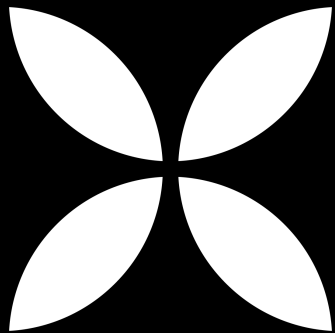




**Rendez vos développeurs autonomes
sur la production**

Mathieu Corbin

Staff Site Reliability Engineer chez Qonto

<https://www.mcorbin.fr/>

@_mcorbin



Développement



Automatisation



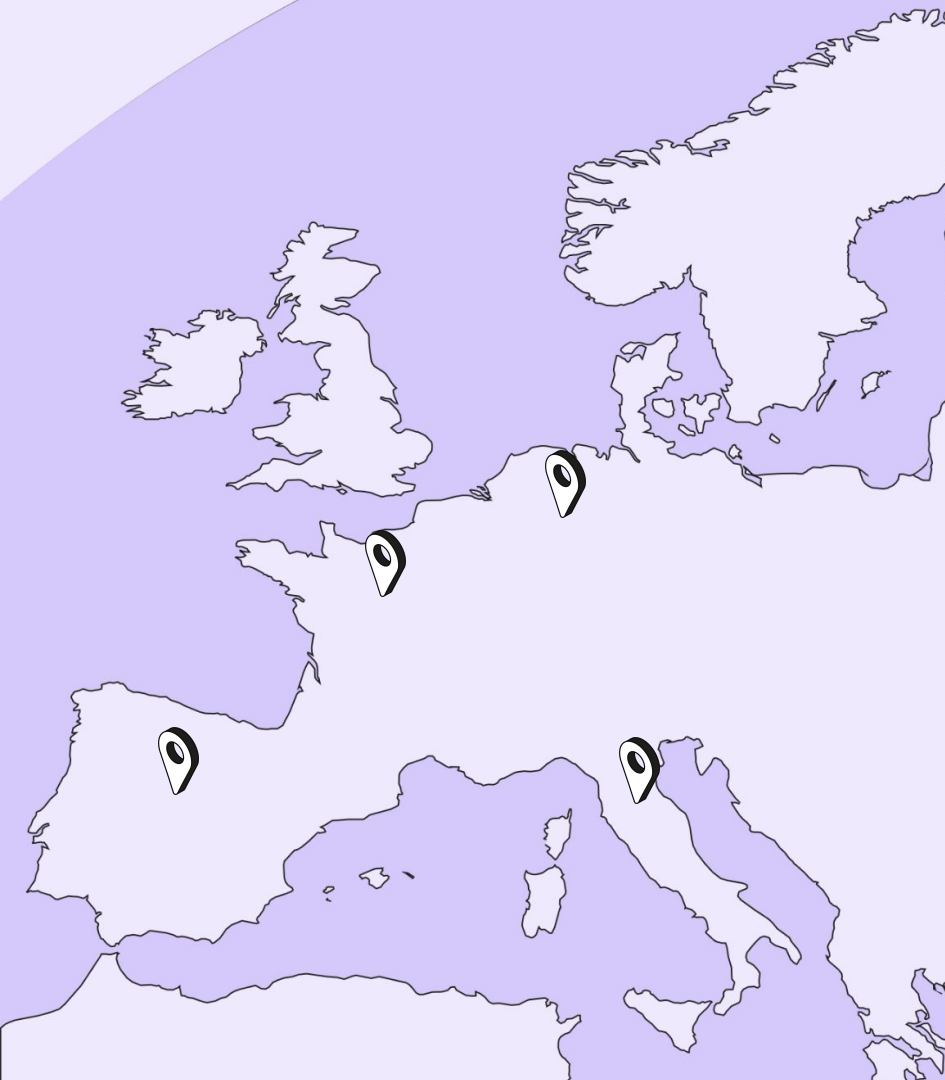
Observabilité



Architecture & systèmes
distribués



CI/CD



Qonto

**Le compte qui booste votre
business**



DevOps

Manager



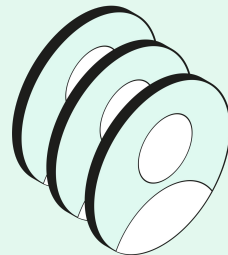
Ops/SRE/...



Dev



Une communication difficile



Ops : c'est MA prod

Personne ne peut faire une action dessus sans mon autorisation explicite

Ops : Les dev ont ENCORE poussé un bug en prod !

On a encore reçu des alertes !

Dev : c'est merge ? Ce n'est plus mon problème

Ce qu'il se passe après ne m'intéresse pas

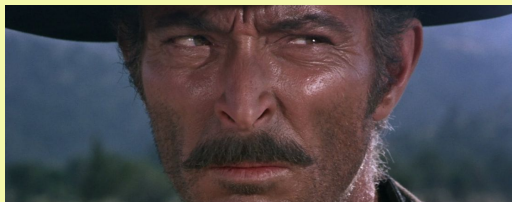
Dev : les ops ont ENCORE dit non pour l'ajout d'une fonctionnalité dont on a besoin !

Tant pis, on va l'installer en douce et ne pas les prévenir

Qui gère quoi au quotidien ?

Tout le monde doit travailler sur la fiabilité de la plateforme

Responsabilités partagées



La production au sens large

Applications, monitoring, cloud...

Ops



CI/CD

Ecriture de pipelines de déploiements

DevOps

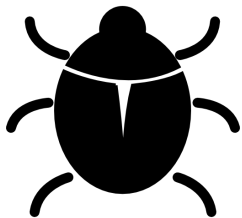


Développement

Ecrire les applications

Dev

Problème: allers-retours inutiles entre équipe



Bug en prod



On a détecté ce bug,
corrige le stp



C'est fait. J'ai un nouveau log
d'erreur, c'est normal ?



C'est corrigé, met la
release 1.4.2 en prod



Attends je regarde



Ok ping moi quand tu sais

You build it, you run it



Développement

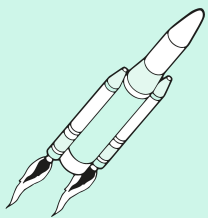


CI/CD

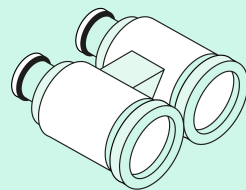


Production

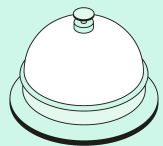
L'objectif est l'autonomie des équipes de développement



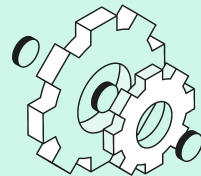
Mise en production & rollbacks



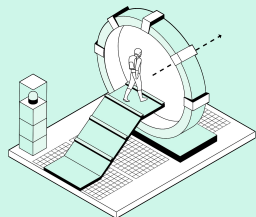
Accès aux logs et métriques



Configuration des alertes



Création de nouveaux projets



Investigation des problèmes



Performance

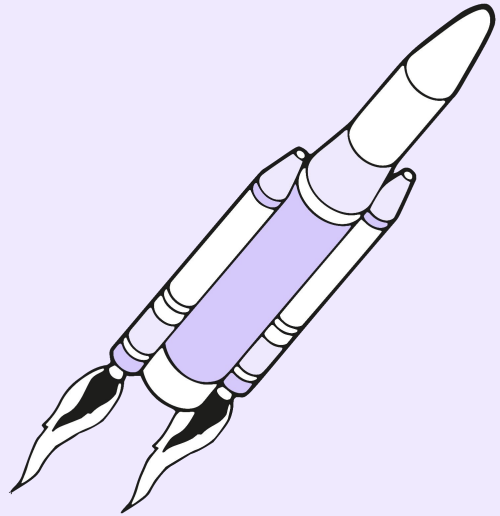
Que font donc les équipes ops ?

Mise en place de l'infrastructure

Abstraction de l'infrastructure (plateforme)

Accompagnement et standards

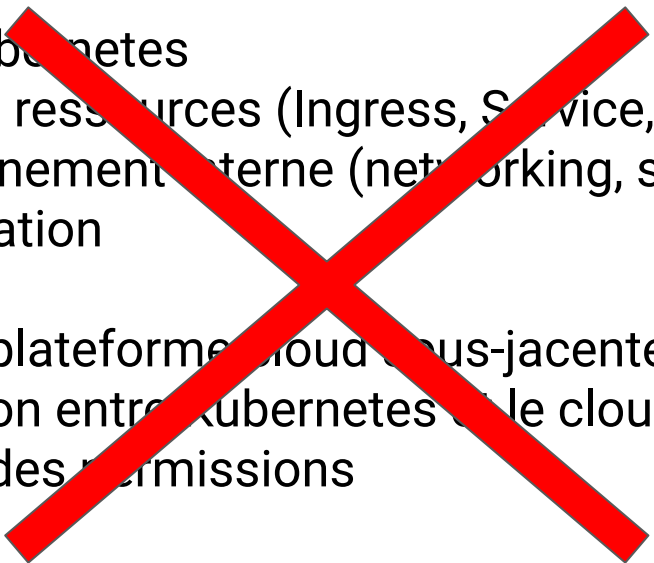
Mise en production



Les développeurs doivent

- Connaître Kubernetes
 - Types de ressources (Ingress, Service, Deployments...)
 - Fonctionnement interne (networking, scheduling...)
 - Configuration
- Connaître la plateforme cloud sous-jacente
 - Intégration entre Kubernetes et le cloud
 - Gestion des permissions

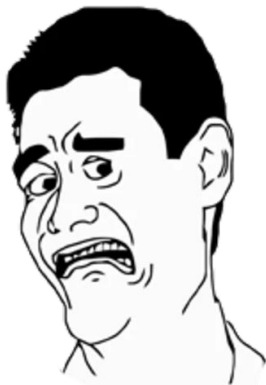
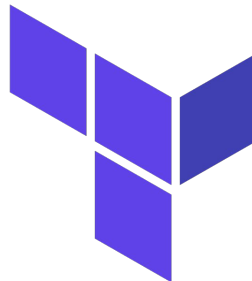
Les développeurs doivent

- Connaître Kubernetes
 - Types de ressources (Ingress, Service, Deployments...)
 - Fonctionnement interne (networking, scheduling...)
 - Configuration
 - Connaître la plateforme cloud sous-jacente
 - Intégration entre Kubernetes et le cloud
 - Gestion des permissions
- 

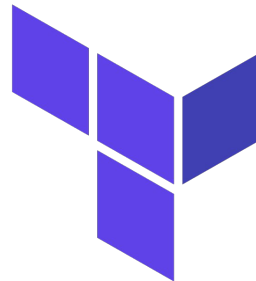
Erreur commune: leak de l'infra côté dev



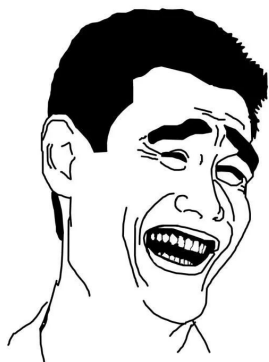
Dev



Une interface claire entre l'infra et les dev



APIs, self service



Travailler ensemble

Dev



Cahier des charges:

- Déployer à tout moment
- Autoscaling
- Service discovery
- Rollbacks
- Libre sur la configuration de l'application

Travailler vite et bien, en autonomie

Ops

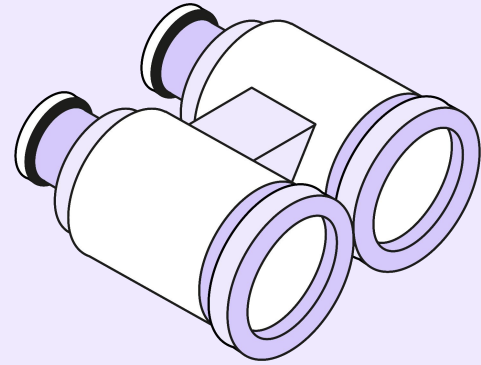


Création de l'abstraction:

- Services Cloud
- Kubernetes
- API, tooling
- CI/CD

Répondre aux besoins de manière fiable et avec une bonne UX

Exploitation en production



Dev

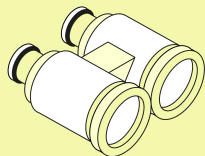


“Depuis la release d’hier mon application est lente, tu sais pourquoi ?”

Ops



Qui a codé l’application ?



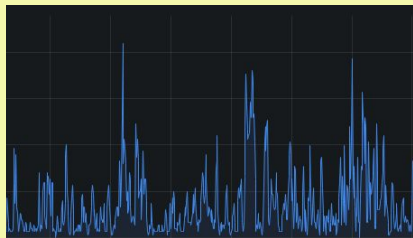
Qu'est ce que je dois monitorer ?

Standards sur le nommage, labels, métriques
(requêtes/sec, erreurs, latence, saturation, SLO...)



Comment j'expose mes métriques ?

*"Expose les au format Prometheus sur **/metrics** et elles seront automatiquement récupérées"*



Comment j'y accède ?

Création de dashboards communs, self service

L'importance du self service



Slow query



On a des lenteurs sur
des requêtes SQL



Attends je t'envoie ça
psql ...



Tu as la liste ?



Ok, il y a un bug sur
cette requête, je
corrige



ok, et celle-là c'est normal ?

L'importance du self service



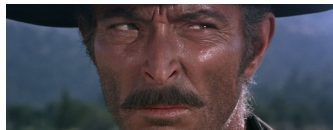
Tu peux lancer cette action en prod sur ce service ?



C'est fait, voilà la sortie



Ok, tu peux lancer ça Maintenant ?

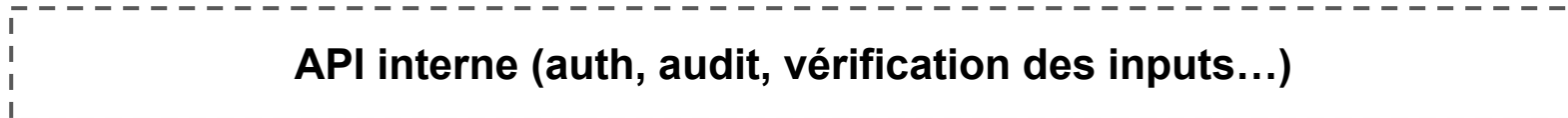
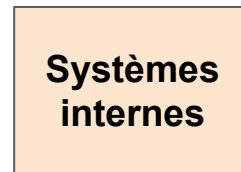
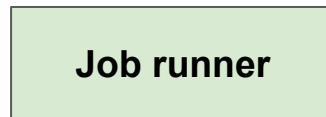
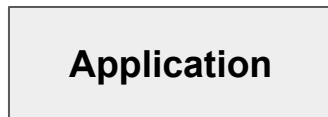
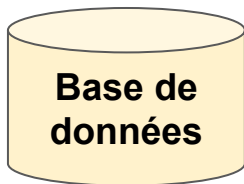


C'est fait



Tu es **sûr** que tu l'as bien fait ?

L'importance du self service: API first



- CLI
- User interfaces
- CI/CD
- ...

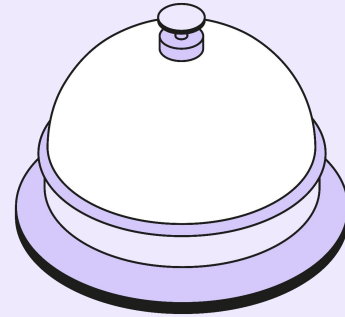
Rendre les équipes autonomes demande de l'outillage

- Quelles sont les actions où l'ops n'a aucune valeur ajoutée (passe plat) ?
- Devons nous *“aller sur la prod”* (**ssh, kubectl exec, psql...**) pour aller chercher des informations ou déclencher des actions ?
 - Toujours risqué et source d'erreurs
- Est-il facile de travailler au quotidien ? Est ce que j'ai des actions pénibles à réaliser régulièrement ?

Créer l'écosystème interne de l'entreprise

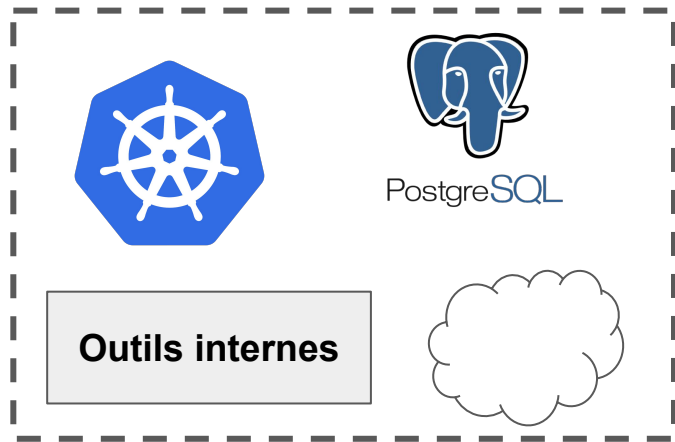
- Fournir de nouveaux services internes
- Réutilisation de code entre outils et services (clients, libraries...)
- Outillage standard et ouvert à contribution
 - CLI
 - API
 - Tooling
- **Accélération du travail au quotidien**

Incidents et astreintes



Traditionnellement, les ops reçoivent toutes les alertes...

Catégoriser les alertes



Ops



Dev



Astreintes 24/7



Nuit et week-end: que les ops alertés



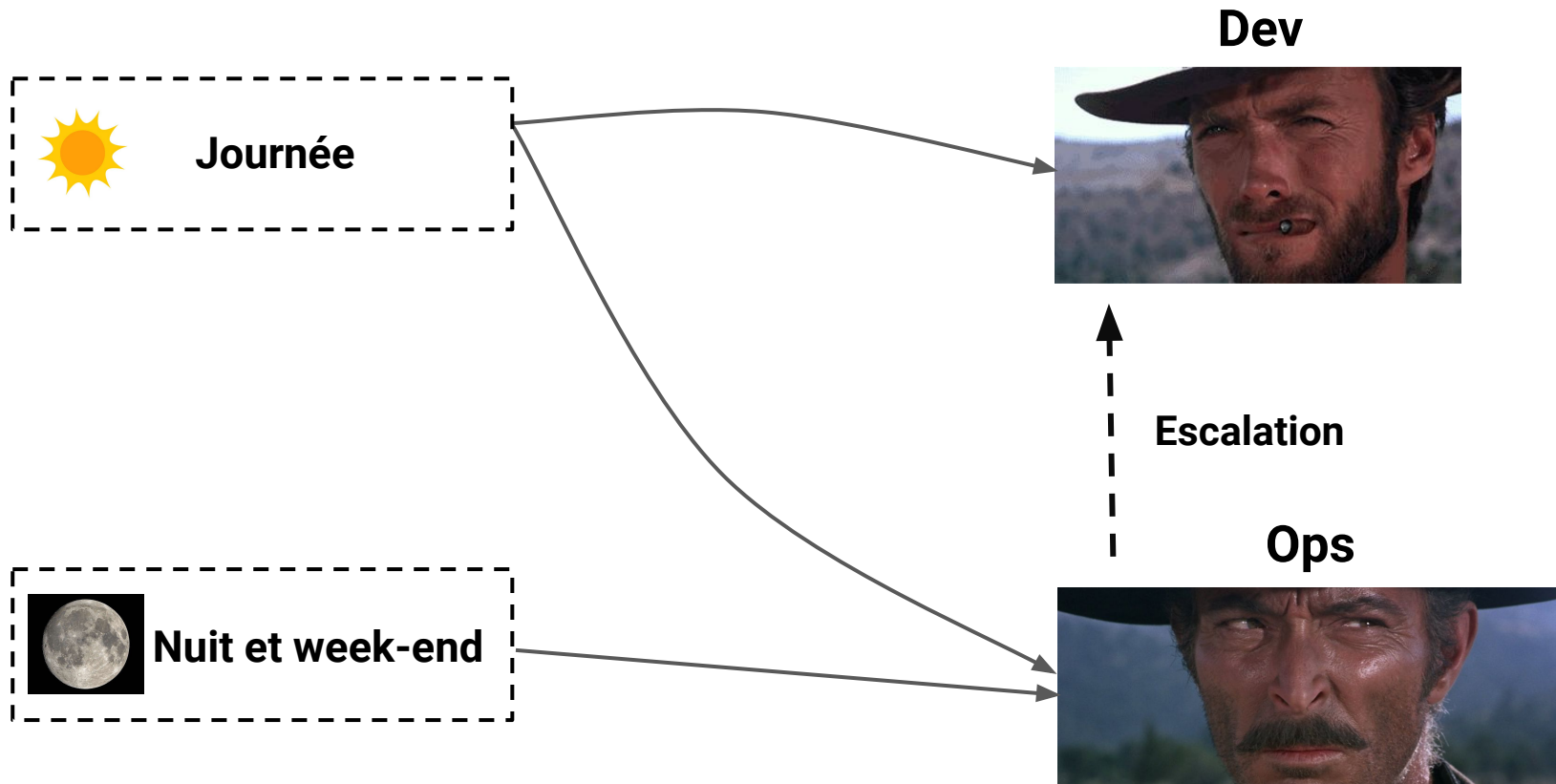
Dev



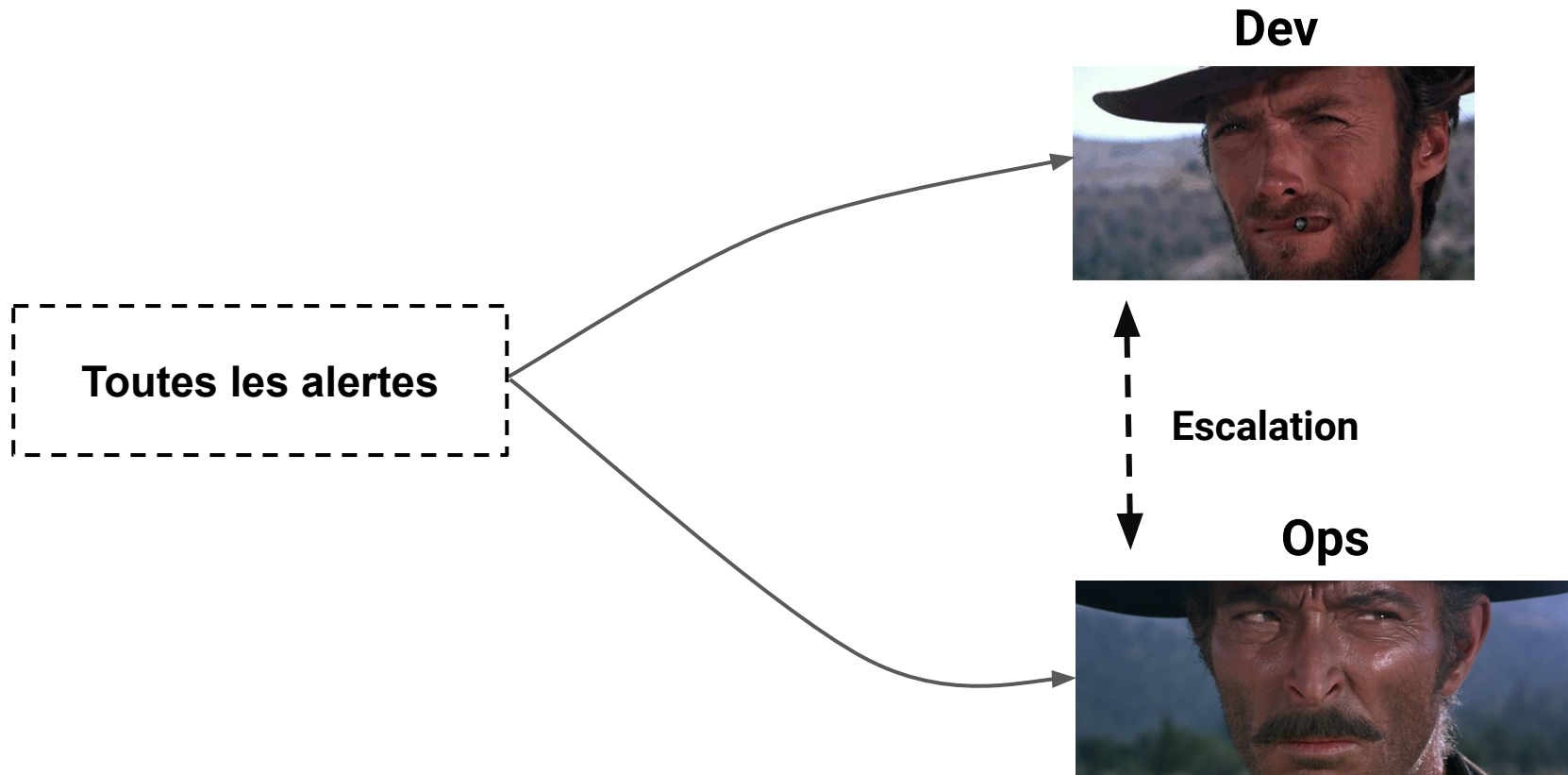
Ops



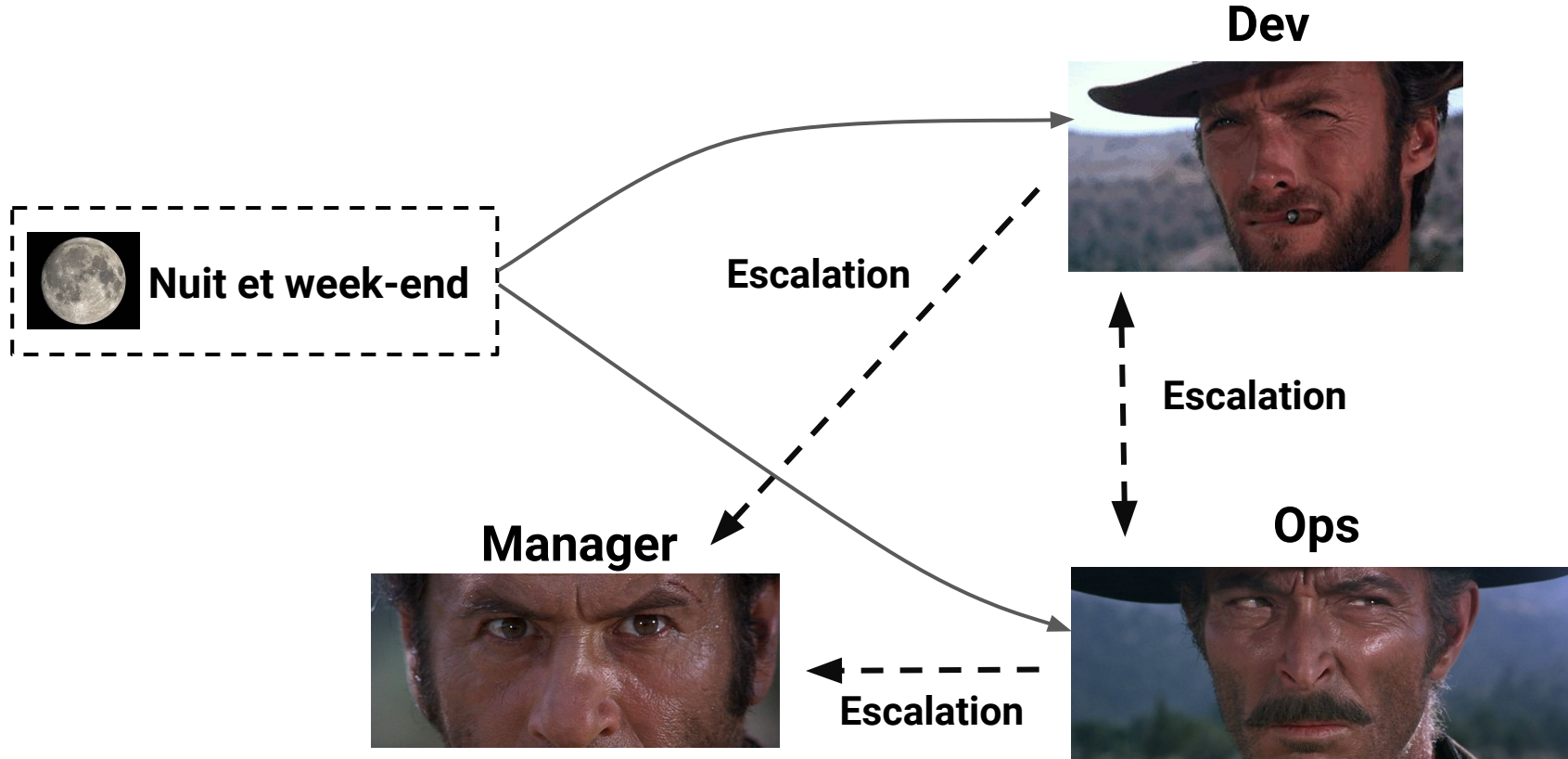
Développeur disponible en escalation



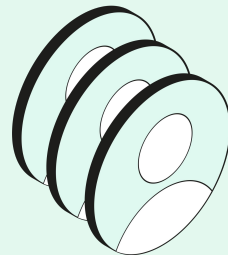
Développeurs et ops en astreinte direct



Astreintes chez Qonto



Culture de la gestion d'incidents



Blameless

On est une équipe, dev comme ops

Procédures claires

Moyens de communication, escalation, incident commander, documentation

Postmortems

Apprendre de nos problèmes et faire en sorte qu'ils ne reviennent plus

Alert fatigue

Les alertes doivent être actionnables, avec des runbooks et de l'outillage

Recevoir des alertes est souvent intimidant

- *Je ne saurai pas répondre à l'incident*
 - Il est impossible de tout connaître
 - Runbooks, formations, shadowing, escalation
- *Je ne suis pas un ops*
 - Ce n'est pas ce qui est demandé
 - Responsabilisation des équipes
- *Je n'ai pas les bon outils pour gérer les incidents*
 - **Souvent un problème**

L'aspect légal de l'astreinte

- Avenant au contrat de travail
 - Ne peut pas être imposée
- Payée
 - Généralement forfait à la semaine + prime horaire lors d'interventions
- Repos obligatoire de 11 heures après intervention

Aller au-delà du code dans les équipes de développement

Cycle de vie des applications

Déploiements, rollbacks

Monitoring

S'assurer de la santé de l'application

Gestion des incidents

Debugging, correction des problèmes

Responsabilité

C'est eux qui connaissent le mieux les applications

Merci !



Notre blog

<https://medium.com/qonto-way/tagged/tech>



Nos postes ouverts

<https://qonto.com/en/careers>



Questions ?

mathieu.corbin@qonto.com