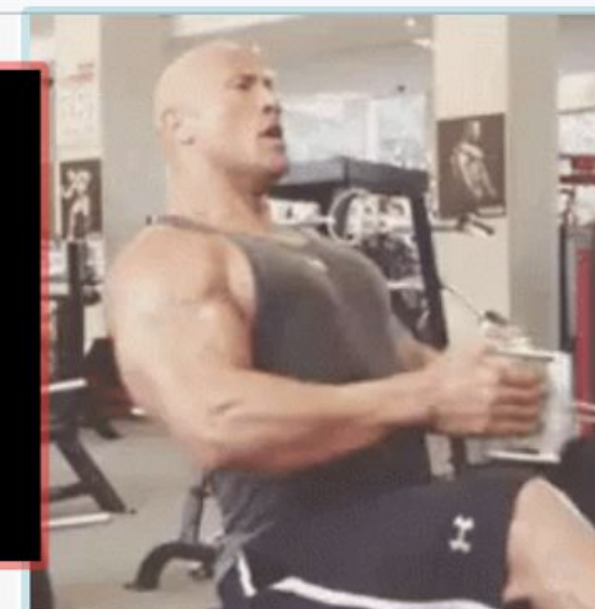




Samuel ANTUNES  
Consultant Ingénieur DevSecOps  
OCTO Technology  
Email : [contact@samuelantunes.fr](mailto:contact@samuelantunes.fr)



Qu'est-ce que vous feriez en ce moment si vous n'aviez pas cours ?





# “ L'origine du DevOps ”



# DevOps ?



**Juin 2009** : John Allspaw et Paul Hammond posent les bases de DevOps lors d'une conférence :

[10+ Deploys a Day: Dev and Ops Cooperation at Flickr.](#)



**Octobre 2009** : le mot DevOps a été inventé par Patrick Debois durant l'organisation des premiers devopsdays en Belgique.

Grosse pression du marché

On veut rapidement de nouvelles fonctions

On veut corriger plus vite les bugs voire les prévenir

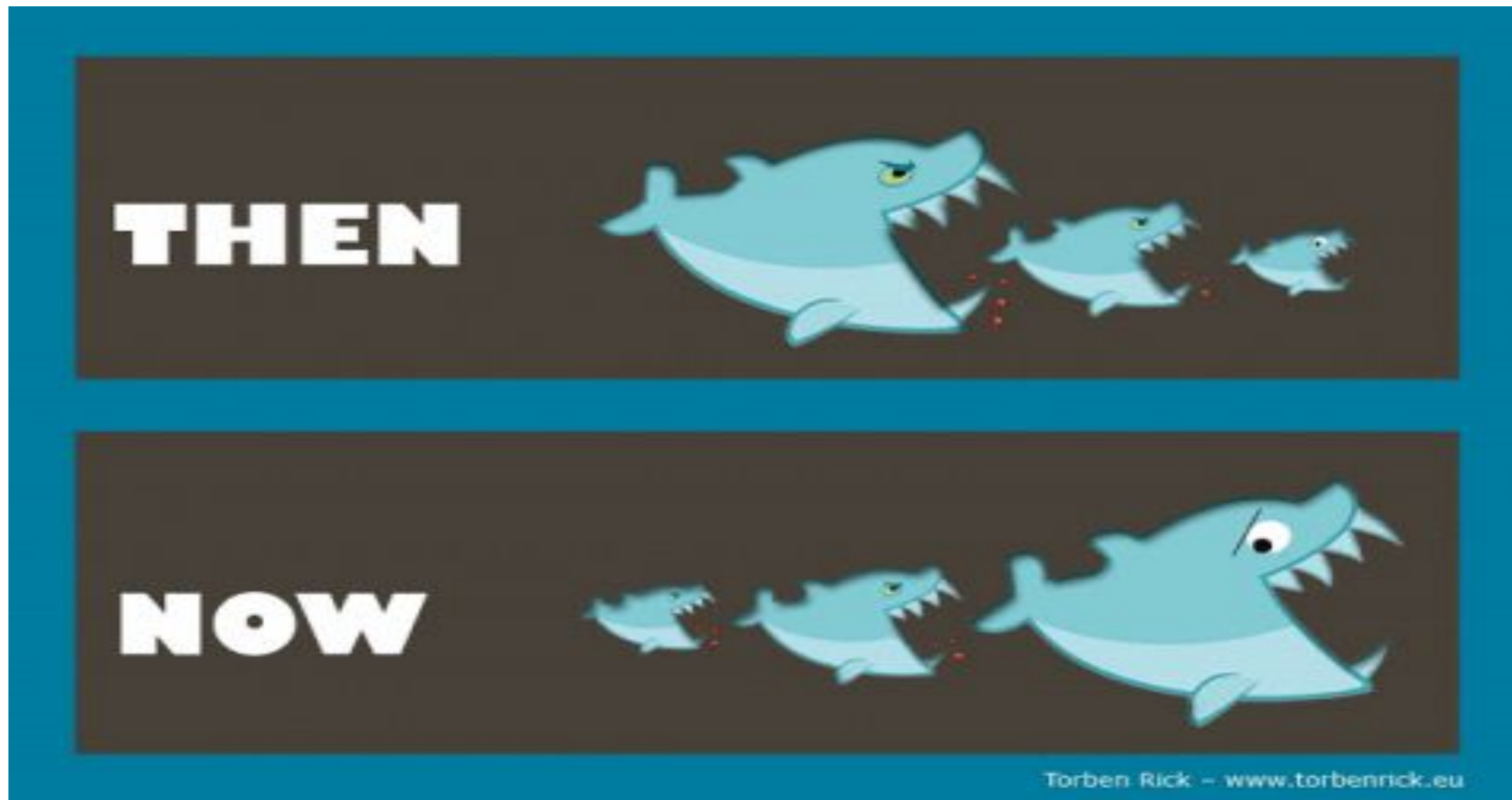
On veut un SLA « 24/24 7/7 »

On veut une meilleure UX

On veut pas payer cher

On veut pouvoir grandir vite

## Ma boîte n'est pas un Géant du Web, alors pourquoi me parler de DevOps?



- Contexte concurrentiel tendu
- Nouveaux entrants sur les marchés historiques
- Enjeux de Time to Market de plus en plus importants
- Enjeux de qualité et de disponibilité de plus en plus importants
- La transformation digitale des métiers



# “ La philosophie DevOps ”



**DevOps** est un ensemble de pratiques qui visent à réduire le Time to Market et à améliorer la Qualité en optimisant la coopération entre les **Développeurs** et la **Production**



Les méthodes Agile accordent une valeur plus importante (pas UNIQUE) à :



Personnes et échanges



Méthodes et outils



Logiciel qui fonctionne



Documentation  
exhaustive



Collaboration avec le  
client



Négociation  
contractuelle

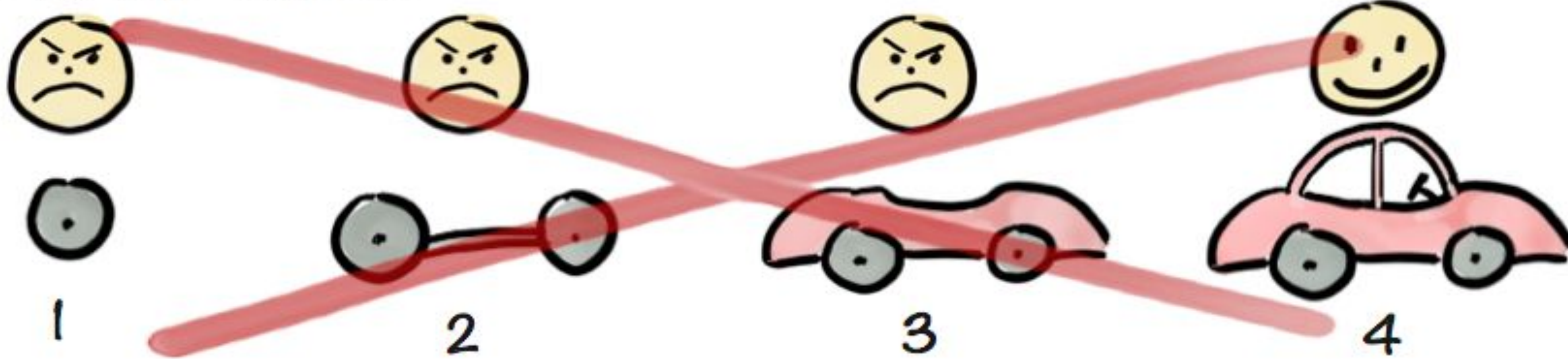


Adaptation au  
changement

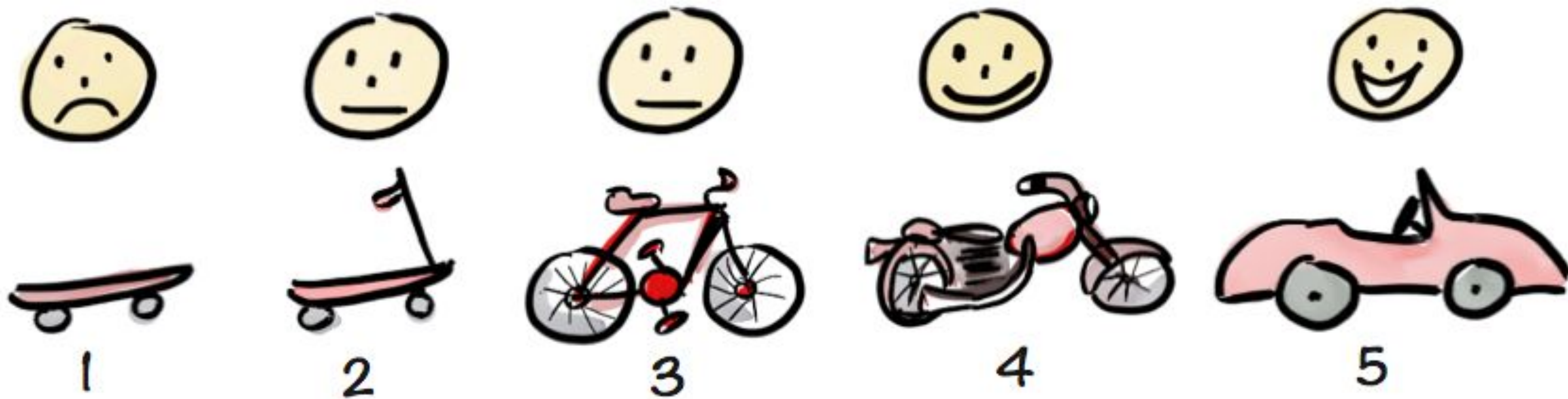


Suivi d'un plan établi

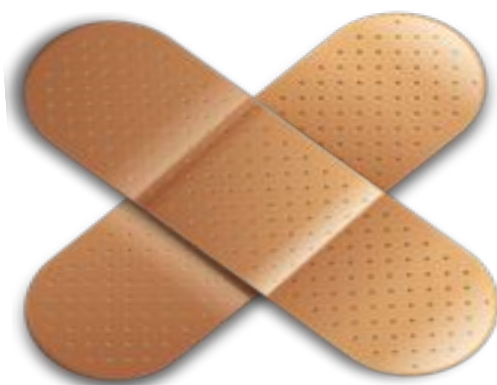
Not like this....



Like this!







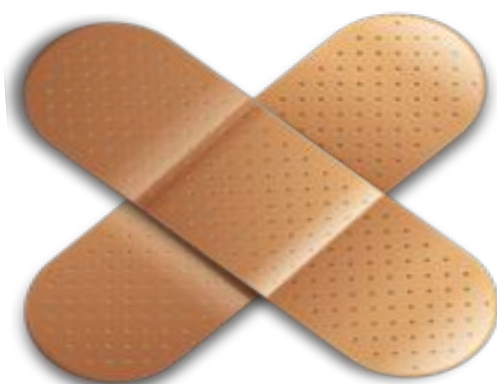
## DOULEUR #1

Trop long pour provisionner des environnements



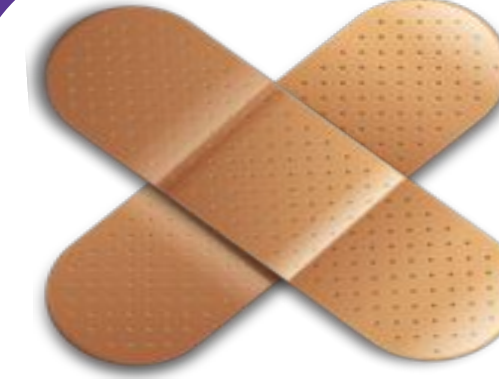
## DOULEUR #2

C'est chaud entre les DEV et les OPS



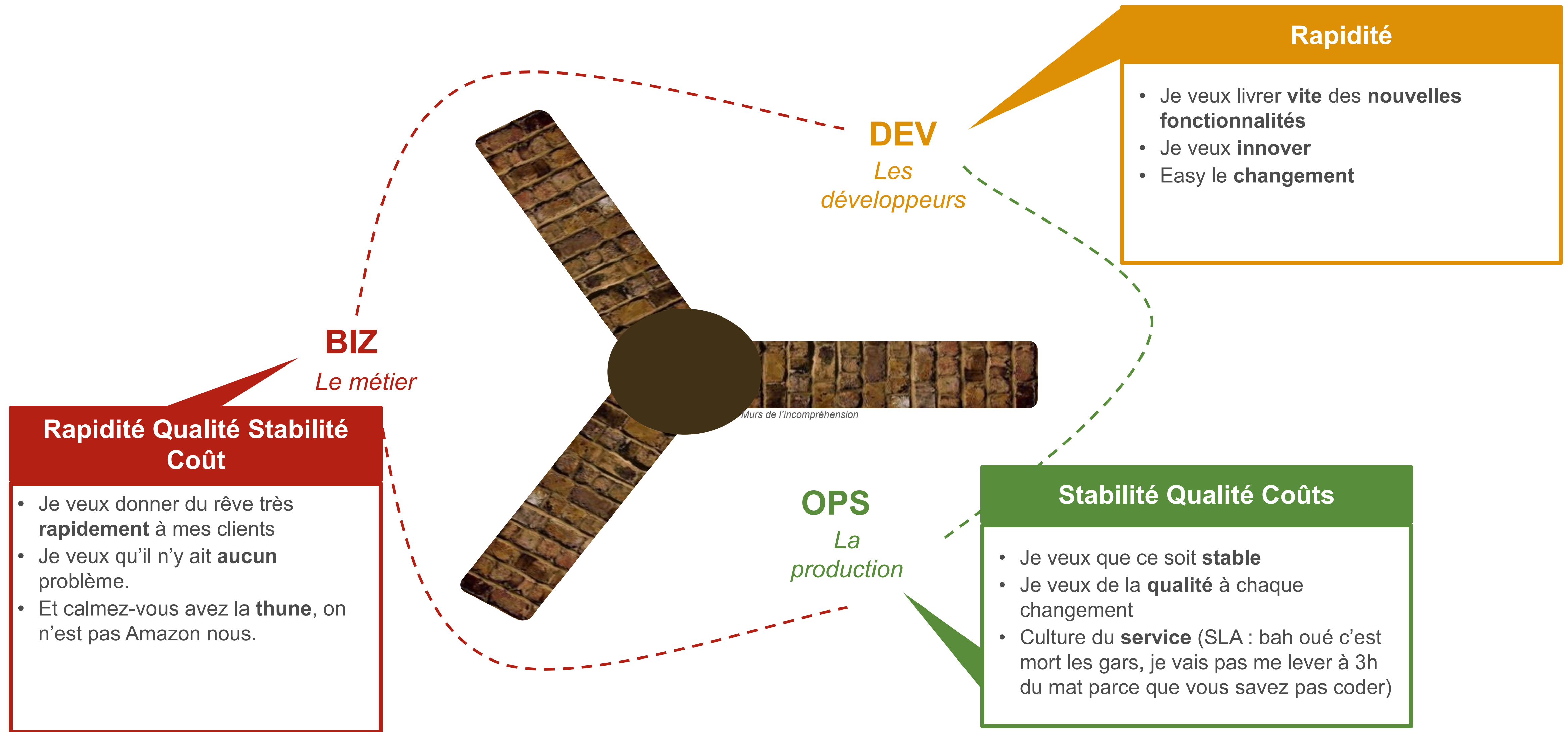
## DOULEUR #3

RIP la qualité qui provoquent des anomalies en prod



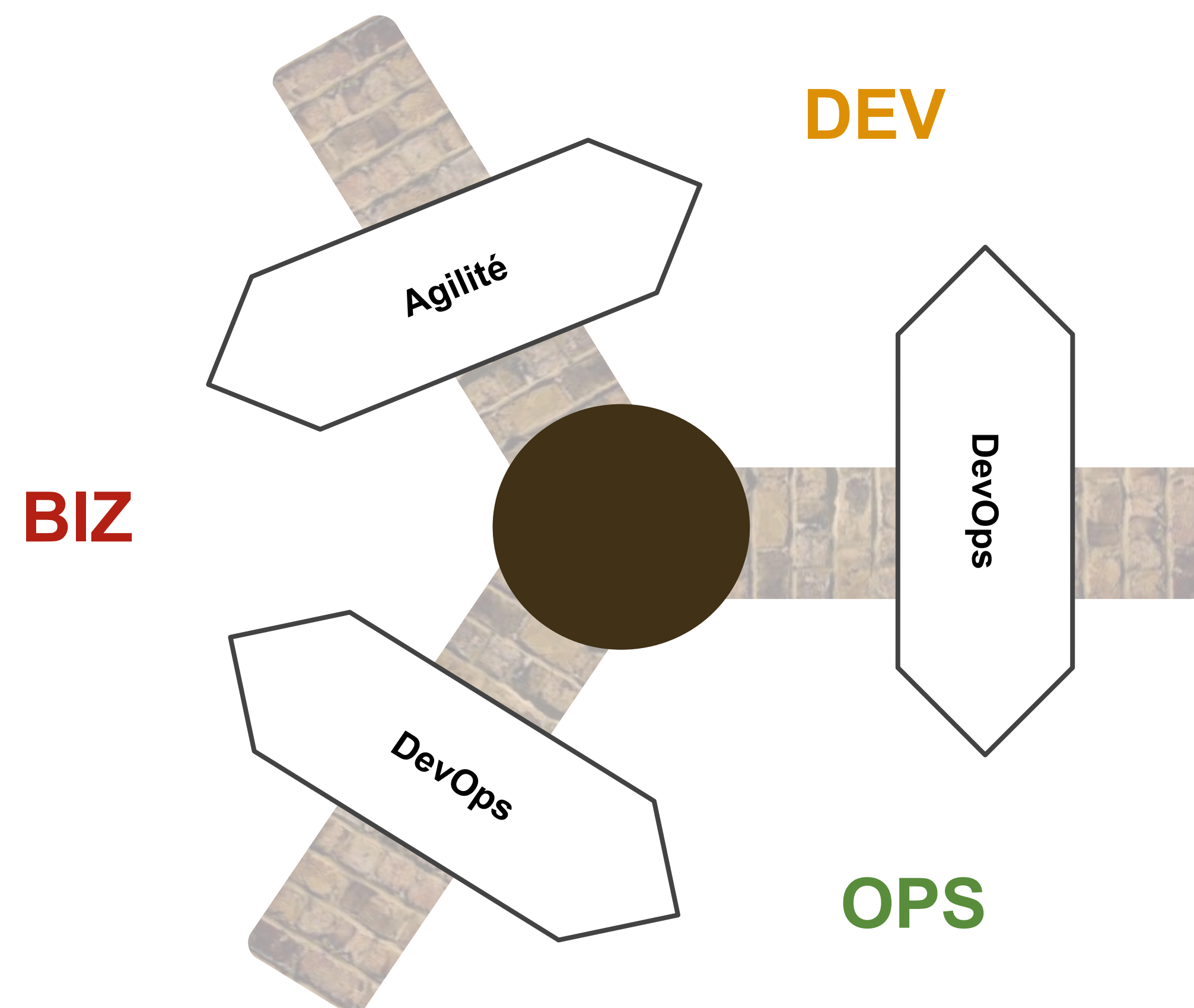
## DOULEUR #4

Le déploiement de l'appli est beaucoup trop long

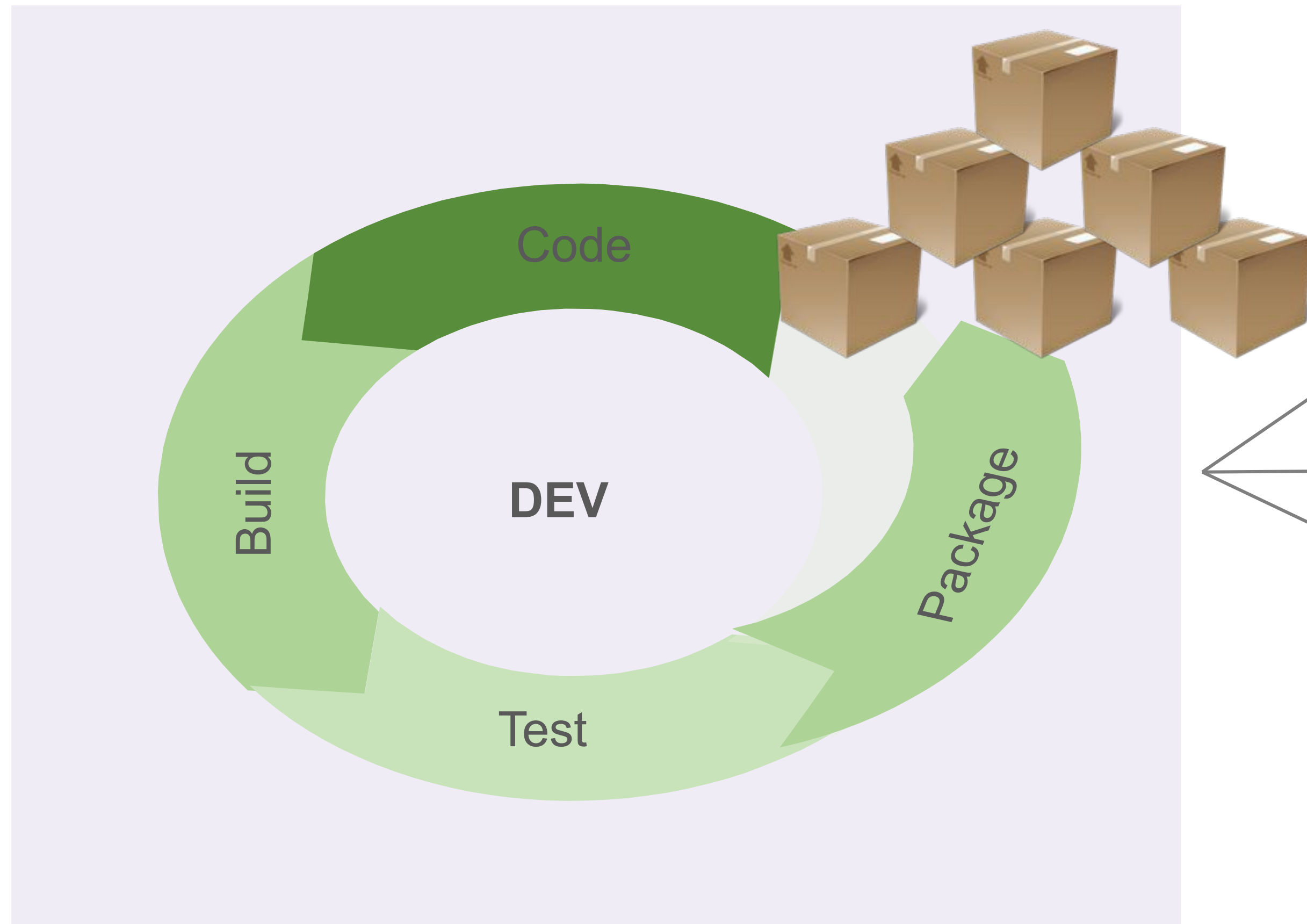




## LES PRATIQUES POUR ROMPRE LES MURS...

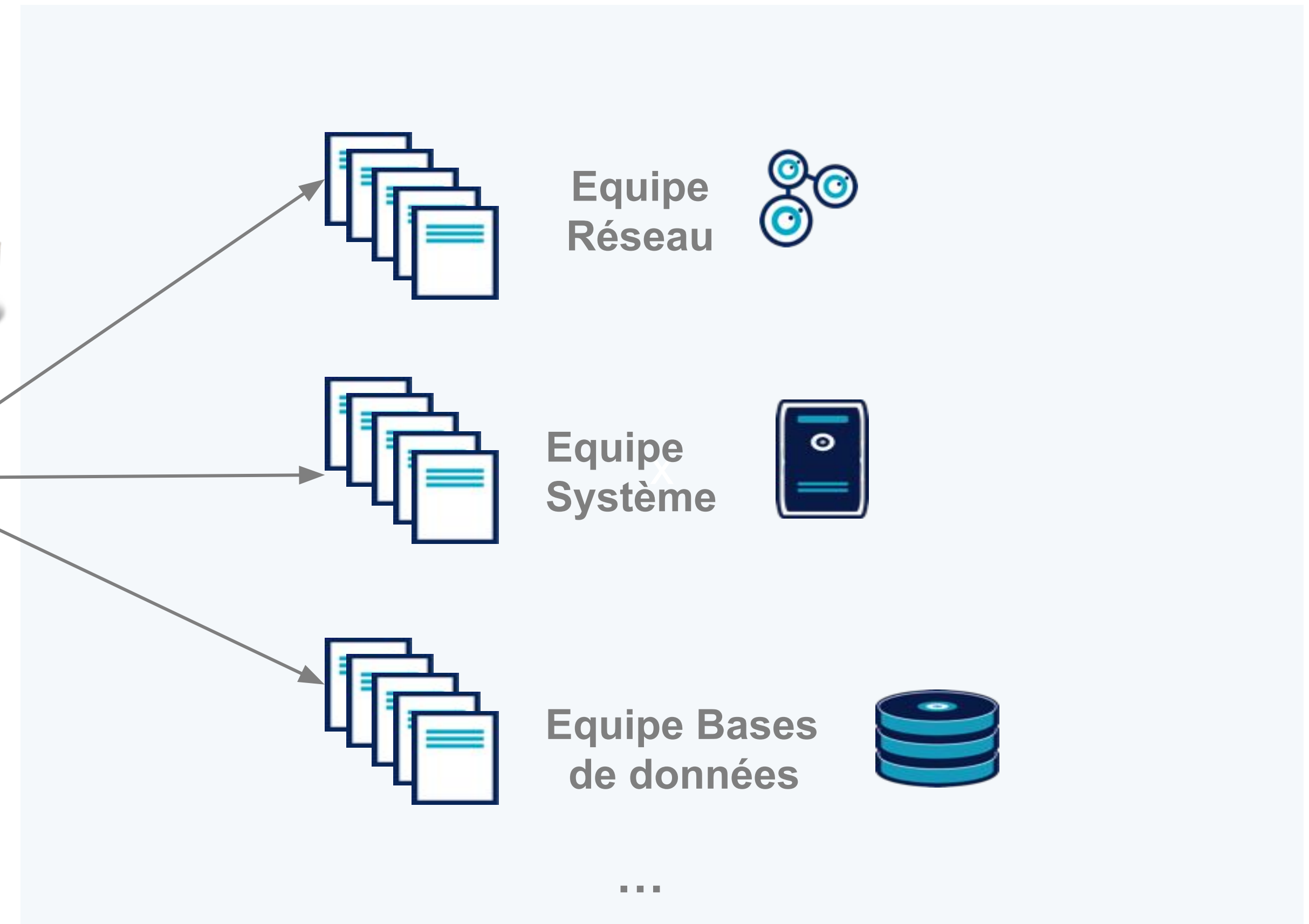


## Développement Agile : équipe autonome



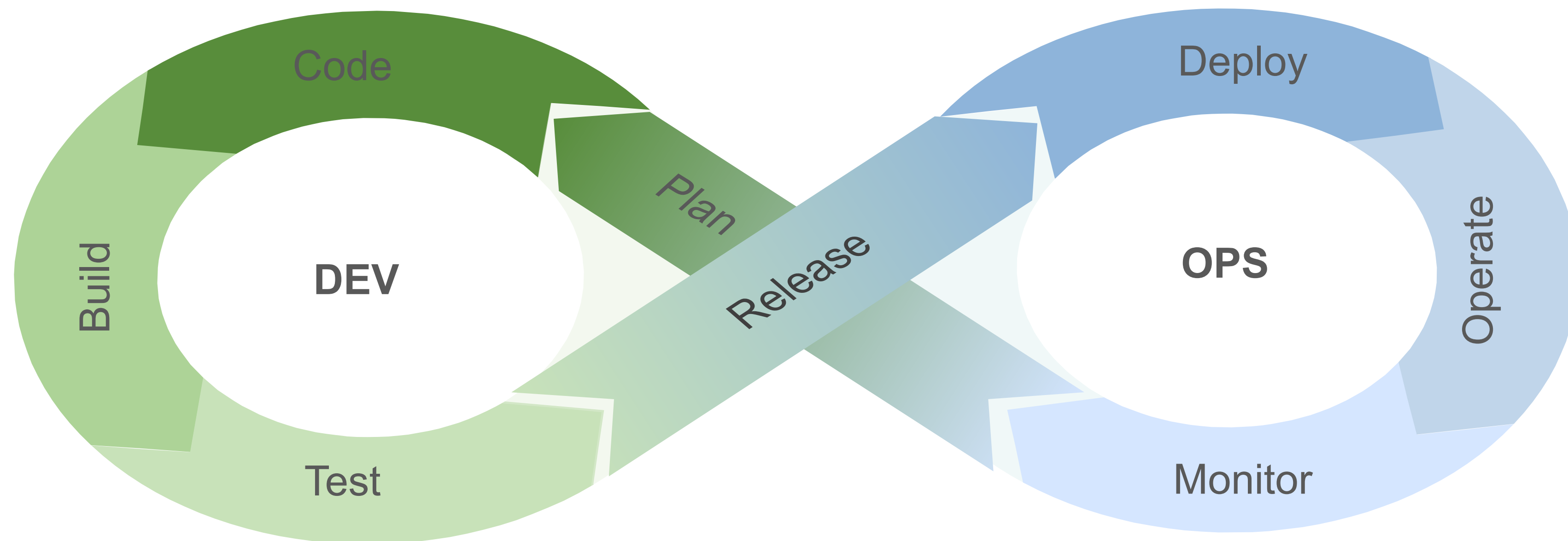
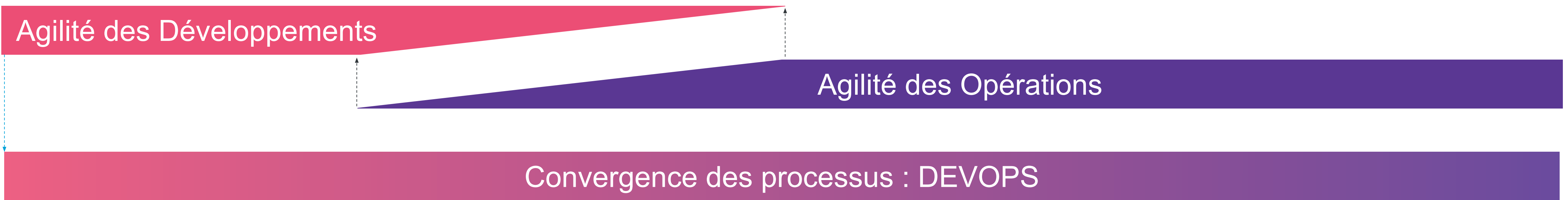
- Durée : 2 - 3 semaines

## Ecosystème en silo



- Durée : 2 - 3 mois





• Durée : 2 - 3 semaines

# “ Les 4 piliers du DevOps ”

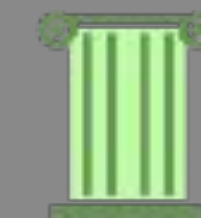






## Architecture et patterns

Architectures et patterns répondant aux préoccupations des Devs et des Ops



## Infrastructure as code "IaC"

Outils d'automatisation pour construire et maintenir l'infrastructure par du code



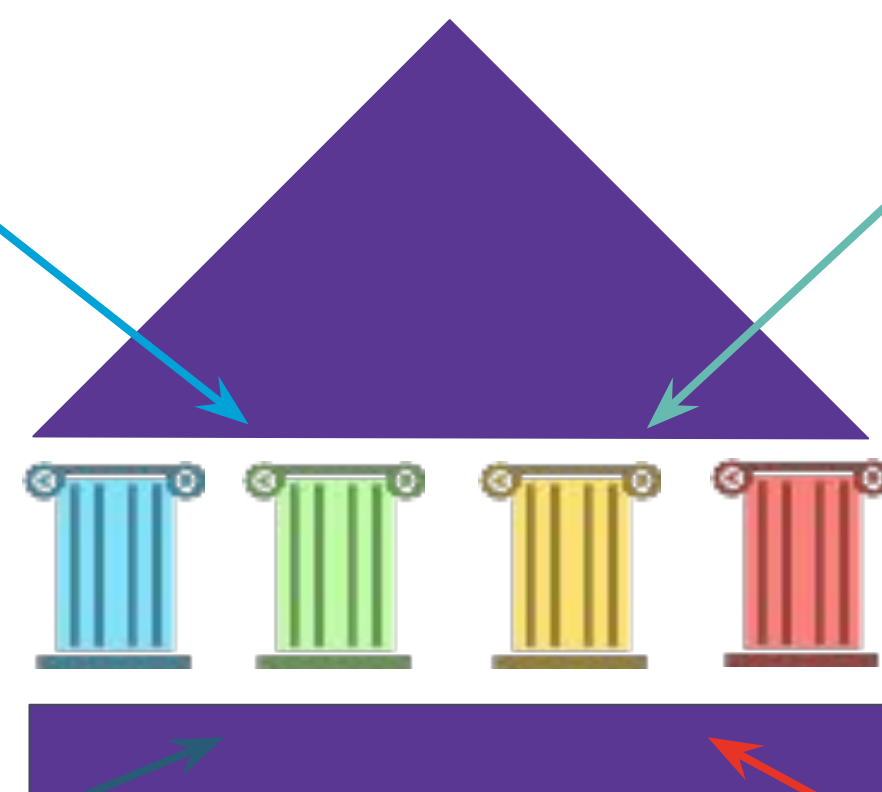
## Construction et déploiement continu

Outils et processus de construction et déploiement continu



## Culture, méthode et organisation

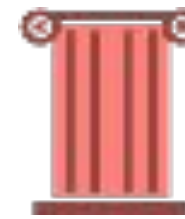
Culture, des modèles d'organisation et des méthodes favorisant le dialogue des Dev et des Ops





- ▷ **Des modèles d'organisation favorisant l'autonomie et la responsabilisation (aussi appelé le Shift-Left)**
- > Décloisonnement des organisations (fin des silos techniques)
- > « *You build it, you run it* »
- > Autonomisation des équipes
- > Les Pizza Teams (celles de chez PizzaHut avec le cheezy crust bien sûr, pas votre domino's là)

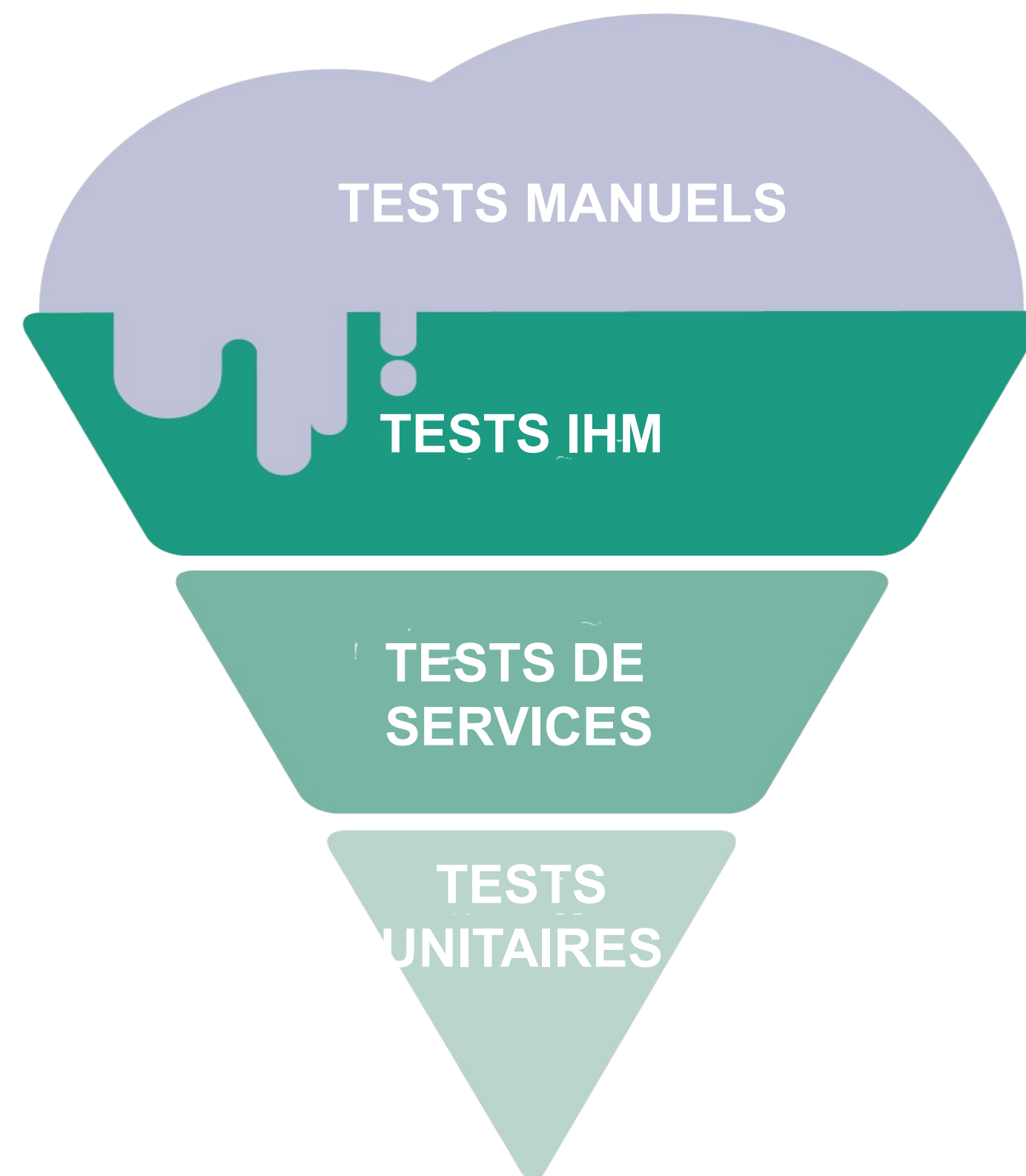




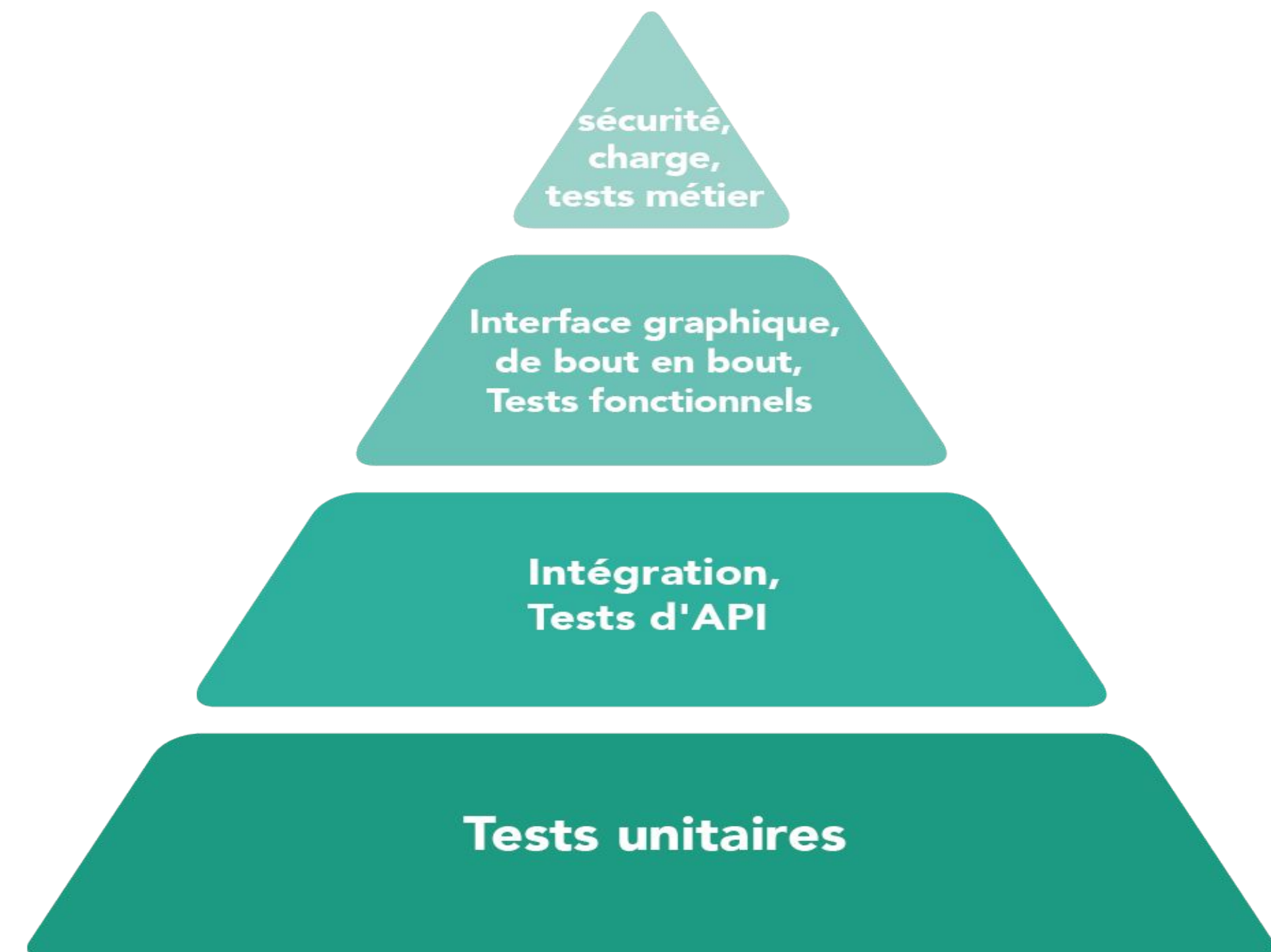
RITUELS	VIS MA VIE (SHADOWING. Qui a dit #VDM ?)	WAR ROOM	PEER REVIEW	BBL
	RÉTROSPECTIVE	POST-MORTEM	PAIR-PROGRAMMING	HACKATHON
ATTITUDES	TRANSPARENCE	OPEN DATA	DROIT À L'ERREUR	SOLIDARITÉ
METHODO	KANBAN	KAIZEN	LEAN STARTUP	5 WHY



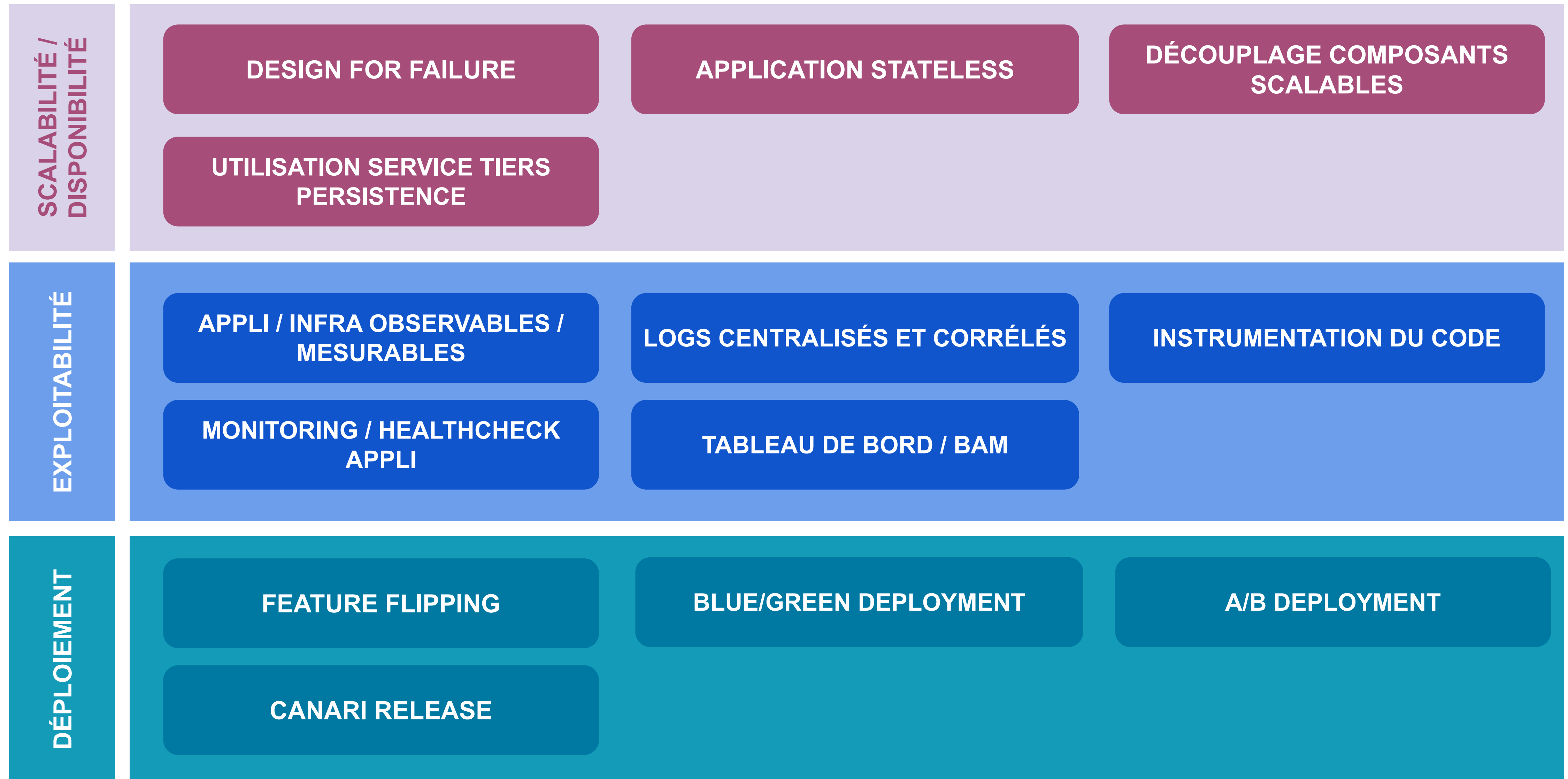
Approche traditionnelle :  
**Trouver les bugs**



Approche agile:  
**Prévenir les bugs**

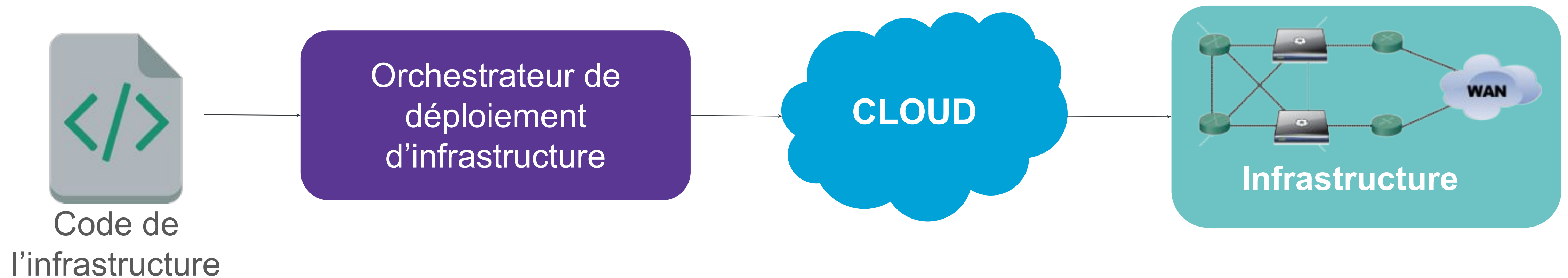








Décrire sous forme de code exécutable et testable la configuration de l'architecture technique **d'infrastructure**







- ▷ **Automatiser la construction des environnements et la répétabilité des processus pour :**
  - > Garantir une infrastructure homogène
  - > Assurer le respect des standards en place
  - > Ouvrir l'exécution de certaines tâches aux développeurs
  - > Configurer un environnement plus rapidement
  - > Avoir un processus déploiement fiable, reproductible et portable
  
- ▷ **Appliquer les pratiques de qualité du monde logiciel à l'infrastructure :**
  - > Tests unitaires
  - > Qualité de code
  - > Code review
  - > Gestion de version
  
- ▷ **Gérer un parc important de serveurs avec peu d'administrateurs système**



Exemples de produits du marché :



Deployit



Déploiement / configuration  
des **Applications**

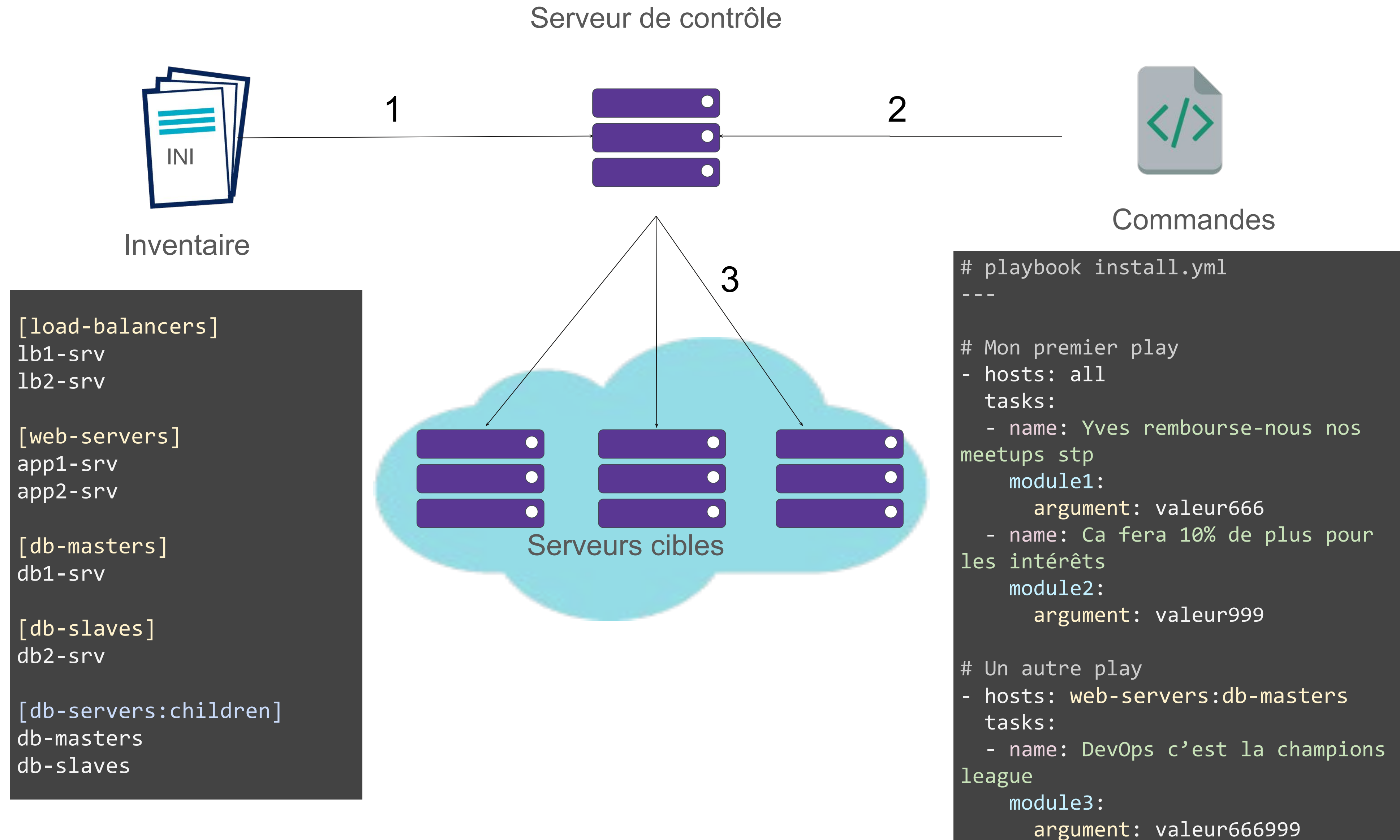


Installation / configuration  
des **Middleware**s



Provisioning de  
l'infrastructure  
(IaaS/PaaS/CaaS)







```
# Configure server front
---
- host: Web-Server
  tasks :
    - name: Ensure apache server is installed
      package:
        name: "apache"
        state: "present"

    - name: Ensure if apache is started and enabled
      service:
        name: "apache"
        state: "started"
        enabled: true
```





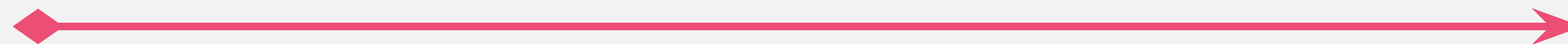
- ▷ L'automatisation du processus logiciel doit emmener le produit jusqu'à son environnement de production



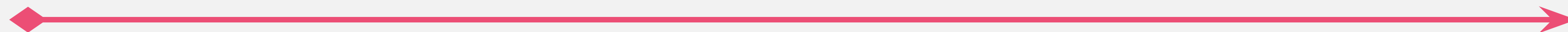
Intégration continue

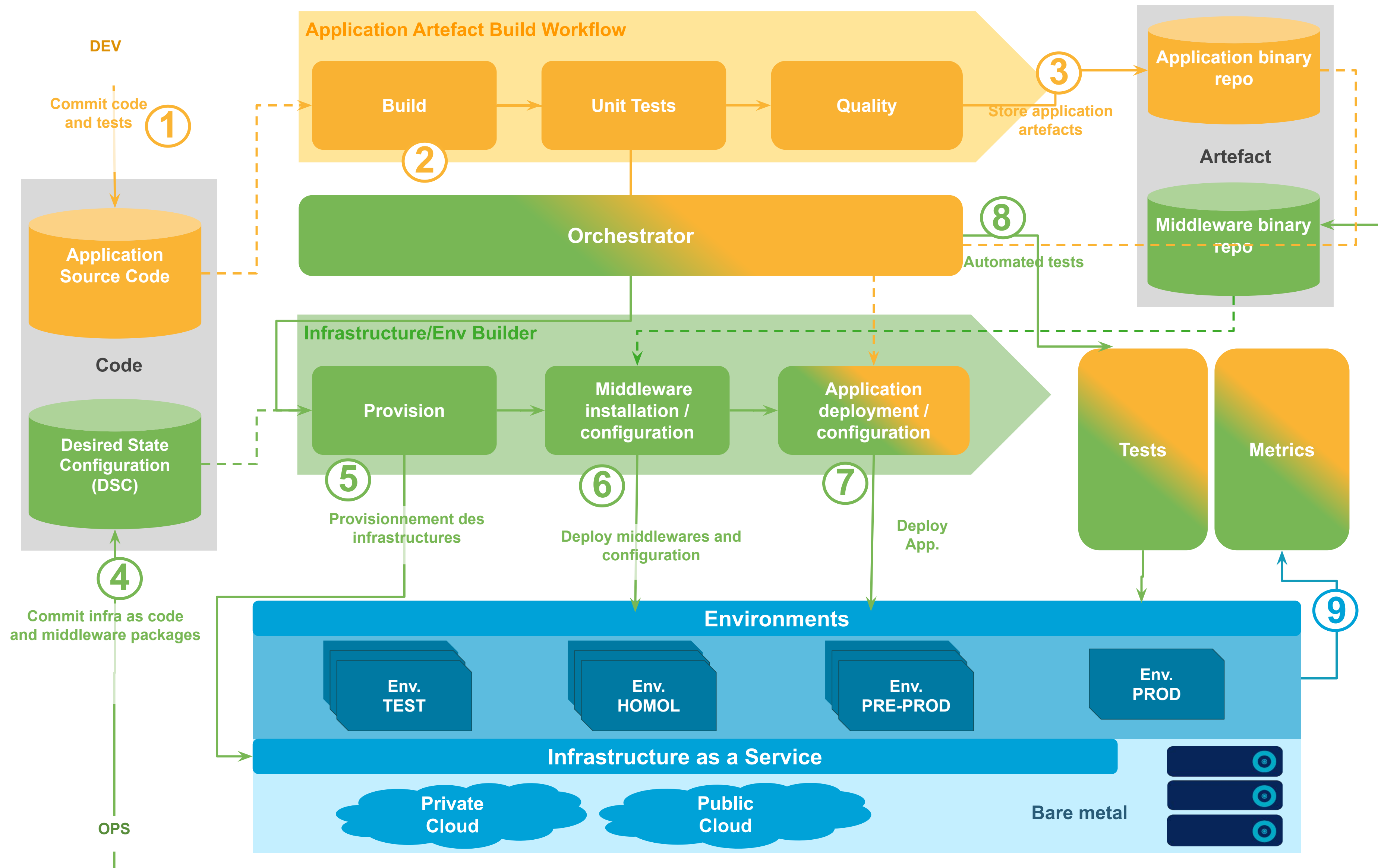


Livraison continue



Déploiement continu

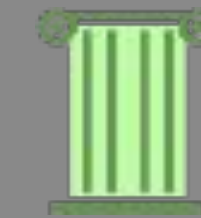






## Architecture et patterns

Architectures et patterns répondant aux préoccupations des Devs et des Ops



## Infrastructure as code "IaC"

Outils d'automatisation pour construire et maintenir l'infrastructure par du code



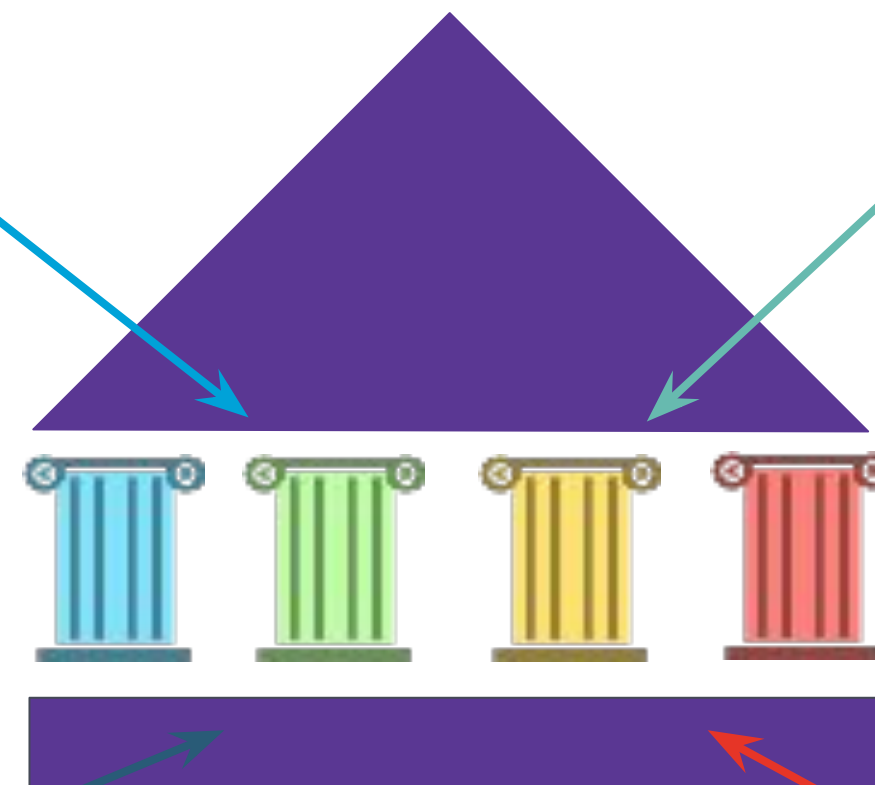
## Construction et déploiement continu

Outils et processus de construction et déploiement continu



## Culture, méthode et organisation

Culture, des modèles d'organisation et des méthodes favorisant le dialogue des Dev et des Ops





# “Projet Semestre DevOps”



Minimum demandé (note "OK") :

- Définition d'une stratégie de déploiement continu (workflow git à expliquer)
- Déploiement d'une API sur AWS EC2
- Automatisation CI/CD avec Gitlab-ci (jusqu'en prod)
- Déploiement de l'infra avec Terraform
- Hébergement sur une VM AWS configurée avec Ansible

Quelques idées en plus (bonne note  $\leq 16$ ) :

- Application 3 tiers (front - api - bdd)
- Ajout de tests / linters dans la pipeline
- Stockage des artefacts ou images docker
- Stack de monitoring (Cloudwatch étant le plus simple)
  - Auto-scaling