage, framework, outils, ...)
- ...

# Conseils

- Commencer sur du code nouveau
- Commencer sur des cas faciles (algorithms, calculs)
- S'y mettre progressivement
- Approche collective (pair-programming, travail en équipe)
- Suivre la couverture de code
- S'entraîner (dojo, kata)
- Se faire former et accompagner

# Merci !



## Xavier NOPRE

*Artisan-développeur & agiliste*



Indépendant : développement, formation, accompagnement, ...

Actuellement : @ Vrtice



xnopre@gmail.com

@xnopre

