

# Architecture - Agnostique

Platform Engineer

**Observabilité**

**Reliability**

**Performance**



kubernetes



# From Infrastructure to Applications

## Business Value

Comment déployer des applications de façon fiable, sécurisée et scalable ?

## KPIs & ROI

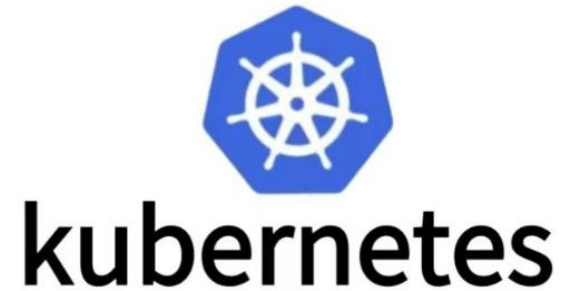
- Deployment
- Error
- Recovery
- Productivity
- Compliance

### BEFORE (Sans GitOps Applications):

- Déploiements manuels et risqués
- Inconsistance dev/prod
- Pas de traçabilité des changements
- Updates security réactives - Scalabilité limitée

### AFTER (Avec GitOps Workflow):

- Zero-touch deployments
- Environment parity garantie
- Audit trail complet dans Git
- Security updates automatiques
- Scaling infini avec ApplicationSets



# Core Applications & Self-Management Pattern

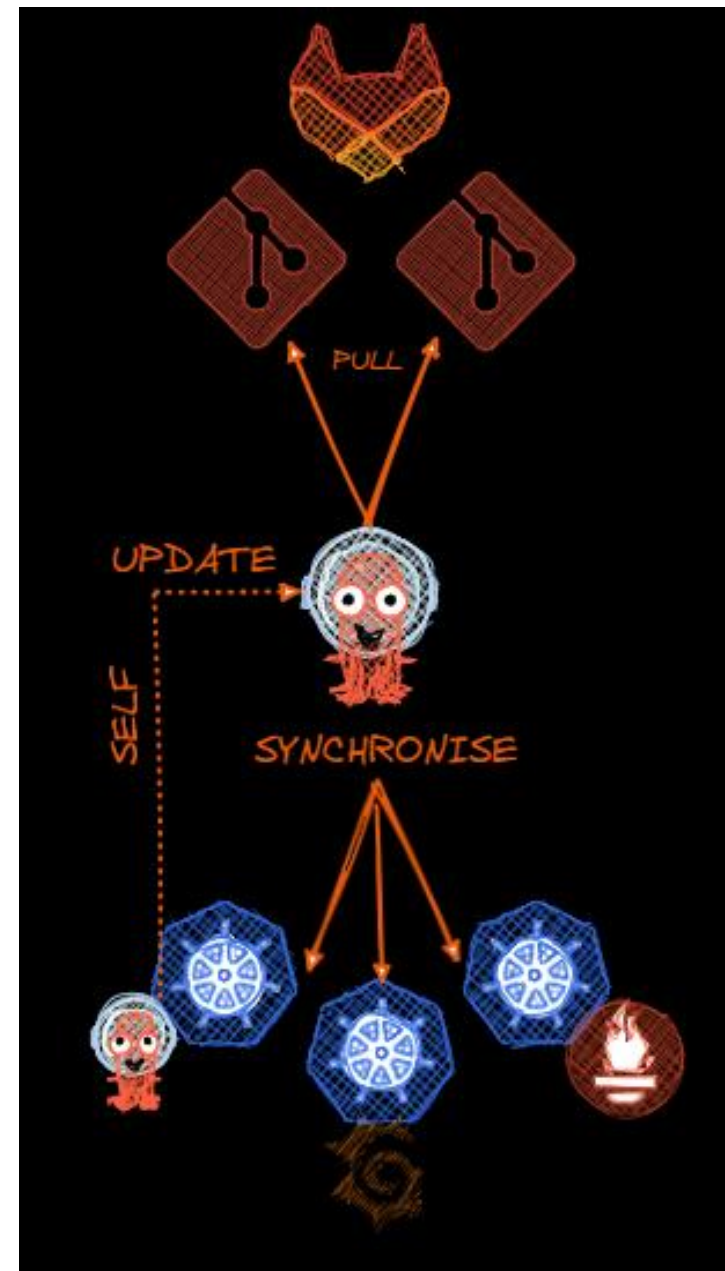
Comment ArgoCD se gère lui-même et bootstrappe les composants critiques du cluster ?

## Bootstrap/core :

- argocd/ (Self-managed GitOps controller)
- cilium/ (CNI + LoadBalancer + Ingress)
- cert-manager/ (TLS automation)
- trust-manager/ (CA distribution)
- reloader/ (Config hot-reload)
- sealed-secrets/ (Encrypted secrets)

## Self-Bootstrap Pattern :

- Déploie sa propre configuration cilium/ (CNI + LoadBalancer + Ingress)
- Bootstrap autres core appstrust-manager/ (CA distribution)
- Manage tout le lifecyclesealed-secrets/ (Encrypted secrets)



# Multi-Environment Platform Architecture

Sécurité, isolation et orchestration

## Platform Environments:

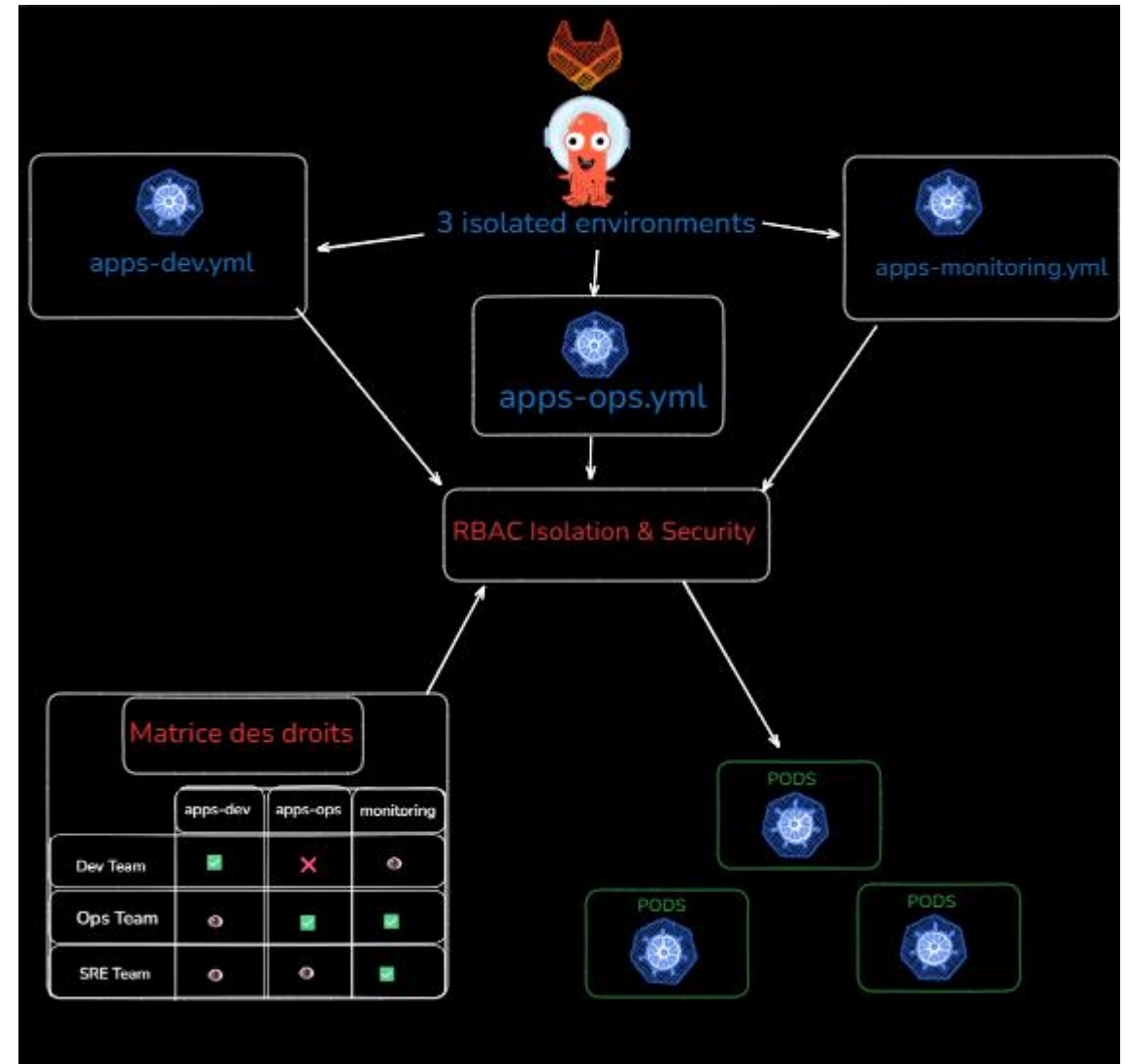
- Dev
- Ops
- Monitoring

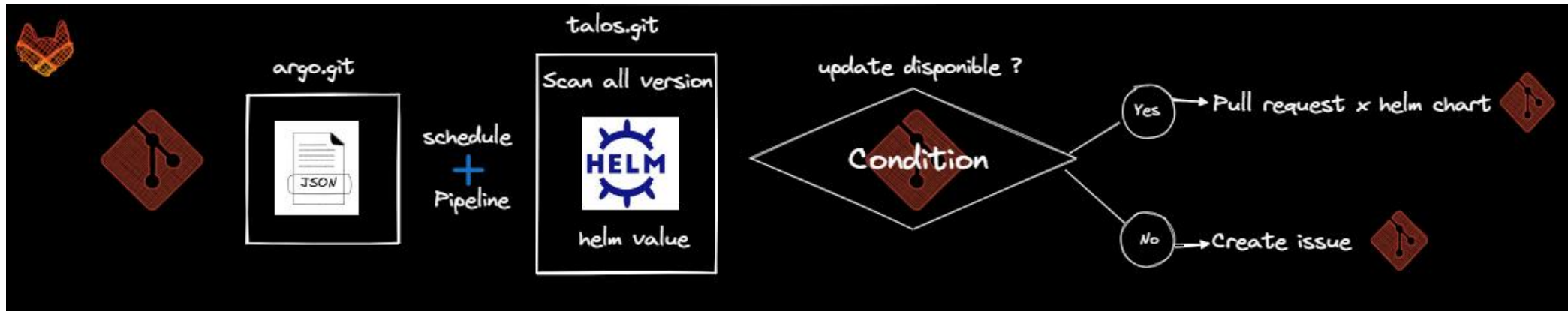
## Platform Governance:

- Project-based isolation (RBAC boundaries)
- Pod Security Standards enforcement
- Controlled resource access patterns
- Safe deletion with PruneLast policies

## Platform Operations:

- Sync wave orchestrationS
- Protected core namespaces (Prune=false)
- Multi-tenant resource management
- Environment consistency guarantees





chore(deps): Update Helm release nginx to v18.3.6 0 of 1 checklist item completed

updated 2 days ago

!5 · created 3 days ago by zack

dependencies

renovate

## Renovate - Mises à Jour automatique des Dépendances

- **Configuration Déclarative:** renovate.json pour définir comportement et règles
- **Regex Managers:** Détection intelligente des versions via patterns regex
- **Helm chart value :** Gestion automatique des versions
- **Semantic Commits:** Préfixe "chore(deps):" pour clarté des messages
- **Limites Concurrentes:** Maximum 2 PR simultanées pour éviter surcharge

# Developer Self-Service

Comment ajouter une nouvelle application en 3 fichiers et 5 minutes ?

1. Créer dossier dans apps/apps-{env}/myapp/
2. Ajouter config.json (5 lignes)
3. Ajouter myapp-values.yaml dans k8s\_talos.git
4. ApplicationSet auto-découvre → App déployée

## Pattern Scaling Proof :

- ingress/ : Multiple ingress centralisés
- gateway-api/ : Nouvelles apps facilement ajoutées
- prometheus-ingress.yaml : Monitoring stack intégré

