AUTOMATISATION DEVOPS

Intégration Continue

Ensemble de pratiques et de solutions logicielles concourant à éliminer les erreurs - « régressions » consécutives à l’évolution d’un projet, pour amener un produit à l’état de livrable, i.e démontrable à un client.

Exemples :

- Gestion automatisée du build / compilation
- Tests automatisés : unitaires, « End To End / E2E », couverture de tests
- Gestion de la qualité : conventions, lisibilités, performances
- Gestion de la sécurité : respects des normes, pentesting
- Gestion de l’information projet : Contrôle de version (VCS), monitoring, datavisualisation, chatOps

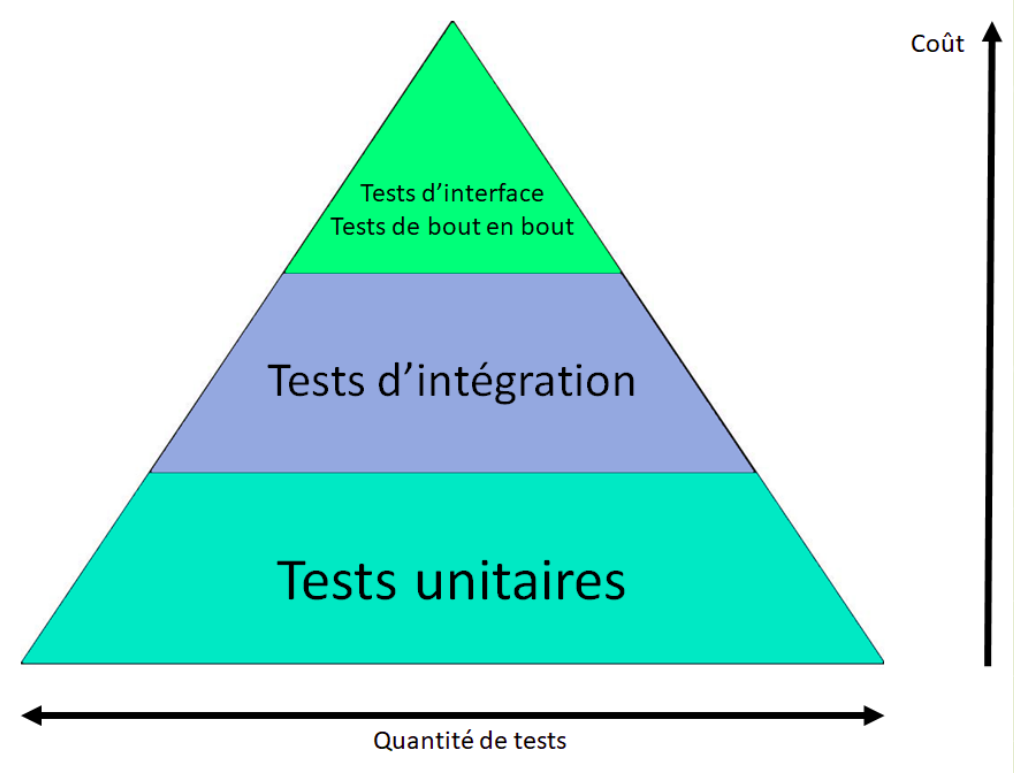
DAWAN - Reproduction interdite

Intégration Continue

- Tester le code

On prend en compte la granularité du code :

- Les tests unitaires valident les éléments logiques élémentaires du code : fonctions ou méthodes d'objet si POO
- Les test d'intégration valident et assurent la cohérence des structures intermédiaires du code : classes, composants, modules
- Les test d'interface ou tests de bout en bout (End to End ou e2e) valident le comportement global de l'application depuis ses points d'entrées (interfaces)

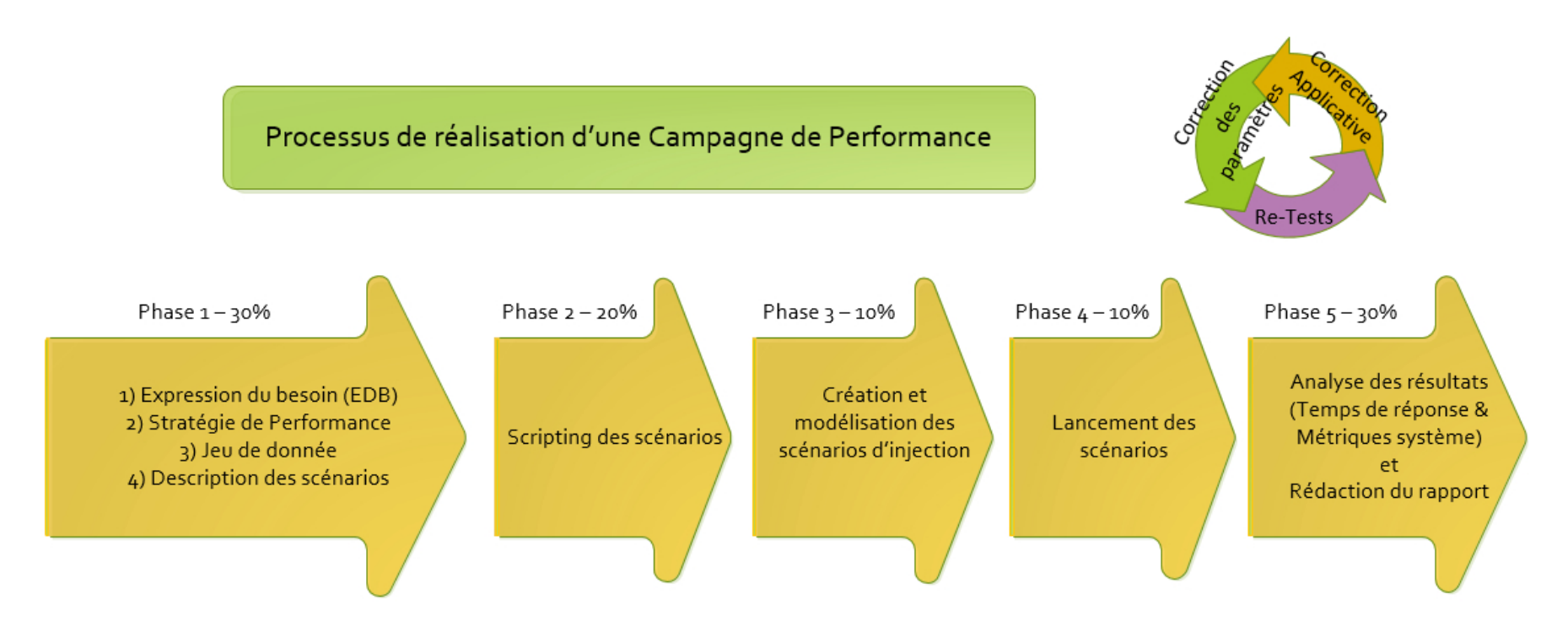


DAWAN - Reproduction interdite

Intégration Continue

- Tester les performances

- Un test de performance vise à mesurer la réponse d'une application en soumettant celle ci à différents scénaris d'usage



DAWAN - Reproduction interdite

Intégration Continue

- Types de Tests de performances

- Un test de charge vérifie la capacité d’une application à répondre à nombre données de connexions utilisateurs
- Un test de montée en charge doit déterminer le nombre maximal d’utilisateurs simultanés que l’application peut supporter
- Un test d’endurance mesure l’évolution des temps de réponse (affichage, réseau, bdd) d’une application soumise à une longue période d’intense activité.
- Un test de stress mesure la réponse d’une application exécutant l’intégralité de ses scénaris fonctionnels en forte activité.
- Un test de résilience mesure la réponse d’une application au déclenchement de défaillances système ou réseau

DAWAN - Reproduction interdite

Intégration Continue

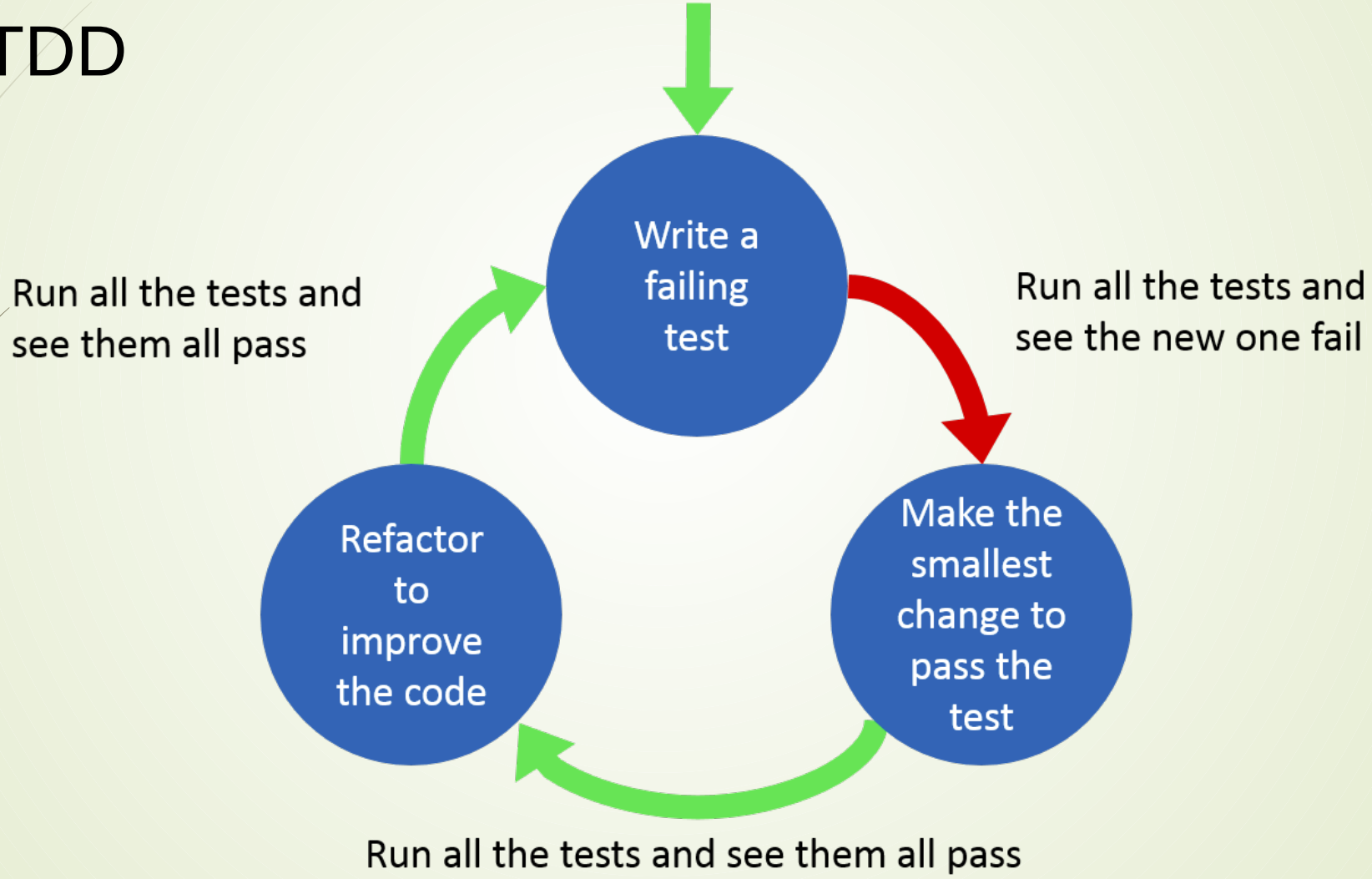
- TDD

- « Test Driven Development » ou développement piloté par les tests :
- Consiste à développer en commençant par coder les conditions d'échec du code : le test
- Ensuite on produit un code qui doit passer le test
- Enfin on remanie ce code à des fins d'optimisation et de non regression « Refactoring »
- On poursuit ces itérations sur l'ensemble des fonctionnalités à coder
- La production de tests systématique, alliée à la standardisation des ces tests par des outils dédiés sont une condition nécessaire de l'automatisation d'un processus d'intégration continue
- La TDD est emblématique d'une démarche « shift Left », car les tests sont avancés au niveau du développement

DAWAN - Reproduction interdite

Intégration Continue

- TDD



DAWAN - Reproduction interdite

Intégration Continue

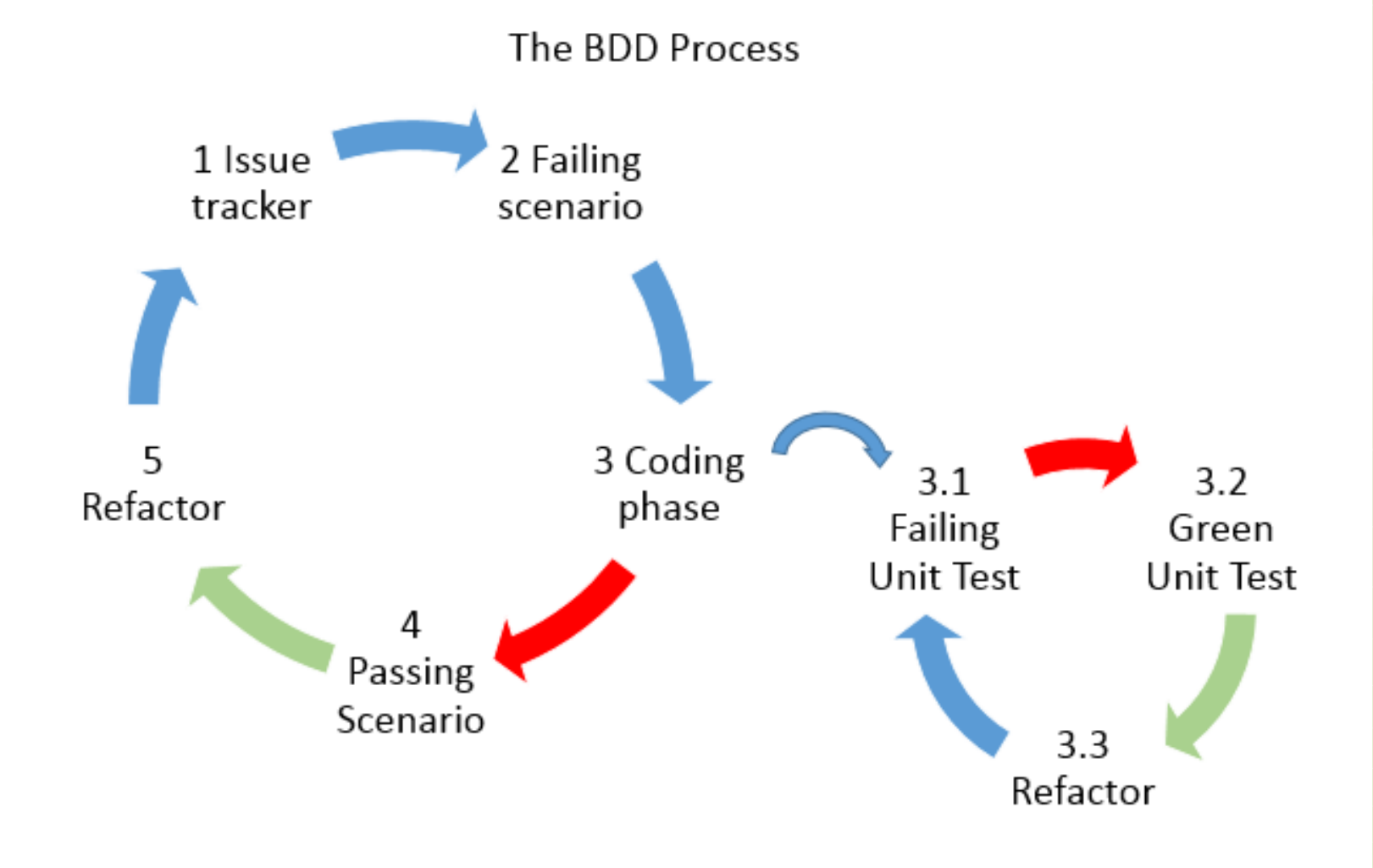
- BDD

- « Behavior Driven Development » ou développement piloté par les comportements :
- On parle ici des comportements attendus du produit, soit les spécifications fonctionnelles (ou « User Stories ») fruits des échanges entre le client et l'équipe technique
- D'abord redigées en langage naturel, ces spécifications sont réécrites dans un cadre formel tel que given / when / then
- On peut agrémenter cette description d'exemples concrets, si possible automatisables
- On applique La TDD sur cet ensemble
- La BDD permet de rassurer le client sur le respect de son expression de besoin là où la TDD valide l'aspect technique du code, ou les spécifications détaillées

DAWAN - Reproduction interdite

Intégration Continue

- BDD



DAWAN - Reproduction interdite

Intégration Continue

- DDD

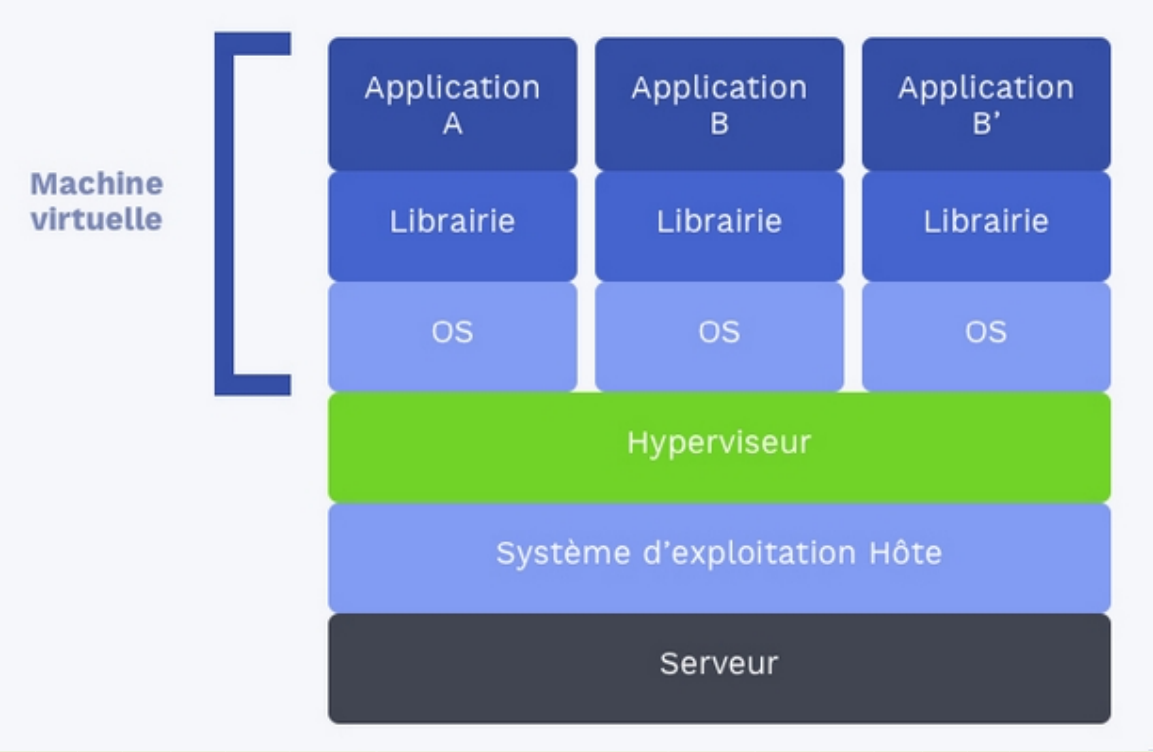
- « Domain Driven Design » ou Conception pilotée par le domaine :
- On augmente encore la BDD en analysant le domaine d'application du produit et on cherche à modéliser le comportement du produit dan son environnement réel
- Adoption d'une nomenclature commune
- Modélisation du contexte, des entités, des évènements, des services…
- Accent mis sur le respect des process métier => identification de microservices
- Recours massif au paradigme de programmation Objet pour la modélisation
- Proche de modèle de conception comme UML

DAWAN - Reproduction interdite

Conteneurisation

- Rappel : virtualisation

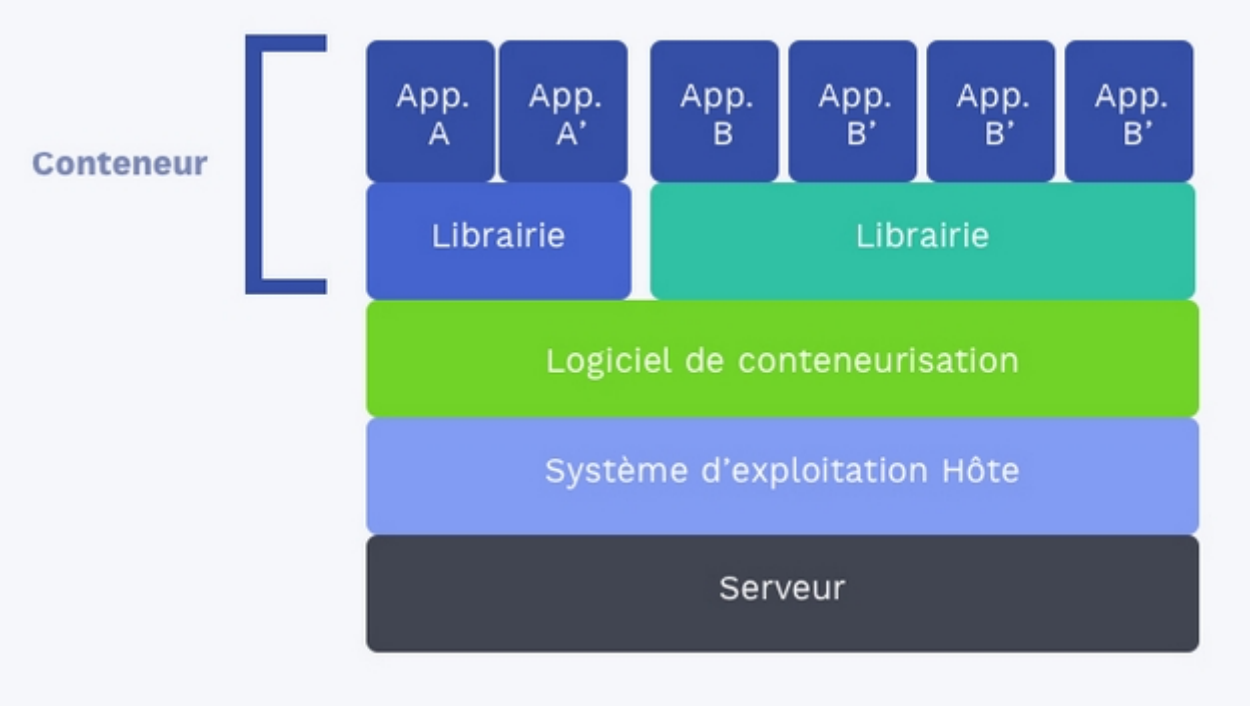
- La virtualisation désigne le partage de ressource d'un hôte entre plusieurs systèmes d'exploitation au moyen d'un logicielle socle nommé hyperviseur



DAWAN - Reproduction interdite

Conteneurisation

- Définition
  - La conteneurisation désigne l'empaquetage d'un programme et de ses dépendances dans un conteneur garantissant l'isolation des processus et des ressources allouées à celui ci, du reste des programmes d'un hôte



DAWAN - Reproduction interdite

Conteneurisation

- Avantages

- Par rapport à la virtualisation, on a nécessairement un gain de performances du fait de l'utilisation directe du noyau de l'OS hôte, et une allocation de ressources dynamique par conteneur
- L'empaquetage du code avec ses dépendances est décrite elle même dans un code qui permet d'unifier les procédures d'installation sur les environnements projet (dev, staging, prod…) => on parle d'infrastructure as code
- Cet empaquetage permet donc de simplifier l'installation d'un environnement de développement pour un nouveau collaborateur, et met en cohérence les environnements Dev et Ops
- Dotés d'interfaces réseaux propres, les conteneurs peuvent communiquer entre eux et avec l'extérieur, permettant de simplifier l'installation de systèmes complexes => architecture microservice

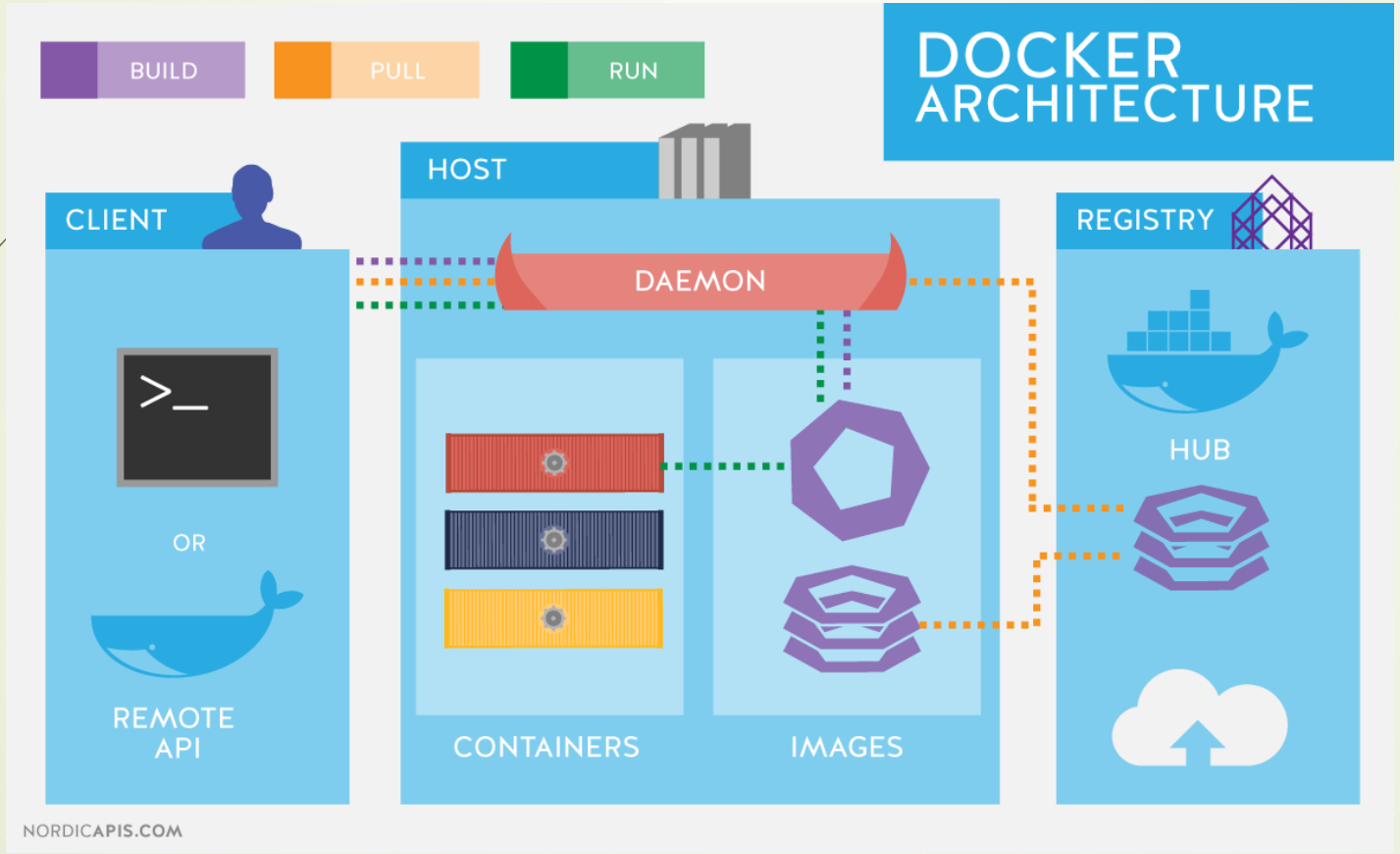
DAWAN - Reproduction interdite

Conteneurisation

- docker
  - Depuis 2012, solution de conteneurisation la plus populaire
  - On écrit dans un fichier Dockerfile la procédure d'installation des prérequis et des dépendances permettant l'exécution d'un code
  - Le fichier Dockerfile est compilé en image docker, équivalent au niveau conteneur à une image disque d'os en vrtualisation
  - Le conteneur, lancé par le démon Docker à partir de l'image, est près à exécuter le code installé

Conteneurisation

- Docker:architecture



DAWAN - Reproduction interdite

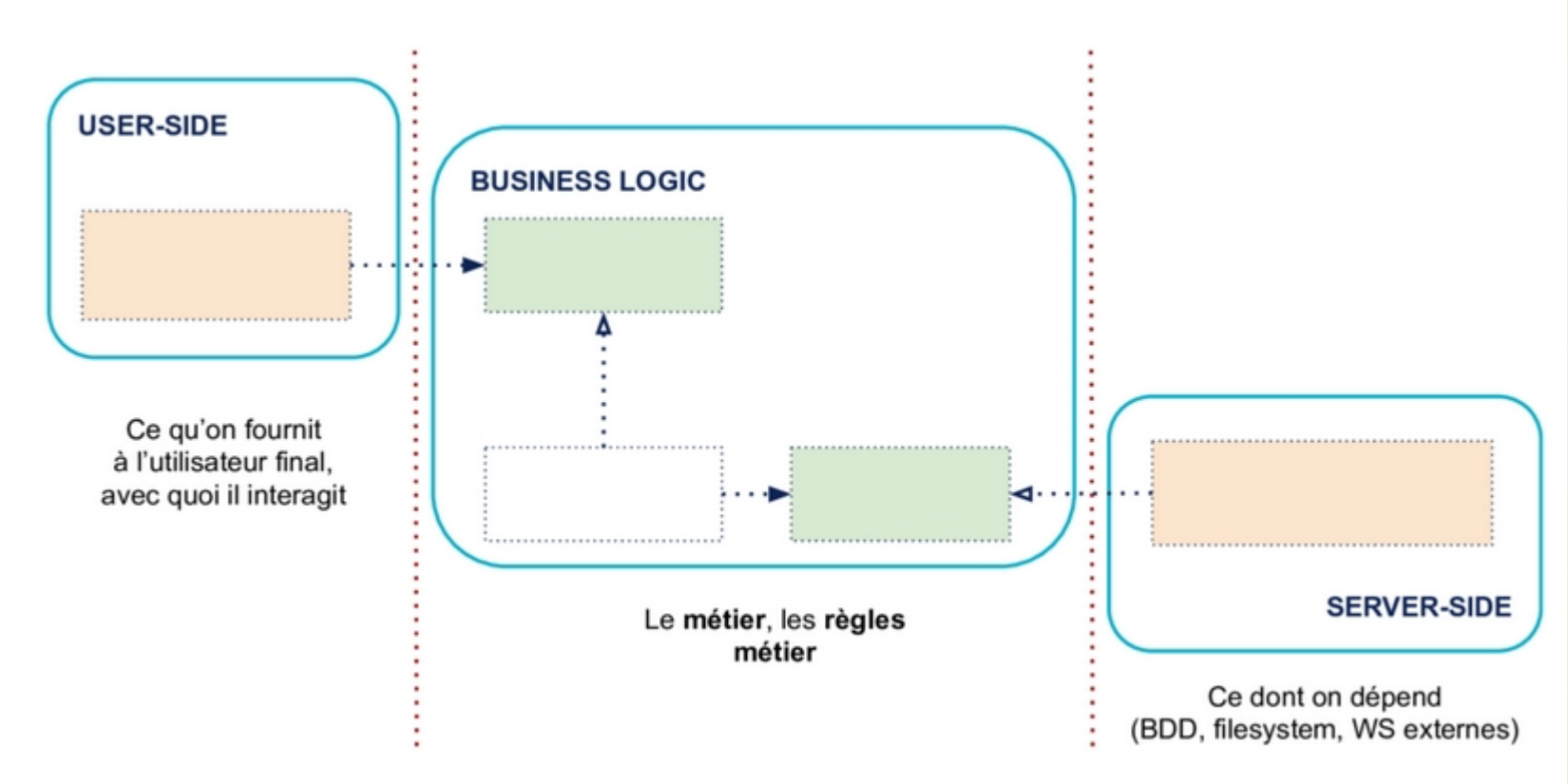
Conteneurisation

- Architecture « hexagonale »
- Principe de séparation des composants d'une application en tois entités :
  - User Side : Gestion de l'interaction de l'utilisateur avec l'application
  - Business Logic : Le code applicatif implémentant les règles métier (représenté par un hexagone)
  - Server Side : Gestion de l'interaction de l'application avec l'infrastructure (bases de données, systèmes de fichiers, webservices)
- Les dépendances entre les entités sont exclusivement à la charge des user et server sides, au moyen d' adapteurs (design pattern) pour que la Business Logic puisse être extraite simplement de son environnement applicatif (test automatiques, revue par des experts métiers)

DAWAN - Reproduction interdite

Conteneurisation

- Architecture « hexagonale »



DAWAN - Reproduction interdite

Conteneurisation

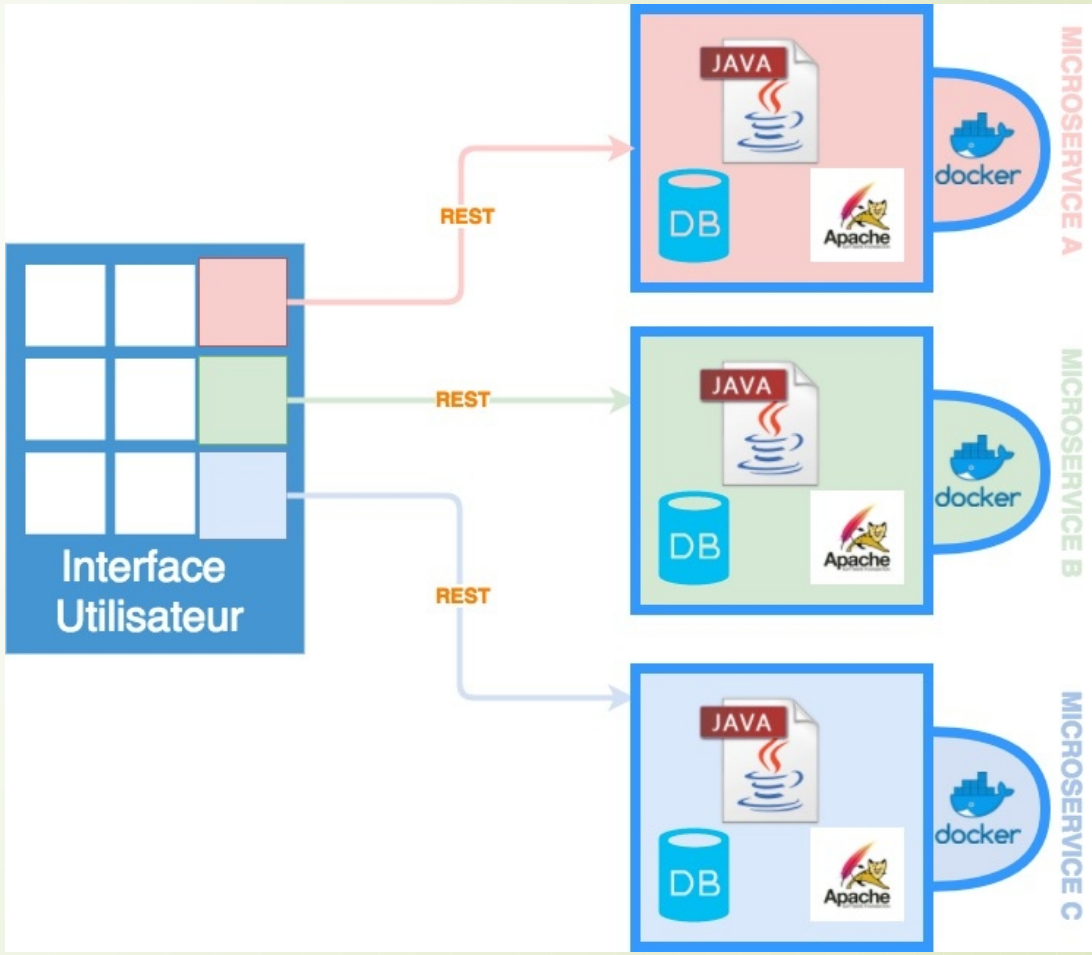
• Architecture microservice

- L'Architecture microservice découpe la logique métier en services indépendants :

- Chaque service forme un architecture héxagonale avec
  - son interface vers l'extérieur (serveur web)
  - sa propre logique fonctionnelle
  - son interface infra (bdd)

- Cette organisation simplifie le développement, et apporte un gain de performances du fait de la parallélisation possibles de l'exécution des services

- La maintenance système est plus délicate, mais facilitée par les outils d'orchestration



Conteneurisation

- Orchestration

- L’orchestration est le processus de gestion du cycle de vie des conteneur distribués sur un pool de ressources serveur (cloud). Ils ont pour fonction d’assurer :

  - L’approvisionnement et les déploiements (via une API)

  - La configuration et la planification (fichiers YAML, Scheduler)

  - L’allocation des ressources

  - La mise à l'échelle du nombre de conteneurs en fonction des charges de travail dans l'infrastructure (Controllers)

  - L’équilibrage de la charge et le routage du trafic (Controllers)

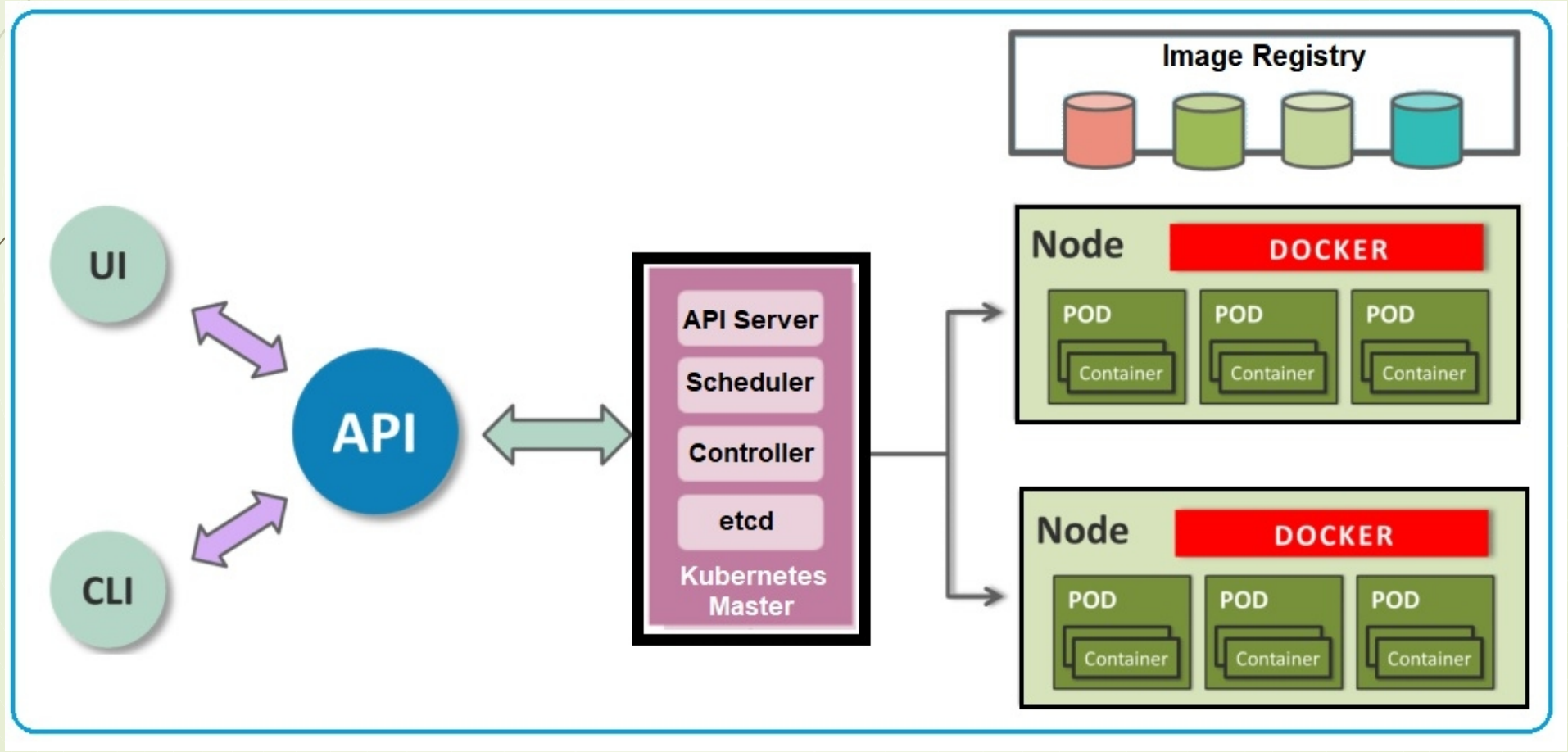
  - La surveillance de l'intégrité des conteneurs (Controllers)

  - La sécurisation des interactions entre les conteneurs (Controllers)

DAWAN - Reproduction interdite

Conteneurisation

- Orchestration : exemple (kubernetes)



Infrastructure as Code

- Définition

- On entend par infrastructure un ensemble de serveurs informatiques en réseau fournissant une somme de ressources :
  - du stockage (espaces disque)
  - de la mémoire vive (RAM)
  - de la capacité de calcul (CPU, GPU)
  - des interfaces réseaux
- L’infrastructure as Code (IaC) signifie la gestion d’une infrastructure IT par des fichiers de configuration

DAWAN - Reproduction interdite

Infrastructure as Code

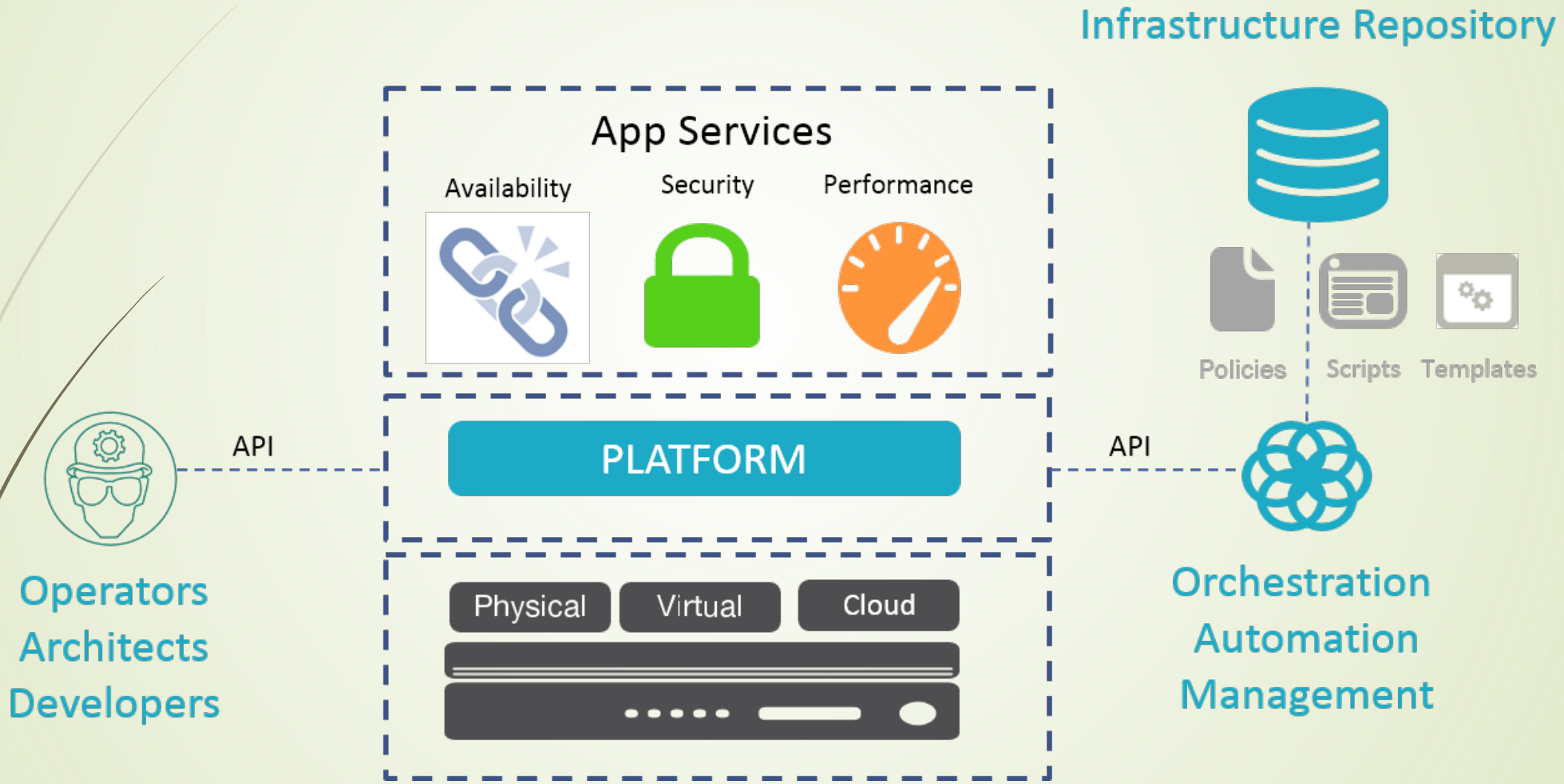
- typologie

- Les solutions d’IaC permettent différentes opérations comme :

  - La gestion de configuration : installations à partir de configuration
  - Le provisioning: allocation dynamique de ressources
  - Le déploiement : transfert des sources sur serveur distant
  - L’orchestration : contrôle du cycle de vie de l’infrastructure

DAWAN - Reproduction interdite

Infrastructure as Code



DAWAN - Reproduction interdite

Infrastructure as Code

- Langages de l'IaC

- Les langages utilisés dans l'Iac sont de trois types :

  - déclaratifs : on décrit l'état final souhaité de la configuration

  - impératif : on décrit les étapes successives pour arriver à l'état final

  - intelligent : on laisse la solution analyser le système pour déduire l'état final optimal

- l'approche déclarative est la plus populaire, permettant de fixer l'état voulu, et garantissant l'idempotence d'une opération ,à savoir l'invariabilité du résultat quelque soit le nombre de répétition de ladite opération

DAWAN - Reproduction interdite

Infrastructure as Code

- Immutabilité de l’infrastructure

- Une infra est mutable lorsqu’on peut modifier l’allocation de ressource à la volée ; cette solution permet des changement rapides, mais éloigne la configuration réelle de celle décrite dans l’IaC => décohérence
- Une infra est immutable lorsqu’on ne peut la modifier une fois provisionnée qu’en la redéfinissant totalement ; cela dit le passage par le code garantit un redépoiement rapide. C’est donc la voie privilégiee pour pratiquer l’IaC

DAWAN - Reproduction interdite

Infrastructure as Code

- Avantages

- Gain de temps et réduction des erreurs par automatisation des installations et mise à jours
- Reproductibilité de l’infrastructure à l’identique en cas de crash matériel
- Scalabilité de l’infrastructure simplifiée par la variabilisation des paramètres dans des templates de fichiers de configuration
- L’environnement IaC devient le référentiel commun des équipes Dev et Ops (Dockerfile, jenkinsfile, *.yml)
- IaaS « Infrastructure As A Service » : On peut piloter les fichiers de configuration à distance au moyen d’API

DAWAN - Reproduction interdite

Monitoring

à répondre à la question « que se passe-t-il ? »

- Définitions

- Le « monitoring » ou surveillance ou supervision, désigne la capacité d'un SI à détecter et comprendre les problèmes qui peuvent survenir

- L'Observabilité est la capacité à répondre à la question « que se passe-t-il ? »
  - observabilité bas niveau : consommations ressources, intégrité des clusters…
  - observabilité applicative : bugs, performances

- L'Alerting est la capacité d'un système à informer en temps réel de l'occurence d'un problème

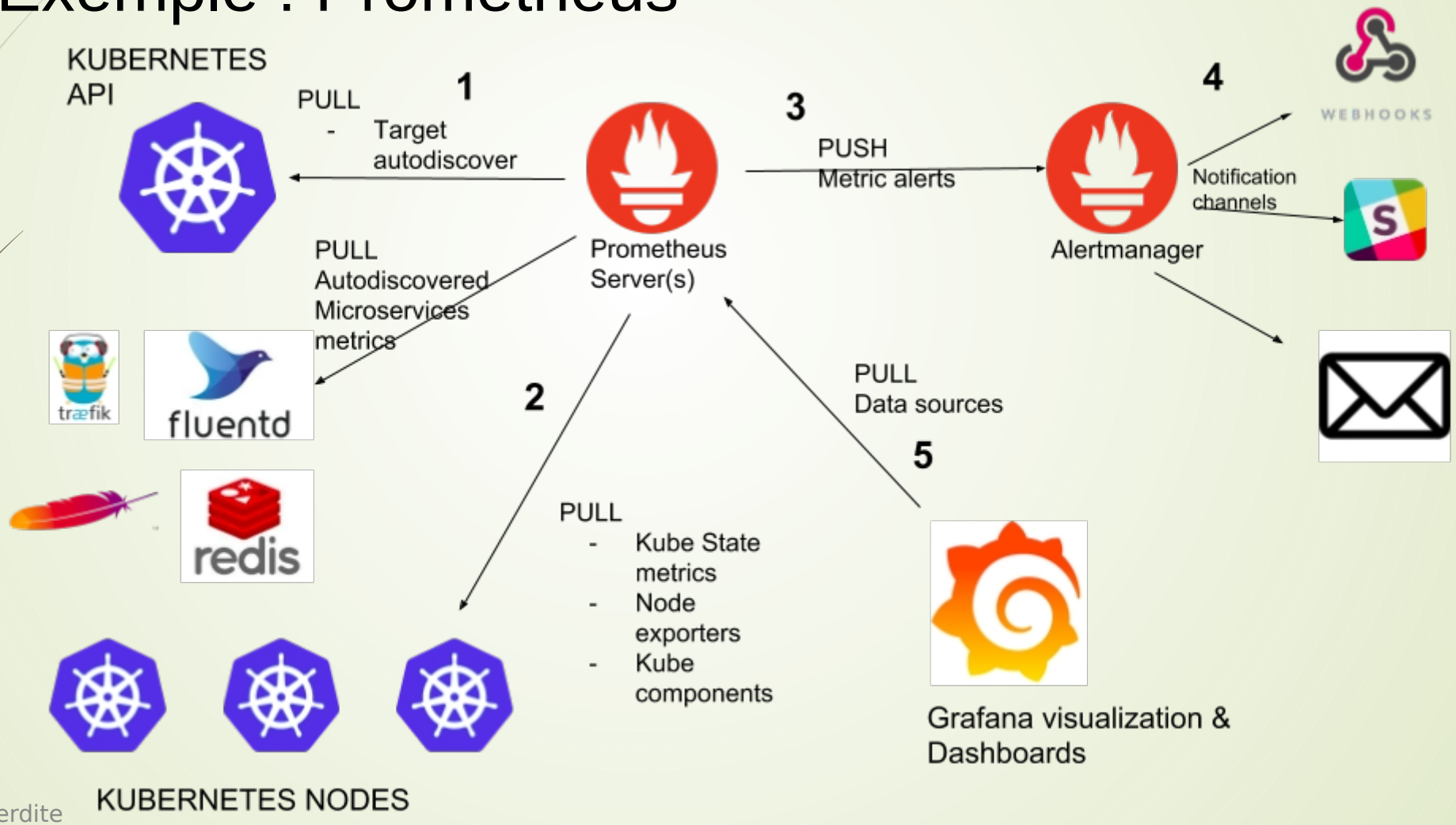
- Exemples de solutions de monitoring
  - Nagios (historique)
  - ELK (Elastic Search / Logstash / Kibana, orienté recherche de métrique)
  - Prometheus / Grafana (orienté microservices)
  - Datadog

DAWAN - Reproduction interdite

Monitoring

à répondre à la question « que se passe-t-il ? »

- Exemple : Prometheus



Monitoring

à répondre à la question « que se passe-t-il ? »

- Sélection des indicateurs

- Le monitoring permet le suivi de l'évolution des KPIs définis au niveau stratégique
- Certaines technologies utilisent des agents appelés exporteurs qui sont spécialisés dans le prélèvement des métriques d'une solution logicielles particulières.
- Une fois l'ensemble des métriques prélevées, on peut les transférer à des outils de visualisation de données capable d'offrir une présentation graphique pertinente de ces métriques et de leur corrélation, à visée de reporting

DAWAN - Reproduction interdite

Monitoring

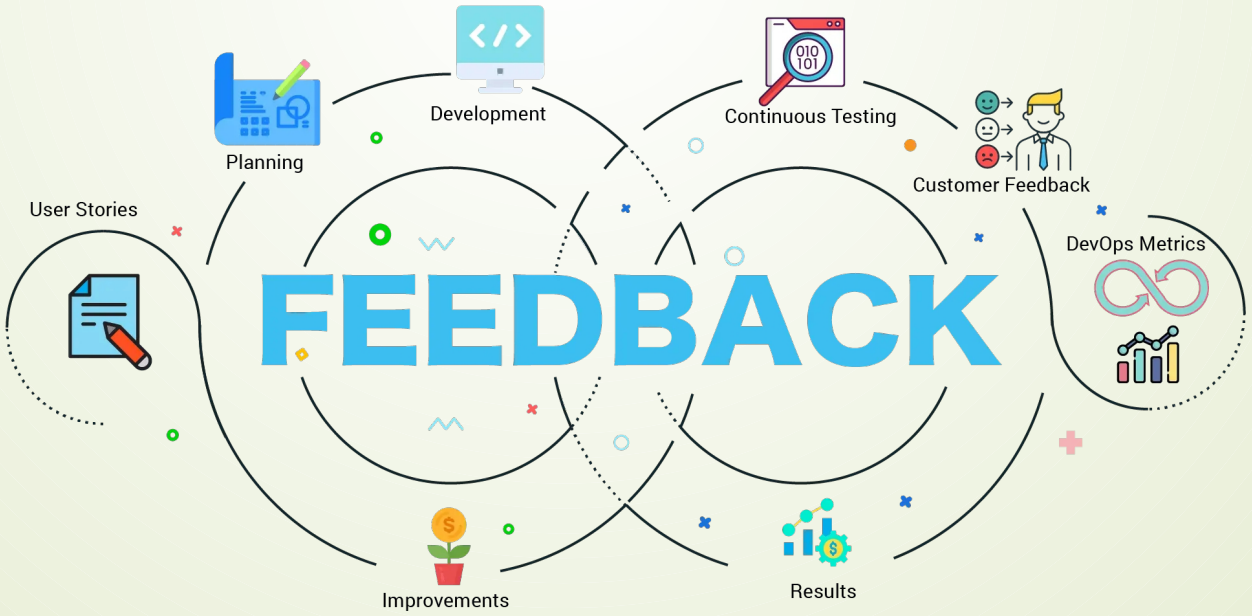
à répondre à la question « que se passe-t-il ? »

- Monitoring et DevOps

- Le monitoring est un excellent outil de communication entre Devs et Ops. Il aide à :
  - aligner leurs enjeux
  - découvrir les corrélations entre les problèmes qu'ils rencontrent
  - avoir des discussions factuelles.

  => outil privilégié pour construire une culture de collaboration avec des objectifs communs.

- Le monitoring est aussi intimement lié au DevOps car il est l'élément structurant de la Feedback Loop.



DAWAN - Reproduction interdite

Déploiement continu

- Définition

Pratiques et solutions logicielles permettant l’automatisation du déploiement d’une solution d’un environnement vers un autre en fonction du résultat d’un processus d’intégration continue

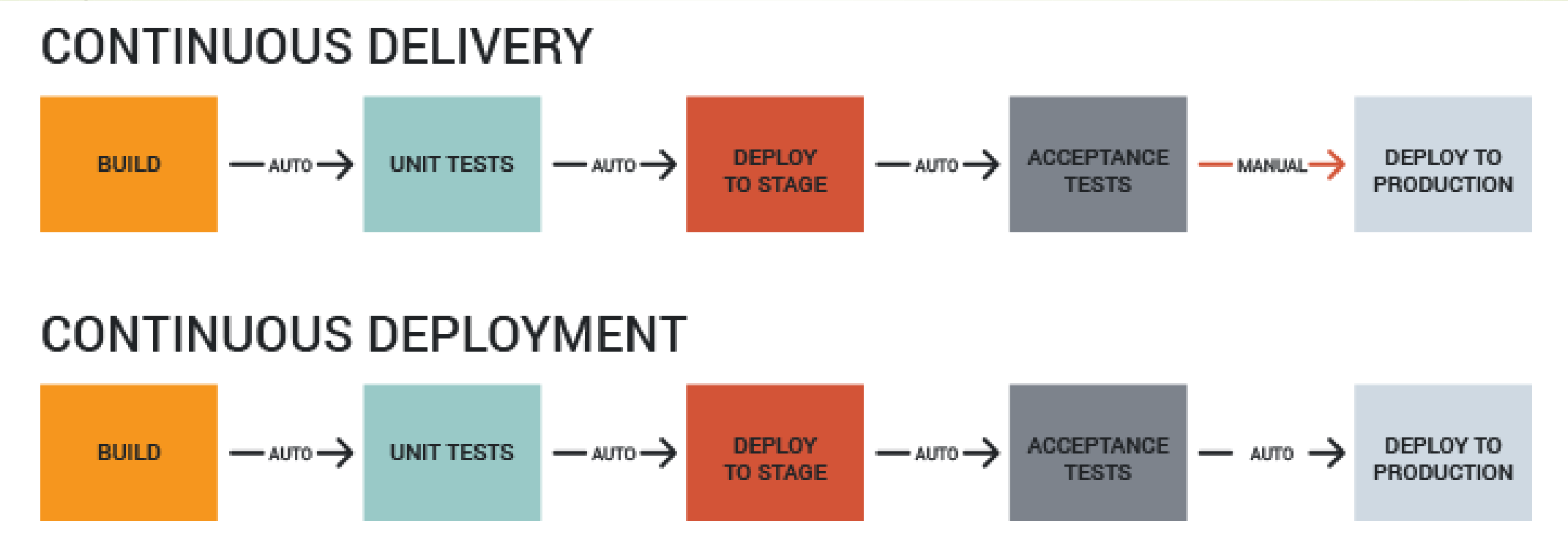
Exemples :

- Gestion de configurations : automatisation des installations / mises à jour
- Orchestration : télé-distribution et allocation automatique des ressources
- Gestion d’un cloud : « Bare Metal », Machines virtuelles, Conteneurs

DAWAN - Reproduction interdite

Livraison vs Déploiement

Un livrable est un code validé par un processus d'intégration continue et qui est démontrable au client - donc potentiellement déployable



DAWAN - Reproduction interdite

Livraison vs Déploiement

- Les stratégies de déploiement

- « Big Bang » : Déploiement de l’intégralité des fonctionnalités d’une version, à la totalité des utlisateurs => très risqué, permet cependant d’évaluer les intéractions des différents composants entre eux. Souvent la marque d’un projet en dehors des délais.

- Pilote : Déploiement de l’intégralité du code à un public restreint : équipe, client, beta testeurs => permet de tester le déploiement en sécurité

- Incrémental : Déploiement progressif de lots fonctionnels formant la version => permet une meilleure visibilité et compréhensiondu produit par le client, mais masque d’éventuelles régrssions liées aux intéractions de composants

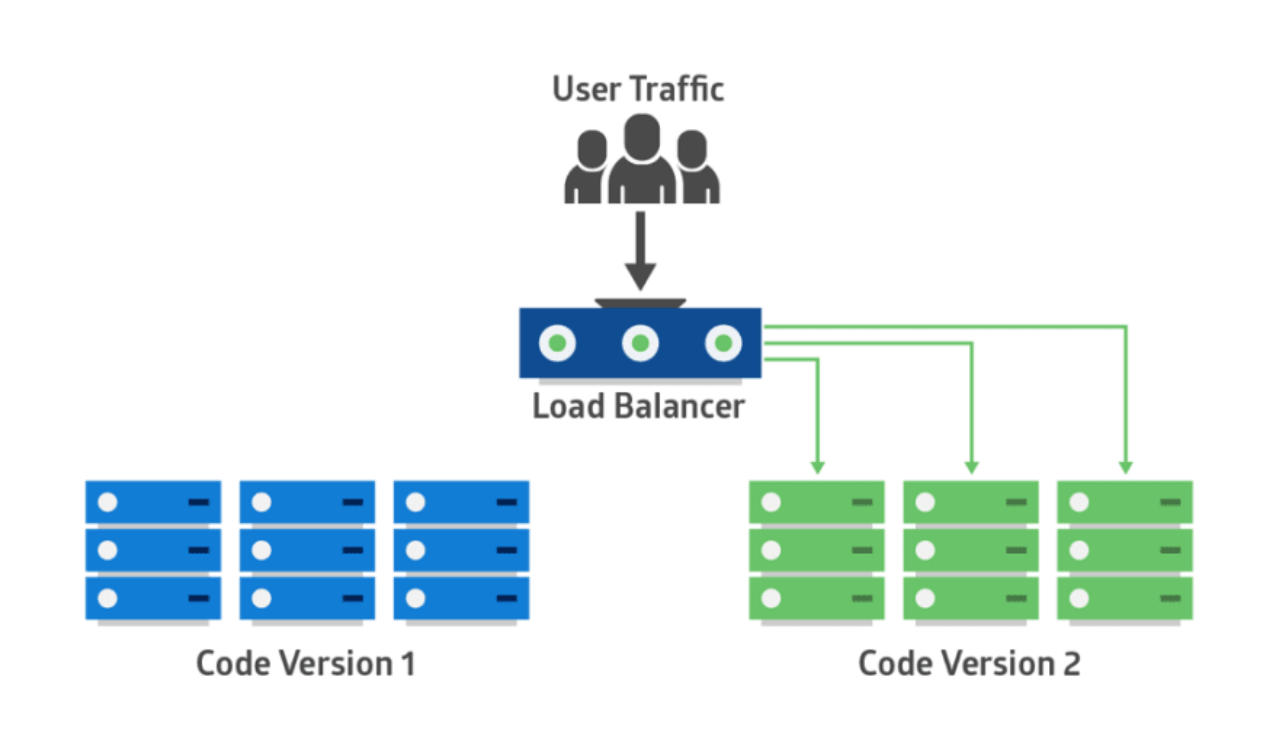
- Combinée : approche croisée des trois précédentes

DAWAN - Reproduction interdite

Déploiement Continu

- Déploiement bleu / vert

- Le déploiement bleu/vert est un modèle de publication d'application qui permet de transférer progressivement le trafic utilisateur au moyen d'un équilibreur de charge, depuis la version antérieure d'une application ou d'un microservice vers une nouvelle version pratiquement identique, ces deux versions s'exécutant en même temps dans l'environnement de production.



DAWAN - Reproduction interdite

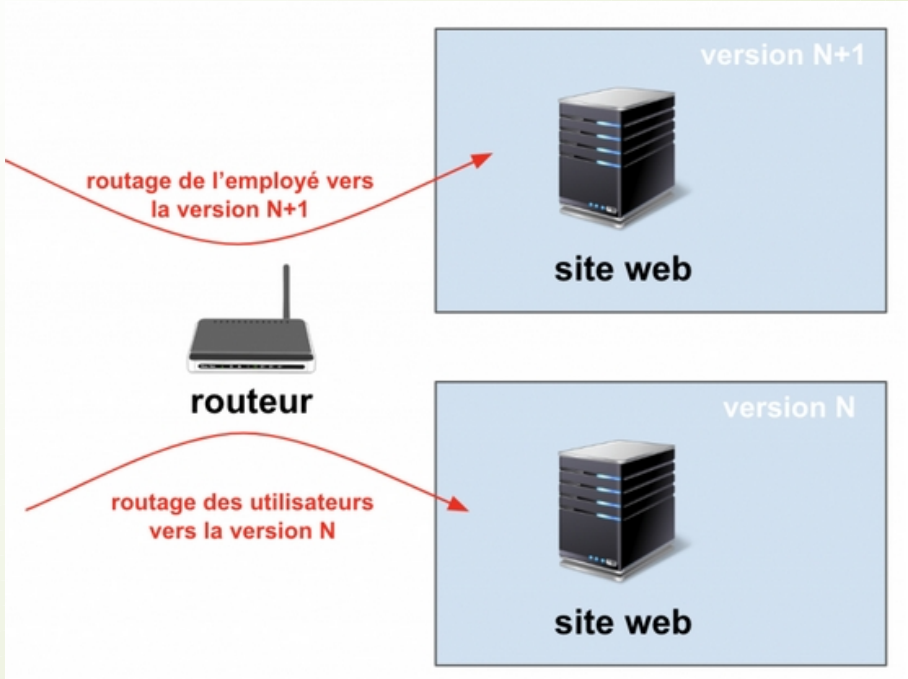
Déploiement Continu

- Déploiement Canari

- Le déploiement canari est un mode de déploiement proche du déploiement bleu / vert, à ceci près que la nouvelle version est proposée à une cible d'utilisateurs sélectionnés. La plupart du temps, il s'agît des employés eux mêmes qui vont pouvoir auditer la nouvelle version en production, relever ses métriques, pour confirmer sa validité

- Ce type d'approche est utile pour tester des idées innovantes en conditions réelles, sans risquer de nuire à l'expérience utilisateur => « lean startup »

- Il est possible d'automatiser l'audit du canari ainsi que l'éventuel « Rollback », ou retour en arrière de version, si les attentes ne sont pas satisfaites. On parle alors de « Cluster Immune System »



Déploiement Continu

- Mécanismes applicatifs

- On parle de « Feature toggles »lorsqu'il est possible de désactiver en production une fonctionnalité défaillante ou non satisfaisante par une option de configuration, sans redéployer de code (depuis un Back Office généralement)

- Généralisé à l'ensemble des fonctionnalités d'une nouvelle version, on parle alors de « dark launch »

DAWAN - Reproduction interdite