De DevOps vers DevSecOps???

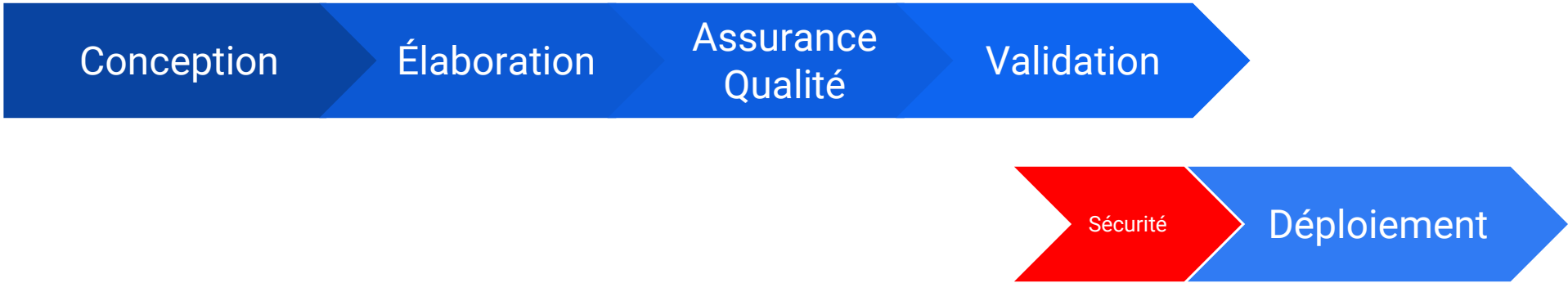
Martin Sauvé
Architecte principal
13 novembre, 2019
msauve@redhat.com

“Depuis plus de 20 ans, la sécurité n’est considérée qu’à la toute fin du cycle de livraison”

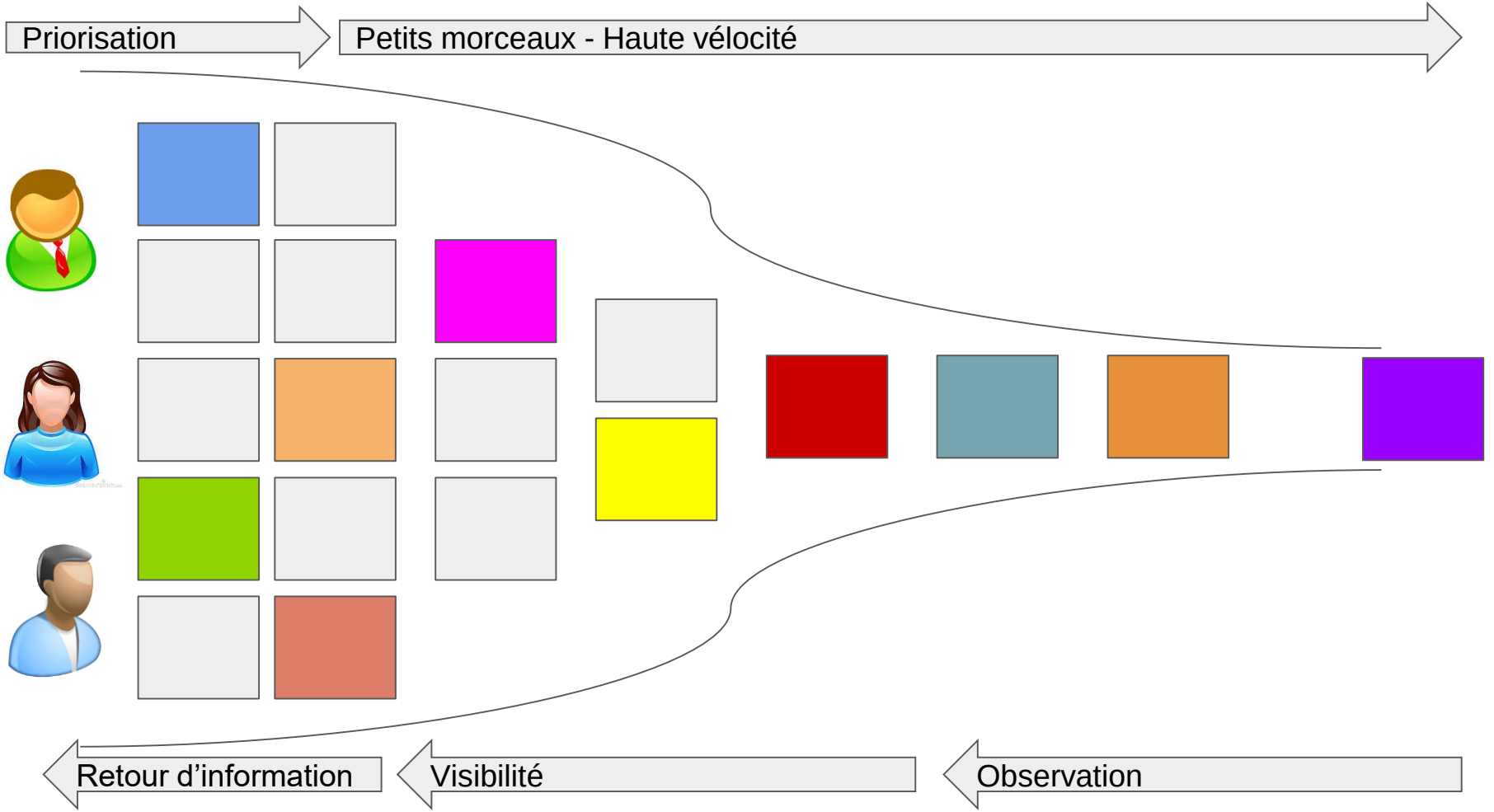
Anonyme


Conception
Élaboration
Assurance Qualité
Validation
Déploiement

Sécurité ?

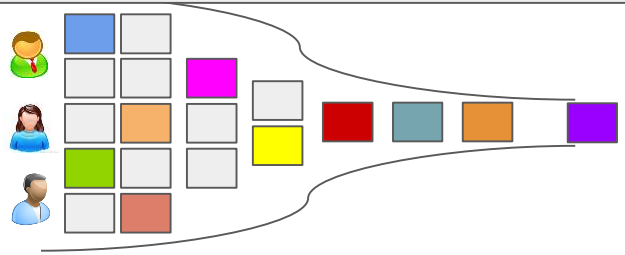

Conception
Élaboration
Assurance Qualité
Validation
Sécurité
Déploiement

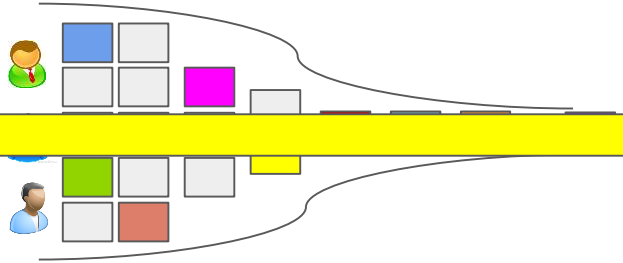

Conception
Élaboration
Assurance Qualité
Validation
Sécurité


Priorisation
Petits morceaux - Haute vélocité
Retour d’information
Visibilité
Observation

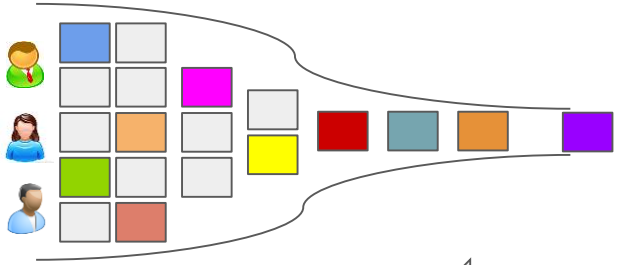
Priorisation

Petits morceaux - Haute vélocité





Automatisation



Retour d’information

Visibilité

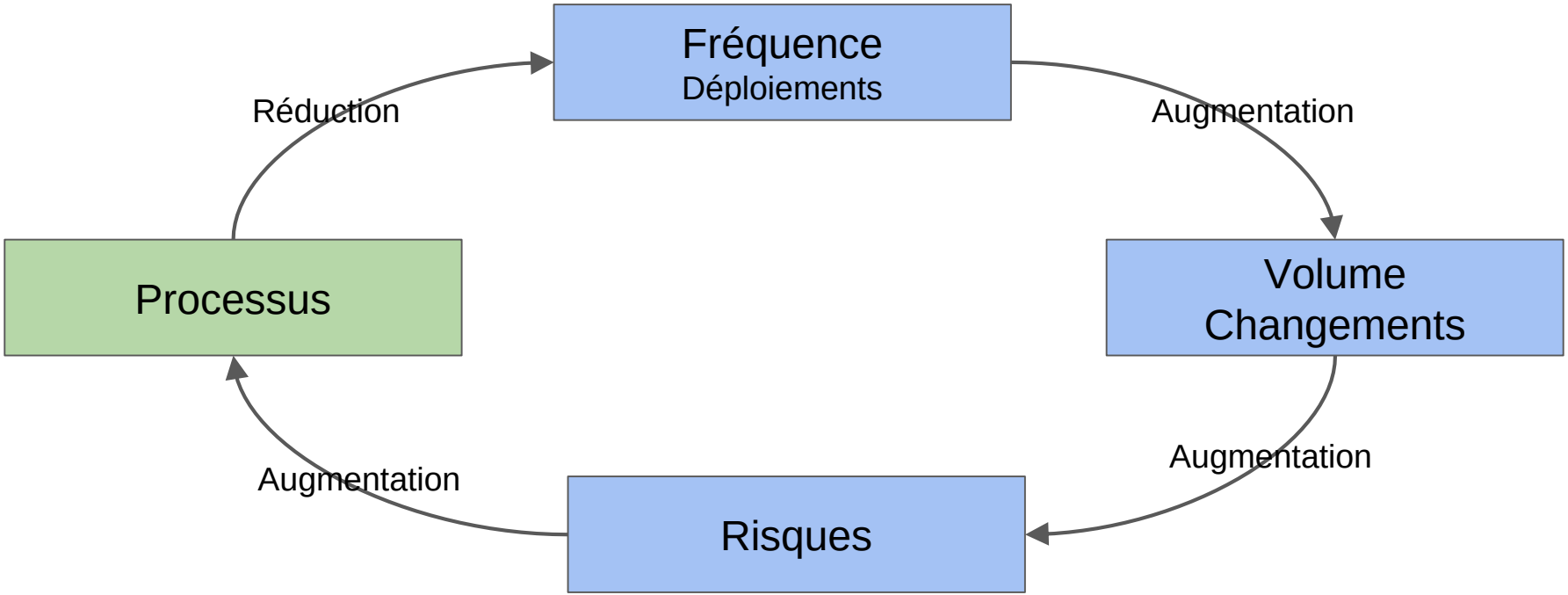
Observation

Source: IT Revolution, DevOps Enterprise abstract word cloud, 2016

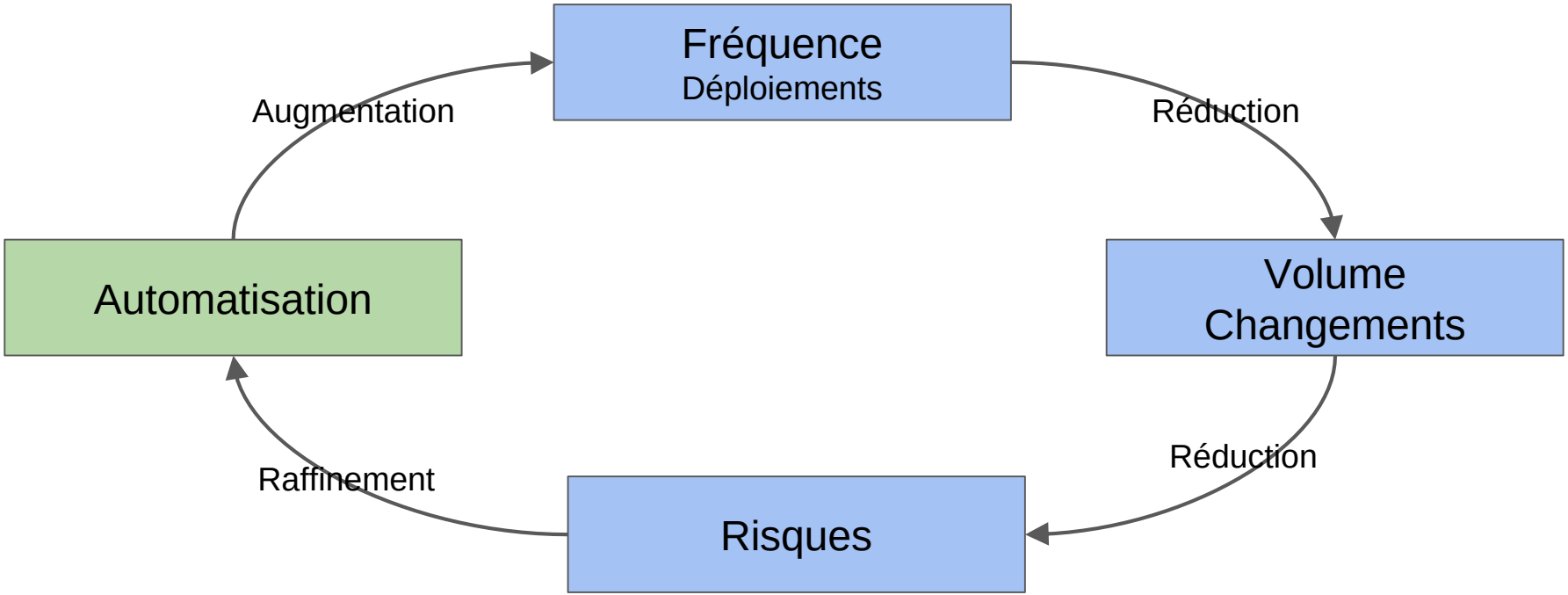
Source: IT Revolution, DevOps Enterprise abstract word cloud, 2016

Source: IT Revolution, DevOps Enterprise abstract word cloud, 2016

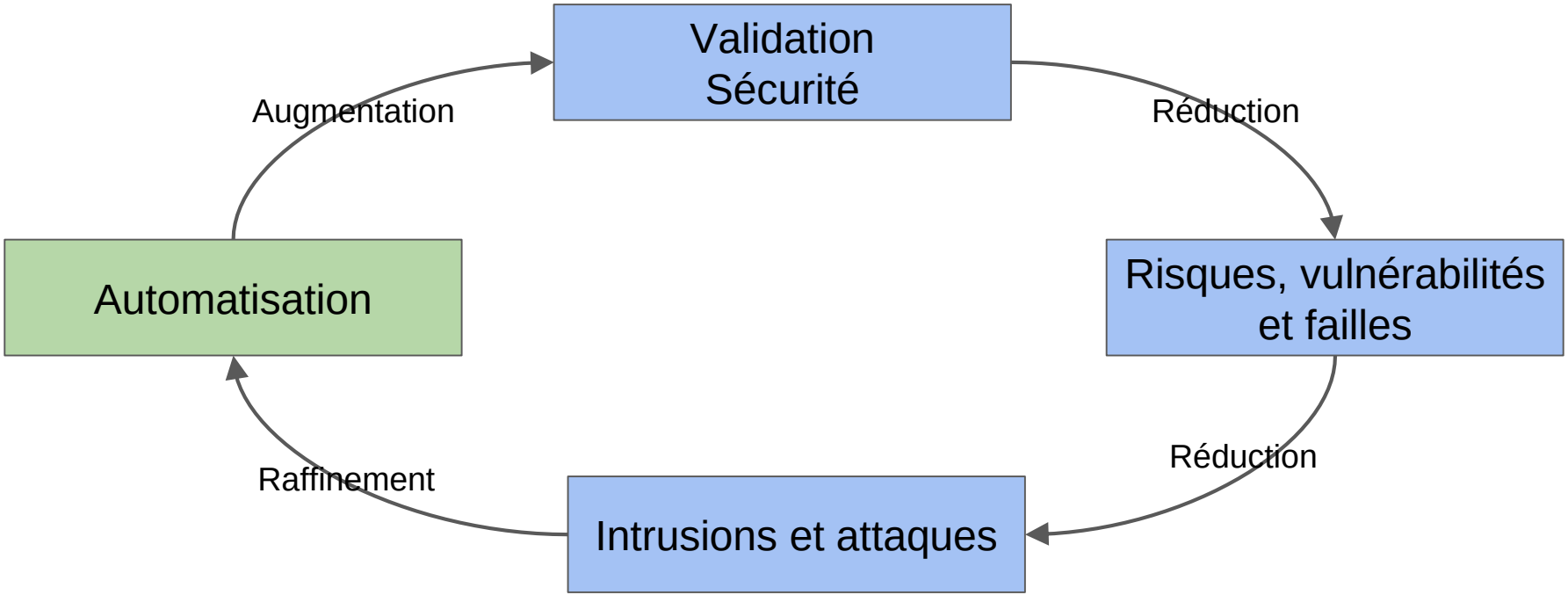
Approche Traditionnelle

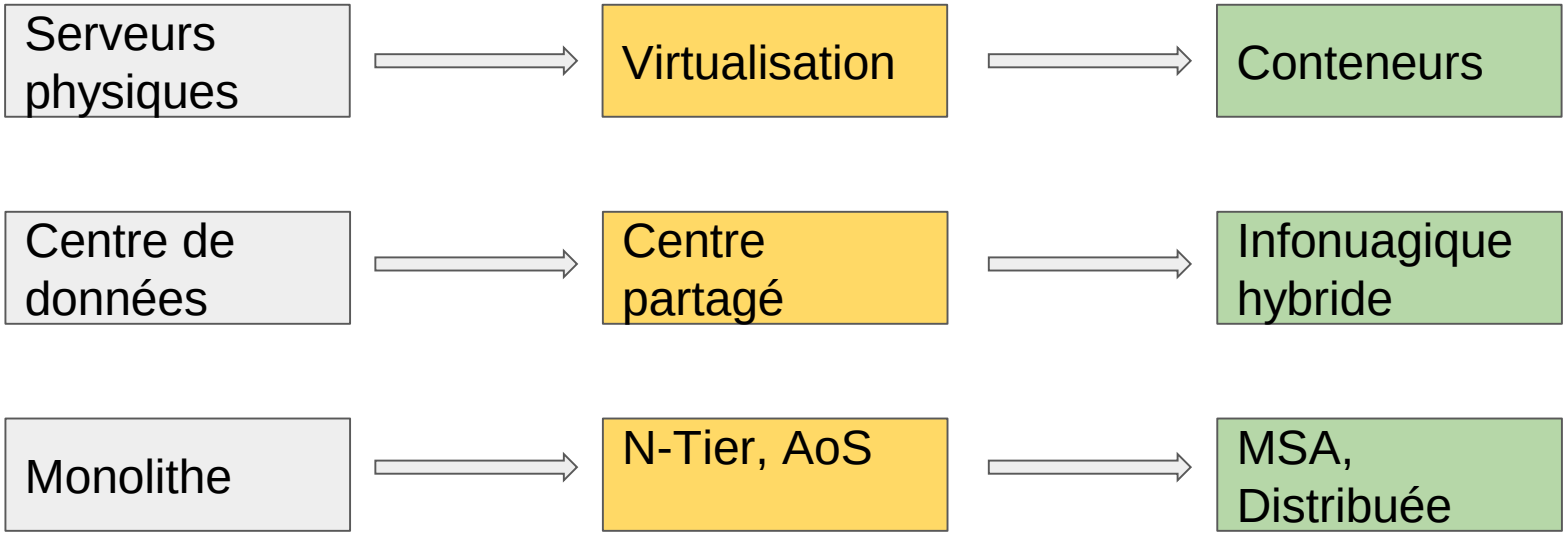


Approche DevOps



Approche Dev(Sec)Ops




Serveurs physiques
Virtualisation
Conteneurs
Centre de données
Centre partagé
Infonuagique hybride
Monolithe
N-Tier, AoS
MSA, Distribuée

- Haute vélocité
- Parallélisme des équipes
- Architecture hautement distribuée
- Environnements hétérogènes
- Culture négligeant la sécurité

- Haute vélocité
- Parallélisme des équipes
- Architecture hautement distribuée
- Environnements hétérogènes
- Culture encourageant l’innovation

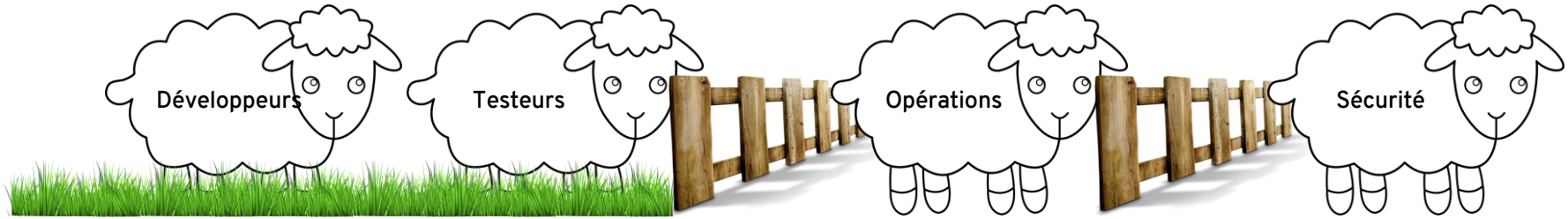
DevSecOps

Comment réussir ?

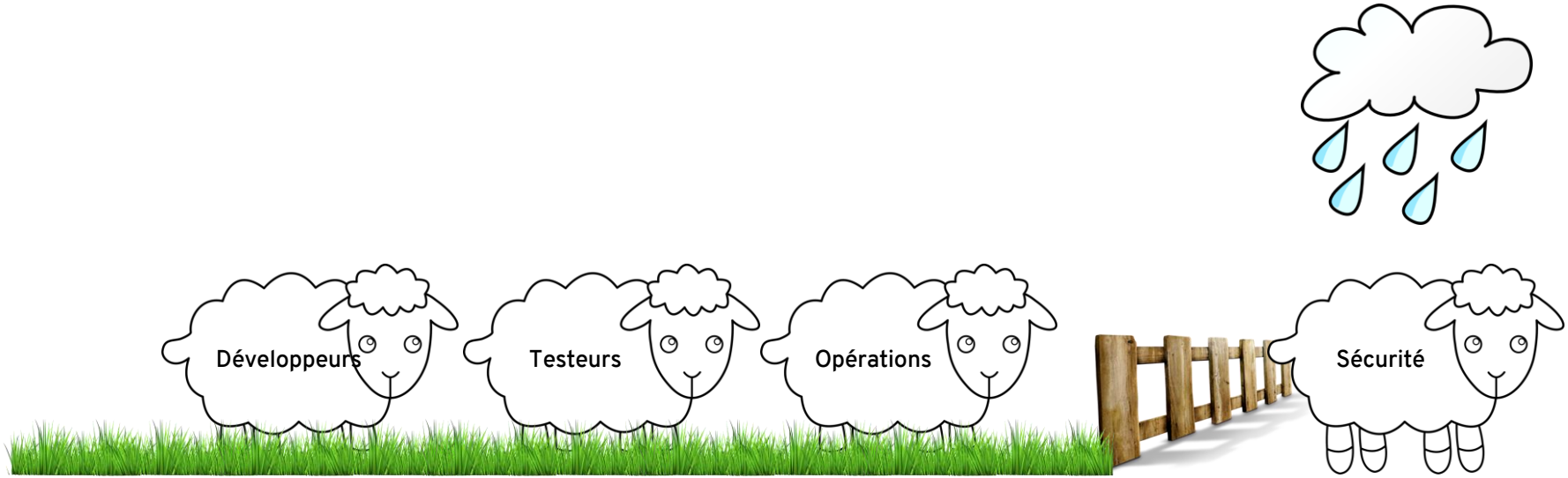

Développeurs
Testeurs
Opérations
Sécurité


Développeurs
Testeurs
Opérations
Sécurité

Agile



Transformation Numérique


Développeurs
Testeurs
Opérations
Sécurité

~~DevTestOpsSec~~?

DevOps

Culture + Processus + Technologie

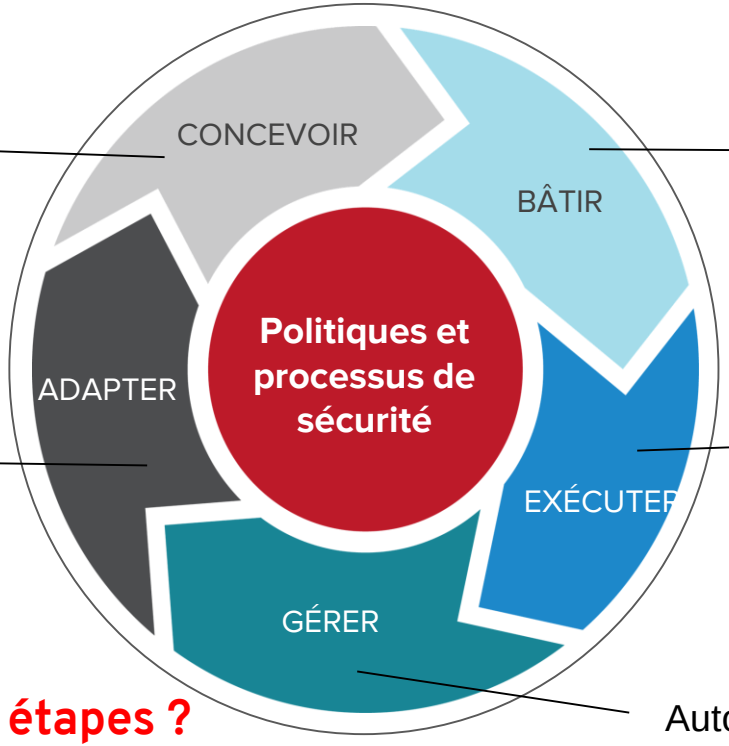
PROCESSUS

SÉCURITÉ EN CONTINU

et intégrée au cycle de vie de vos applications

Définition de la gouvernance et besoin en sécurité

Sécurité intégrée depuis le début; pas ajoutée



Révision, mise à jour, correctifs

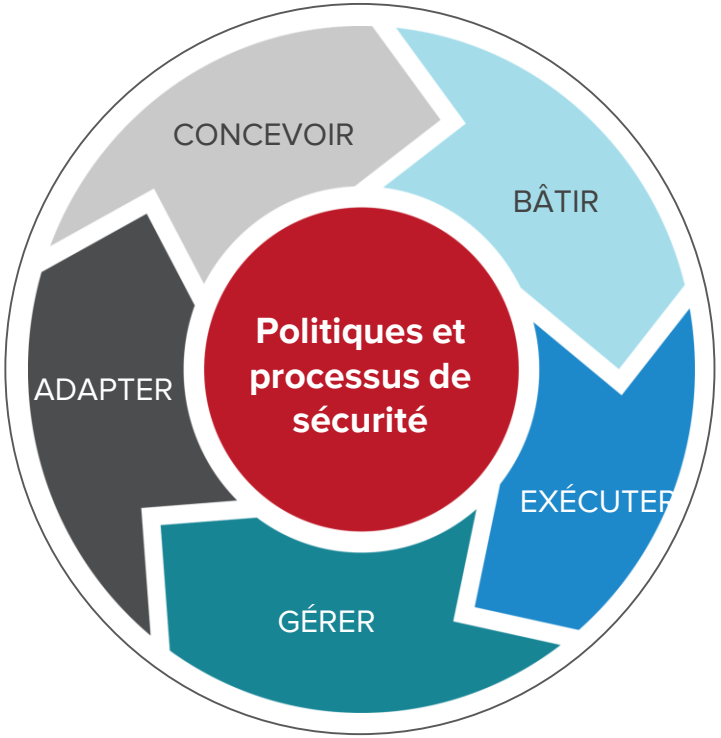
Déploiement sur des plateformes avec des capacités de sécurité élevées

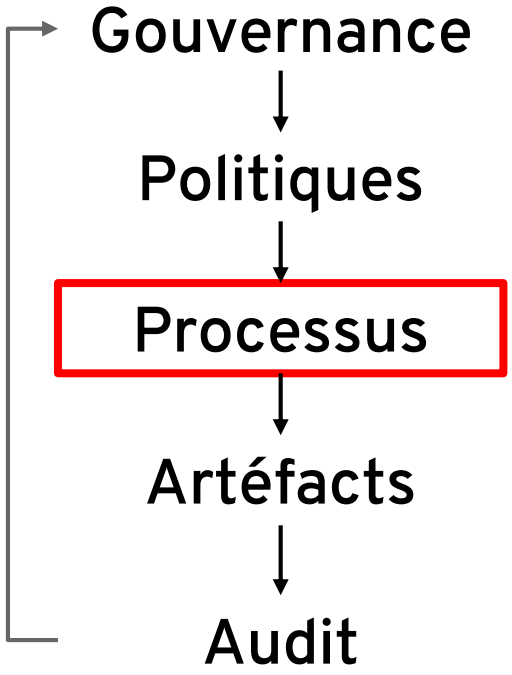
Est-ce DevOps sans ces étapes ?

Automatisation de la sécurité et la conformité

SÉCURITÉ EN CONTINU

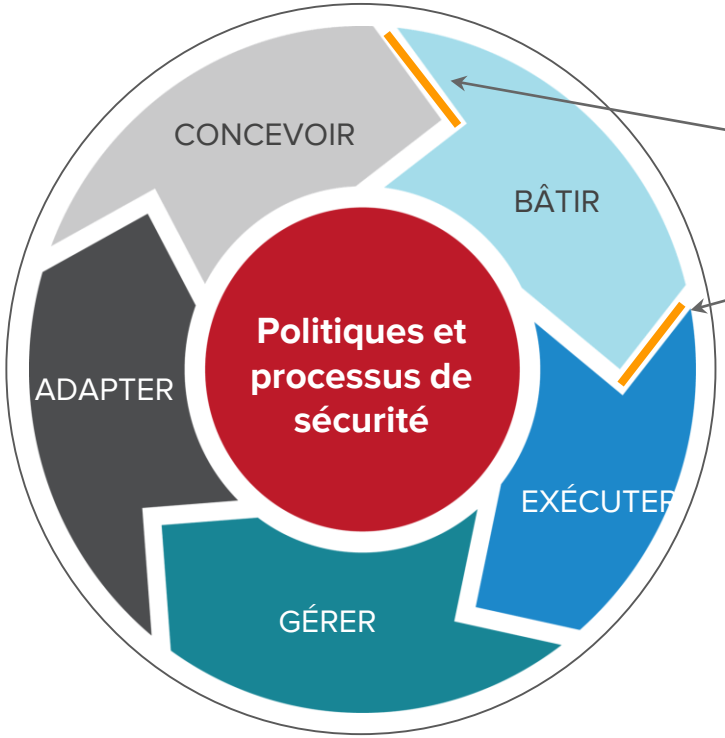
et intégrée au cycle de vie de vos applications





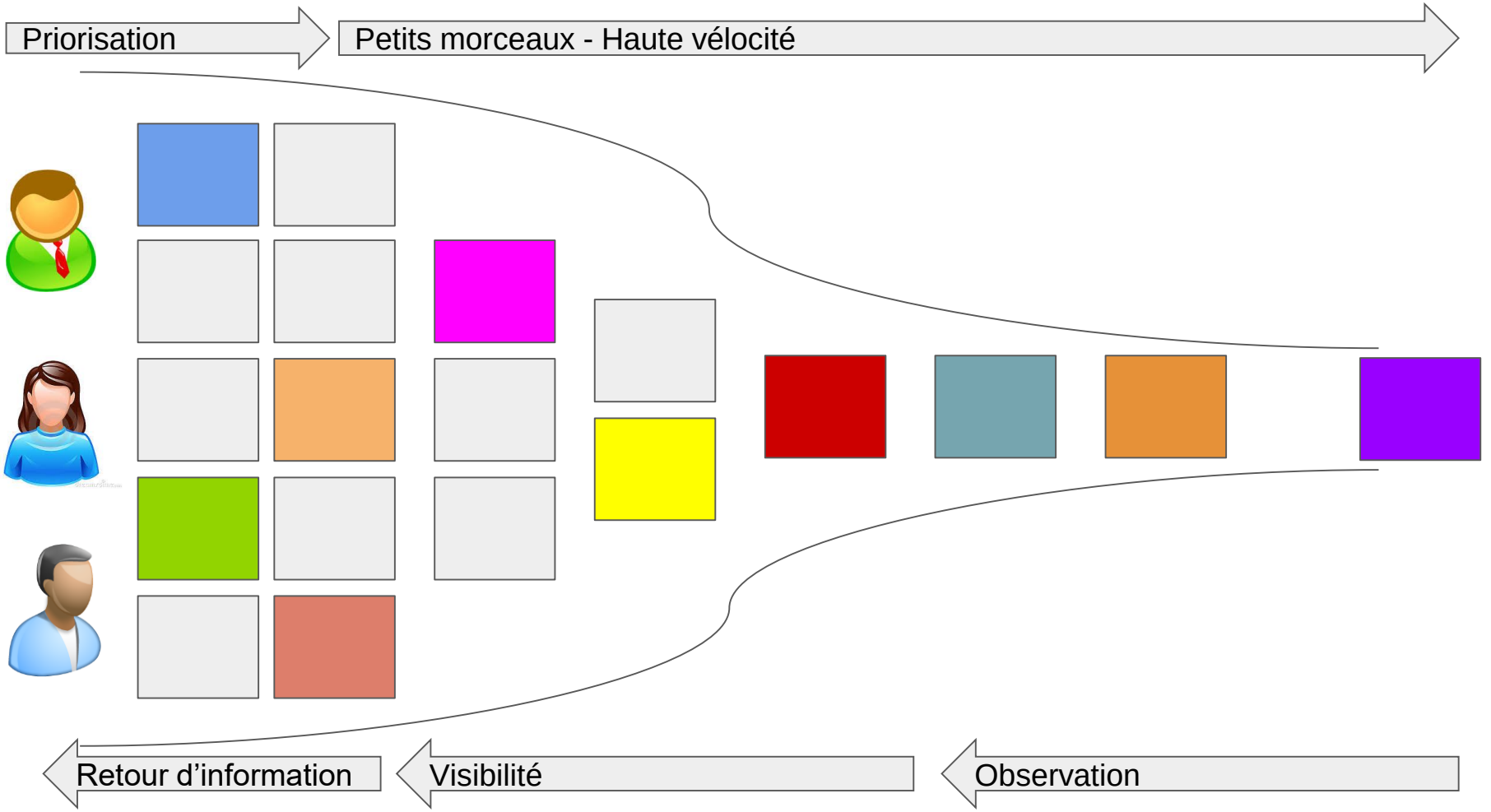
SÉCURITÉ EN CONTINU

Example

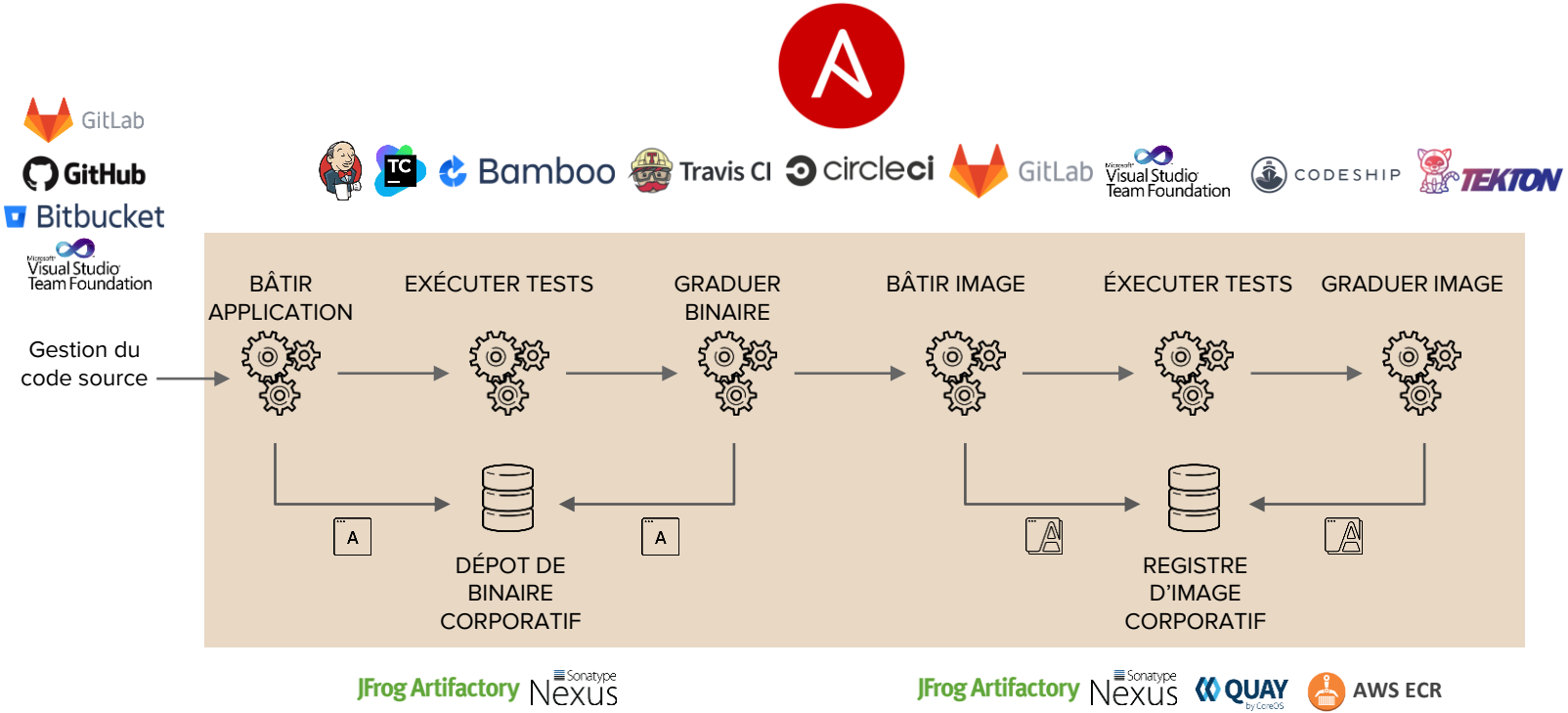


Points de contrôle

- Membranes perméables
- Automatiser
- Validation des politiques
- Exercise de la gouvernance
- Log, audit


Priorisation
Petits morceaux - Haute vélocité
Retour d’information
Visibilité
Observation

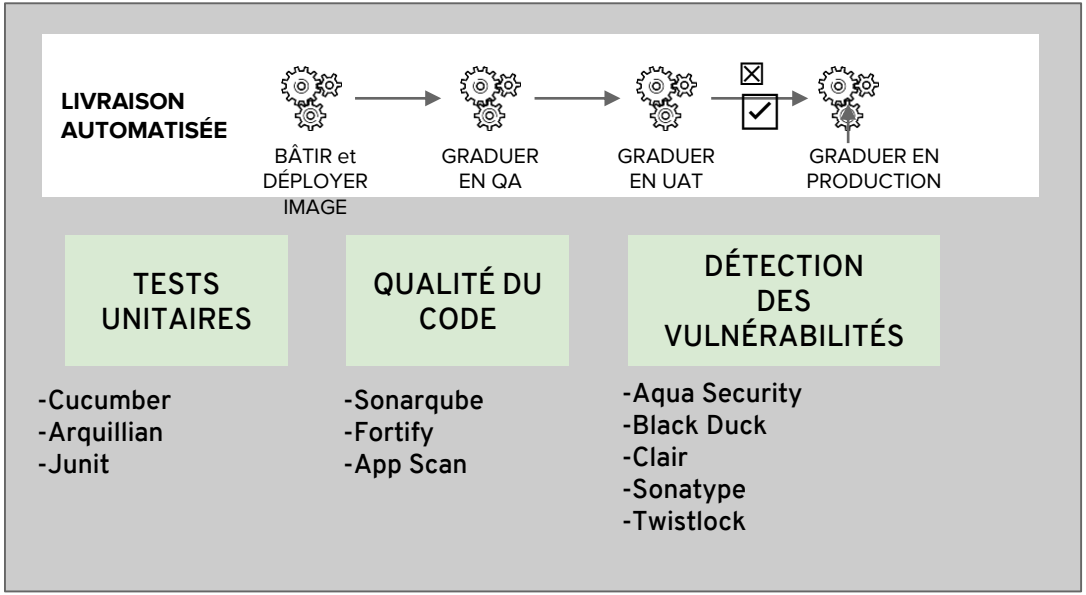
AUTOMATISATION DE LA LIVRAISON



Red Hat

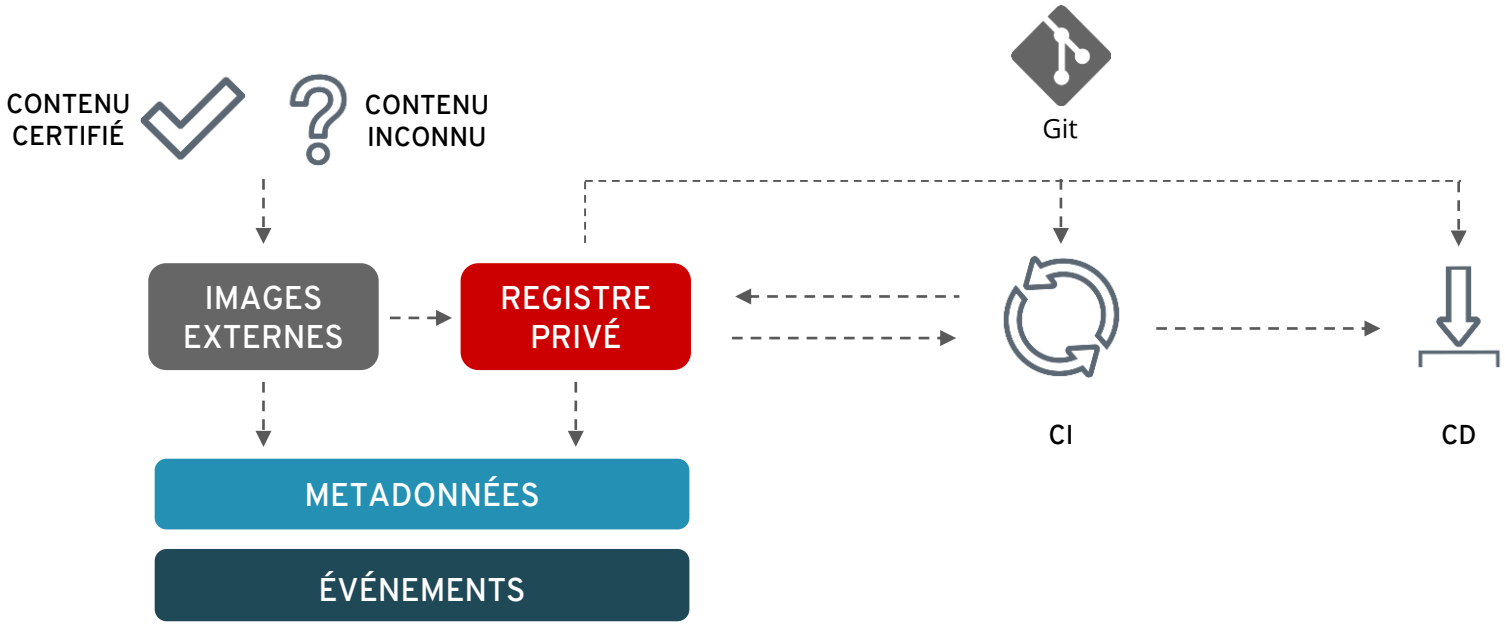
AUTOMATISATION DE LA SÉCURITÉ (CI/CD)

- Tests de sécurité intégrés à la livraison
- Utilisation de politiques automatisées comme marqueur
- Signature des images



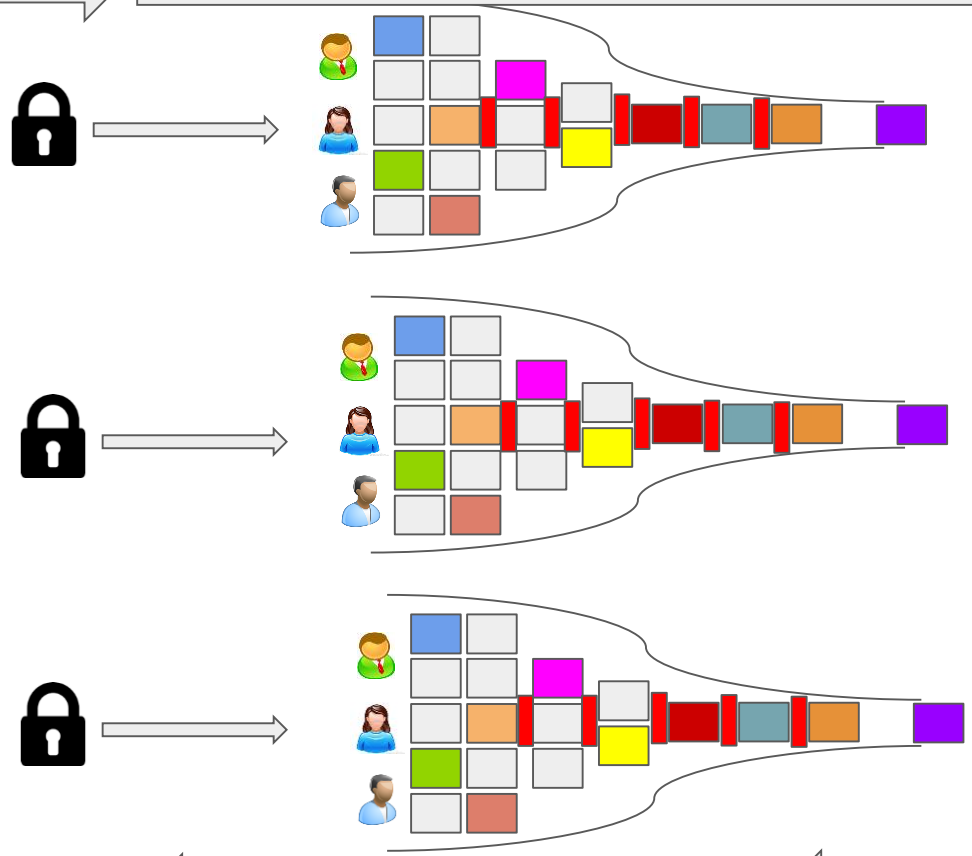
SÉCURITÉ ET AUTOMATISATION

La confiance est temporelle; bâtir et déployer au besoin



Priorisation

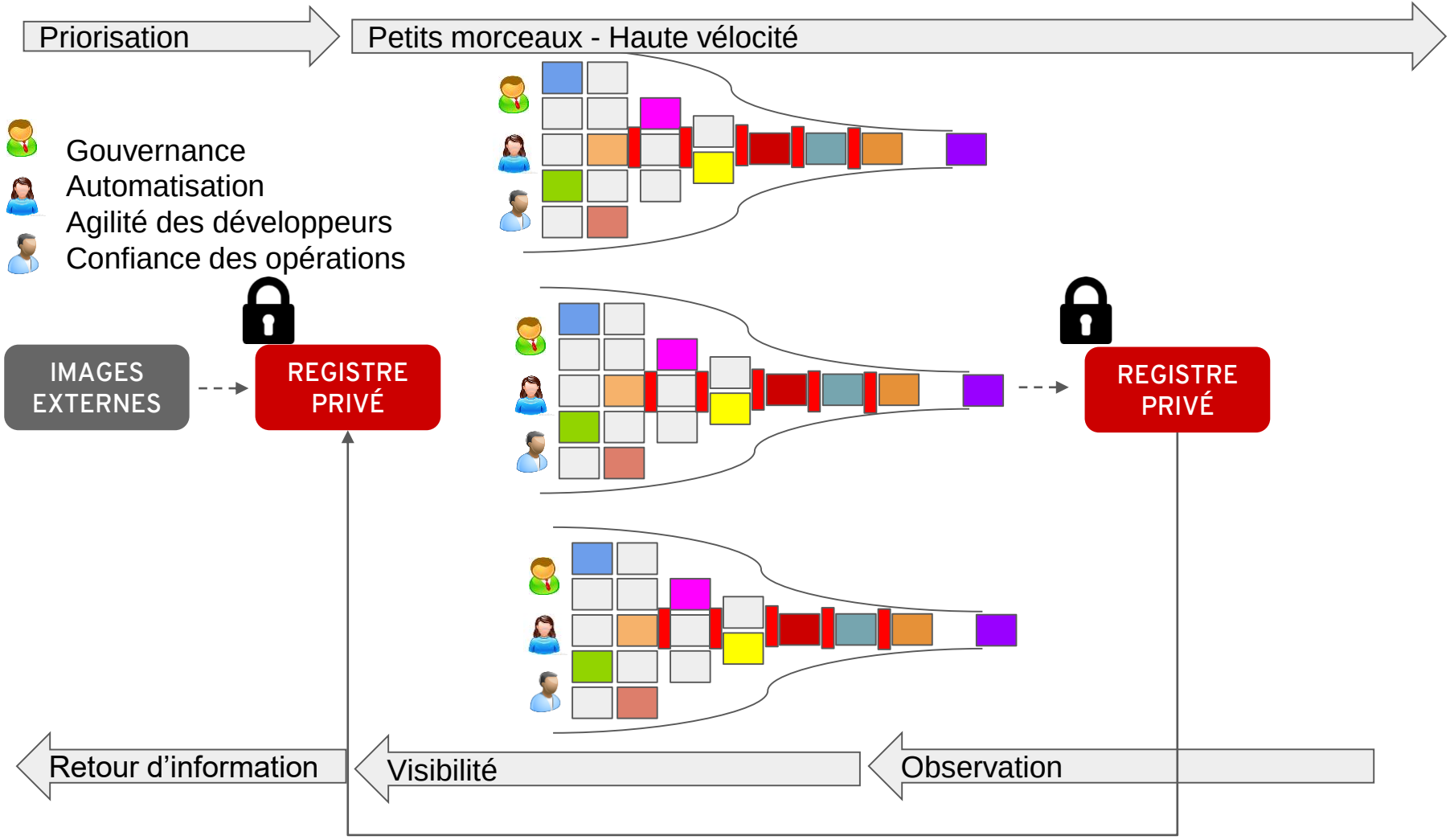
Petits morceaux - Haute vélocité



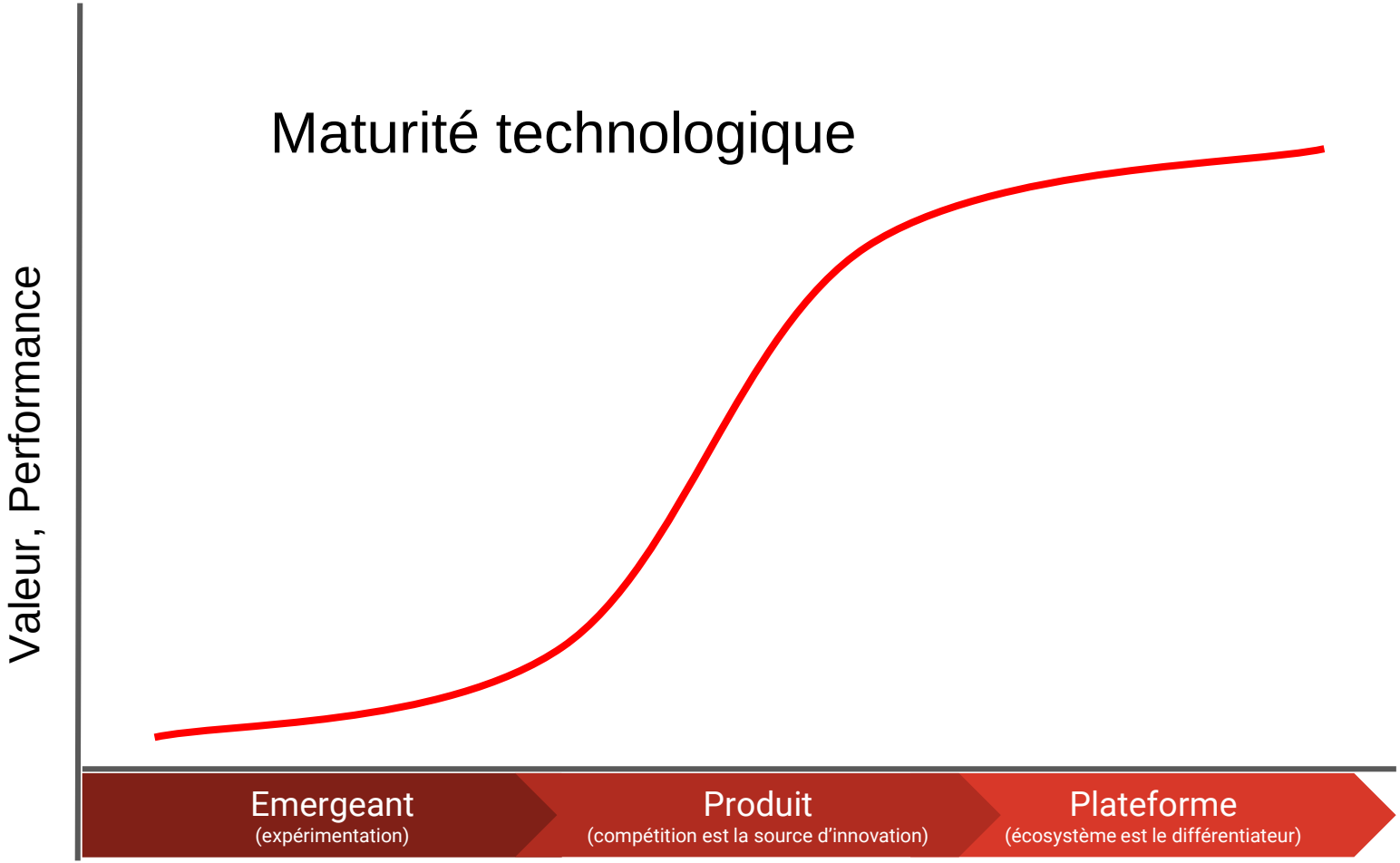
Retour d’information

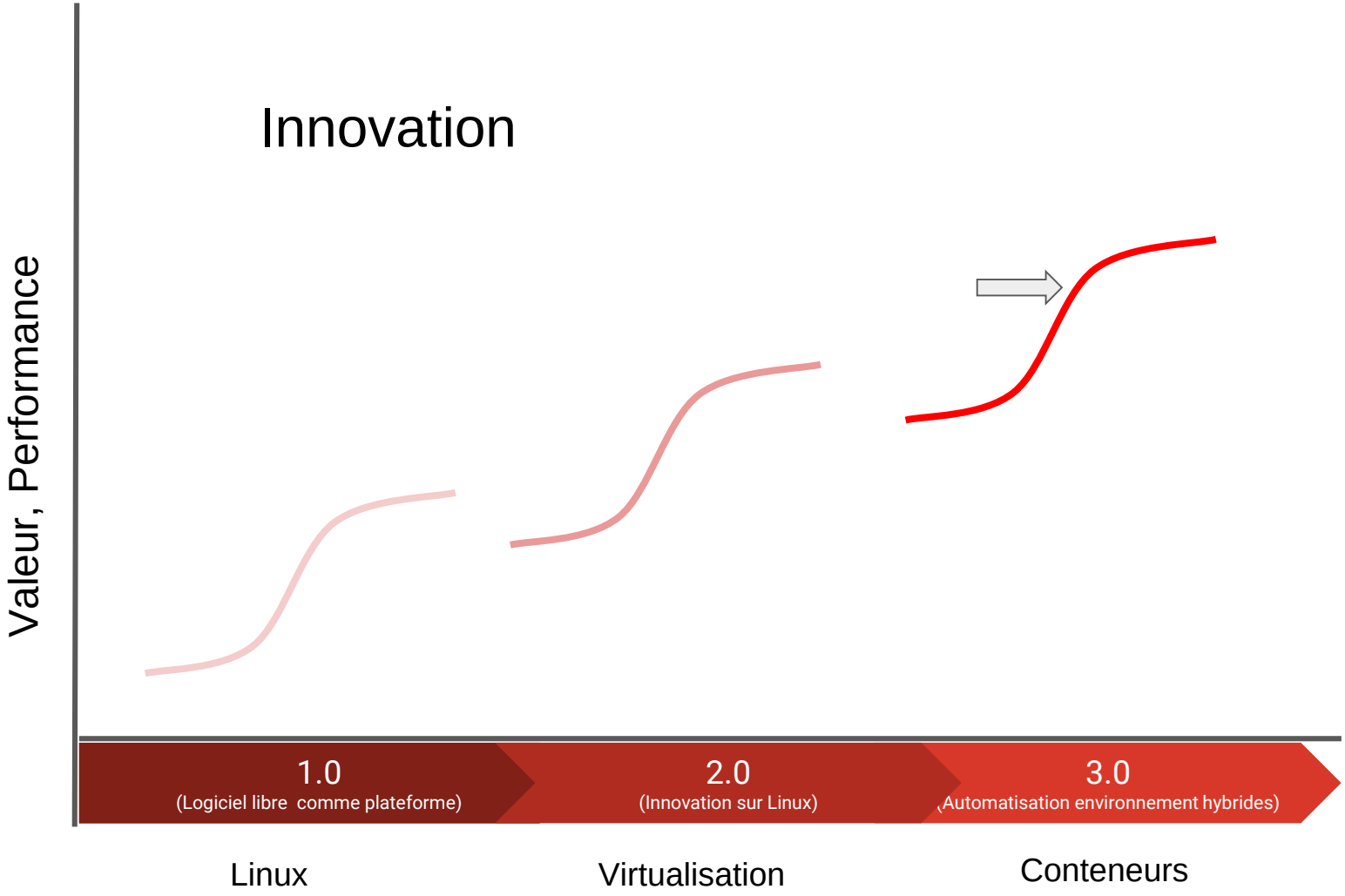
Visibilité

Observation

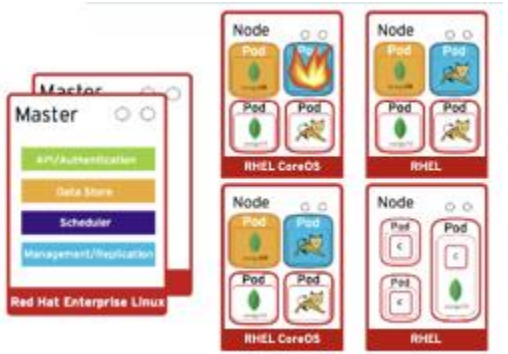

Priorisation
Petits morceaux - Haute vélocité
Gouvernance
Automatisation
Agilité des développeurs
Confiance des opérations
IMAGES EXTERNES
REGISTRE PRIVÉ
REGISTRE PRIVÉ
Retour d'information
Visibilité
Observation

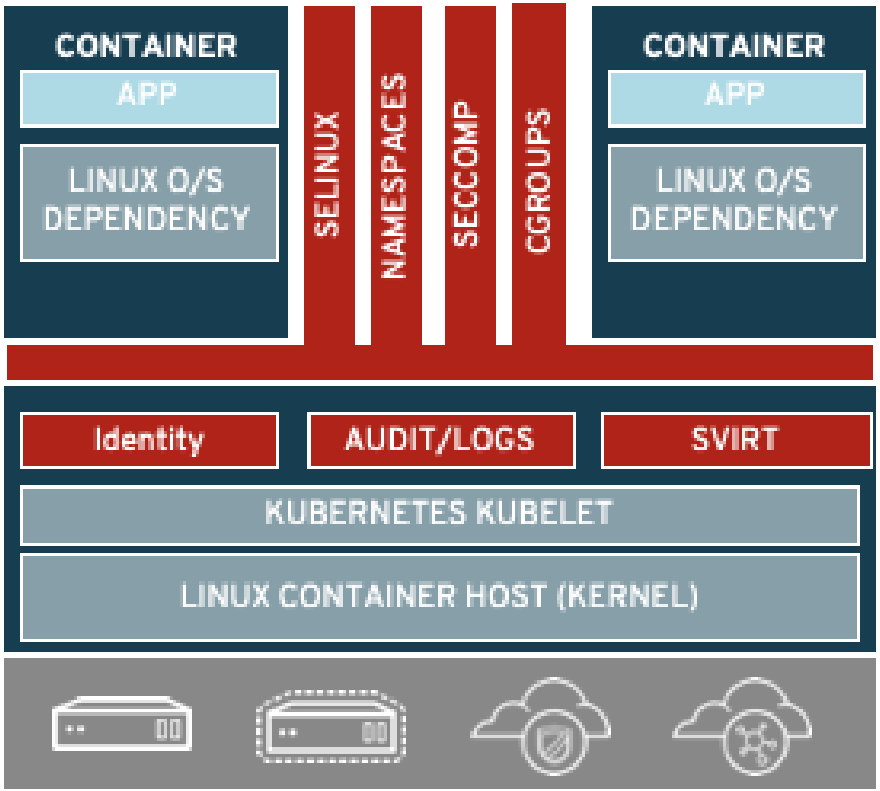
TECHNOLOGIE

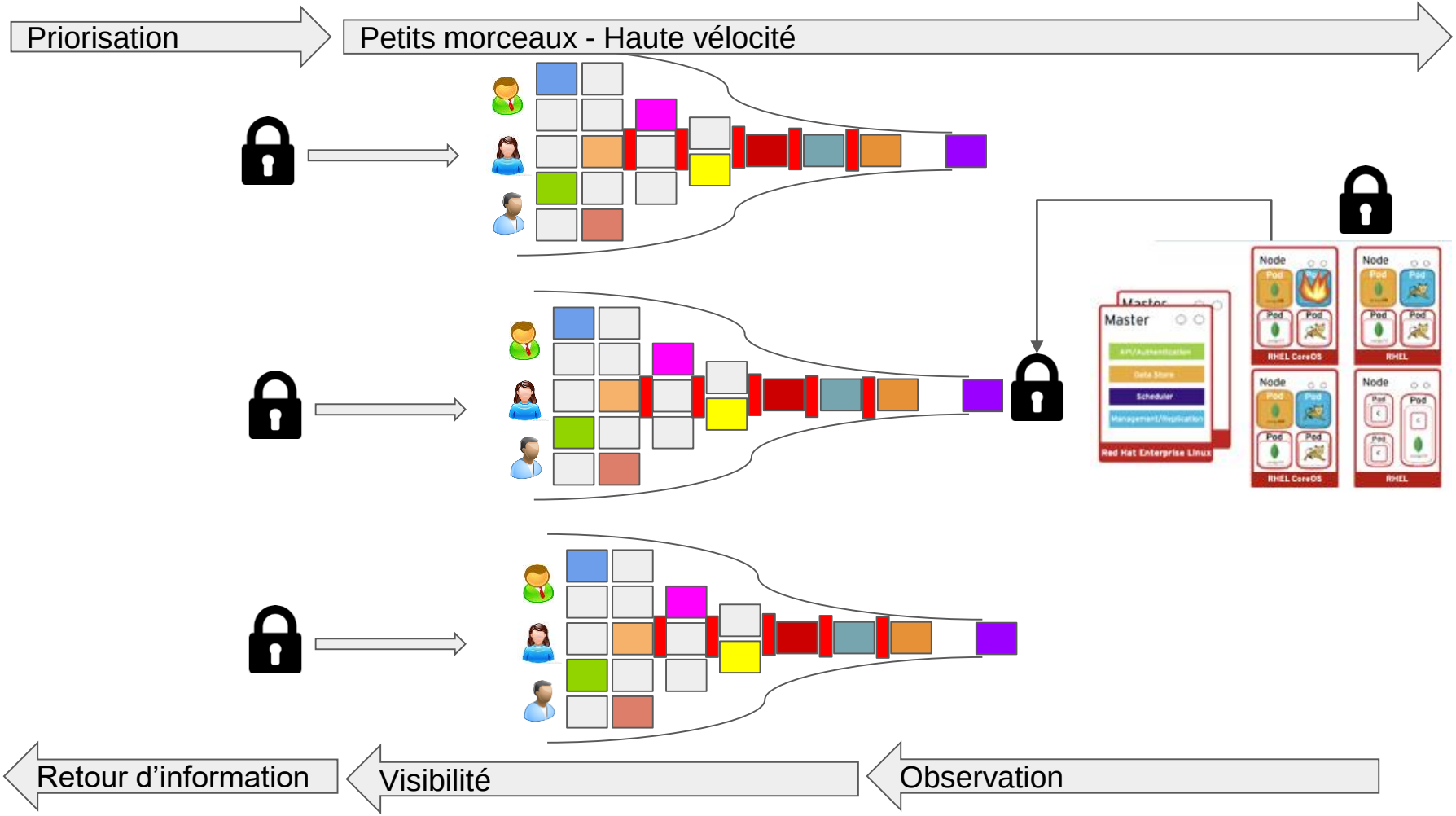

Maturité technologique
Valeur, Performance
Emergeant
(expérimentation)
Produit
(compétition est la source d'innovation)
Plateforme
(écosystème est le différentiateur)


Innovation
Valeur, Performance
1.0
(Logiciel libre comme plateforme)
2.0
(Innovation sur Linux)
3.0
(Automatisation environnement hybrides)
Linux
Virtualisation
Conteneurs

1. Plateforme sécurisée par défaut
    a. Gestion des identitités
    b. Rôles et contrôles des accès
    c. Politiques réseau
    d. Gestion des informations confidentielles
    e. Logs, audit, archivage




CONTAINER
APP
LINUX O/S DEPENDENCY
SELINUX
NAMESPACES
SECCOMP
CGROUPS
CONTAINER
APP
LINUX O/S DEPENDENCY
Identity
AUDIT/LOGS
SVIRT
KUBERNETES KUBELET
LINUX CONTAINER HOST (KERNEL)

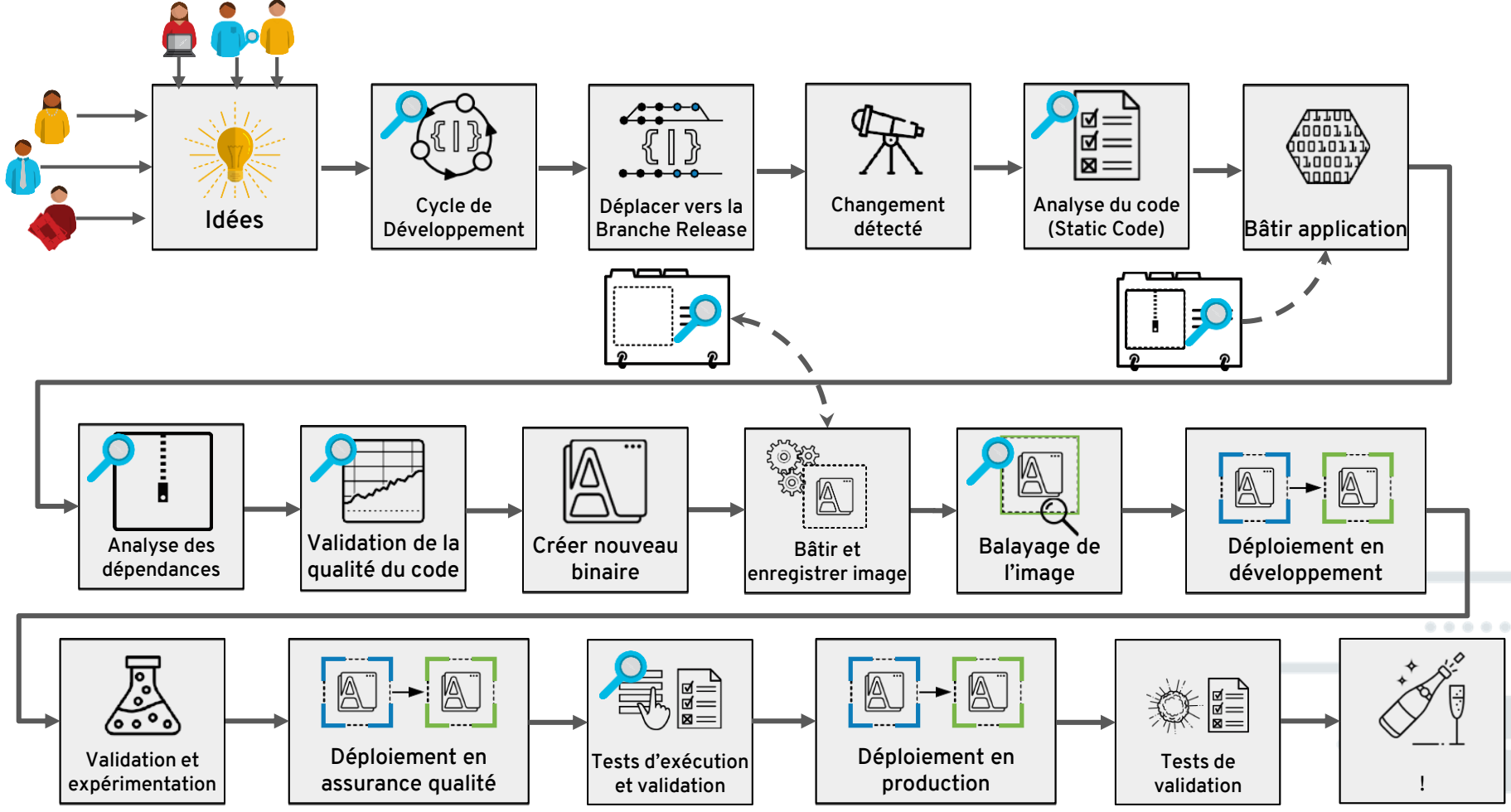

Priorisation
Petits morceaux - Haute vélocité
Master
Node
Pod
RHEL CoreOS
RHEL
Retour d'information
Visibilité
Observation

CULTURE

CULTURE

1. Support exécutif
2. Choix des ressources - enthousiasme, désir de changement, désir de bien faire les choses
3. Bâtir des équipes multi-disciplinaires
4. Considérer les obstacles (humains, technologiques…)
5. Commencer petit et améliorer à chaque cycle
6. Les échecs vont arriver … et le succès
7. OUVERTURE
    a. Ouverture à la rétroaction
    b. Ouverture aux erreurs
    c. Ouverture aux changements

MERCI!


Idées
Cycle de Développement
Déplacer vers la Branche Release
Changement détecté
Analyse du code (Static Code)
Bâtir application
Analyse des dépendances
Validation de la qualité du code
Créer nouveau binaire
Bâtir et enregistrer image
Balayage de l'image
Déploiement en développement
Validation et expérimentation
Déploiement en assurance qualité
Tests d'exécution et validation
Déploiement en production
Tests de validation
!