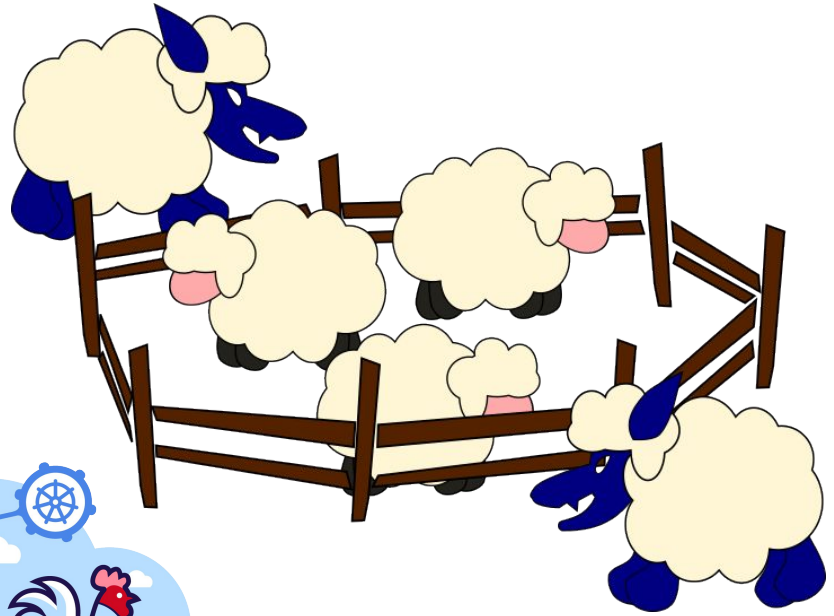
CENTQUATRE-PARIS
5 rue Curial, 75019

Mardi 3 Février
2026

CLOUD NATIVE DAYS
FRANCE 2026

Attestation à distance d’un cluster Kubernetes ou comment protéger la bergerie des loups numériques

Speakers :
Alice Frosi, alicefr
Timothée Ravier, travier



Agenda

- Cluster confidentiel
- Attestation à distance
- Fonctions de l'opérateur Kubernetes

Informatique Confidentielle (Confidential Computing)

Informatique Confidentielle ?

- Chiffrement des données **en mémoire vive** et pas uniquement lors du stockage ou des transferts réseaux
  - Exemples : AMD SEV/SNP, Intel TDX
- Protection du système **vis à vis de l'opérateur cloud** ou des **autres machines virtuelles**
  - Attention : Pas de protection vis à vis d'un administrateur ou `root` malveillant
- Chez un **fournisseur de cloud** ou en **Bare Metal via KubeVirt**
  - OVHcloud : Confidential Computing sur nos serveurs Bare Metal

Objectifs : Cluster Confidentiel

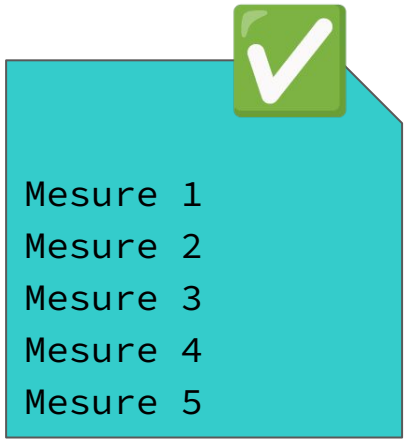
- Obtenir un **cluster Kubernetes**
- Avec **toutes les machines** fonctionnant en **mode confidentiel** (Confidential Computing)
- **Vérifier** que c'est **vrai** (attestation à distance)
- **Automatiser** tout le processus avec un **opérateur** Kubernetes

Attestation à distance

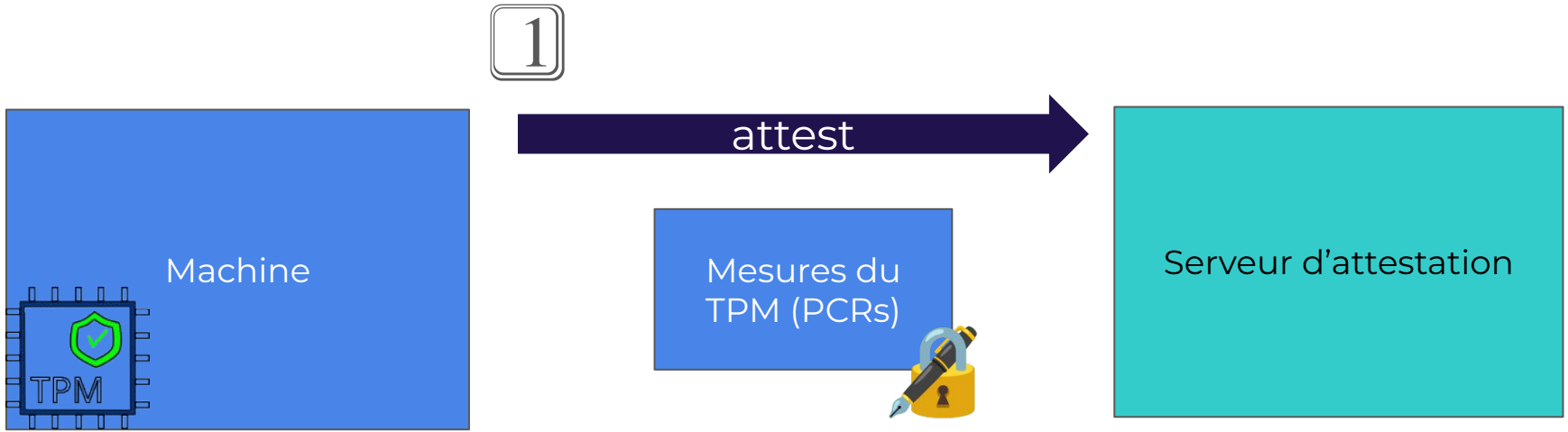
(Remote Attestation)

Attestation distante ?

- Mesures d'intégrité à l'aide du **TPM** (Trusted Platform Module)
- Serveur d'attestation (**Trustee**)
  - https://github.com/confidential-containers/trustee
- **Valeurs de référence** et politique de sécurité
- Récupération d'un **secret**

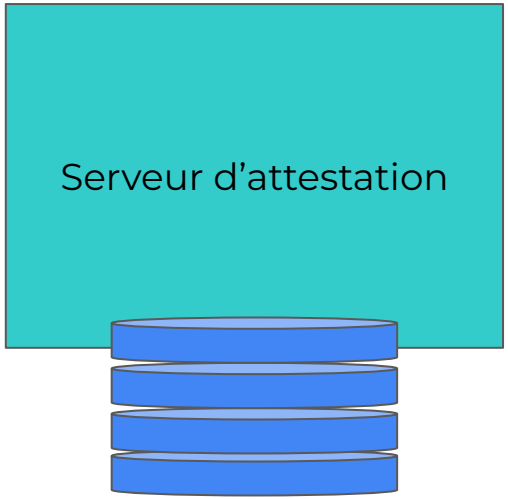


Attestation des machines confidentielles

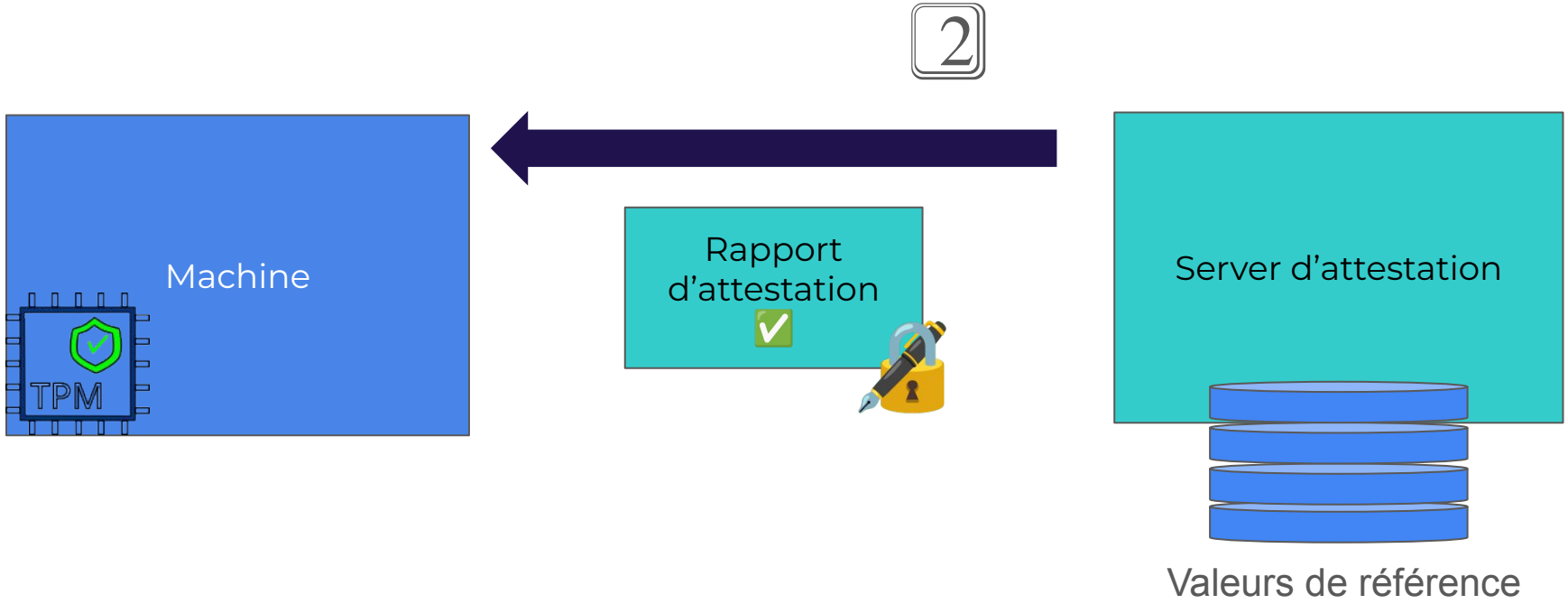


Politique de sécurité d'attestation

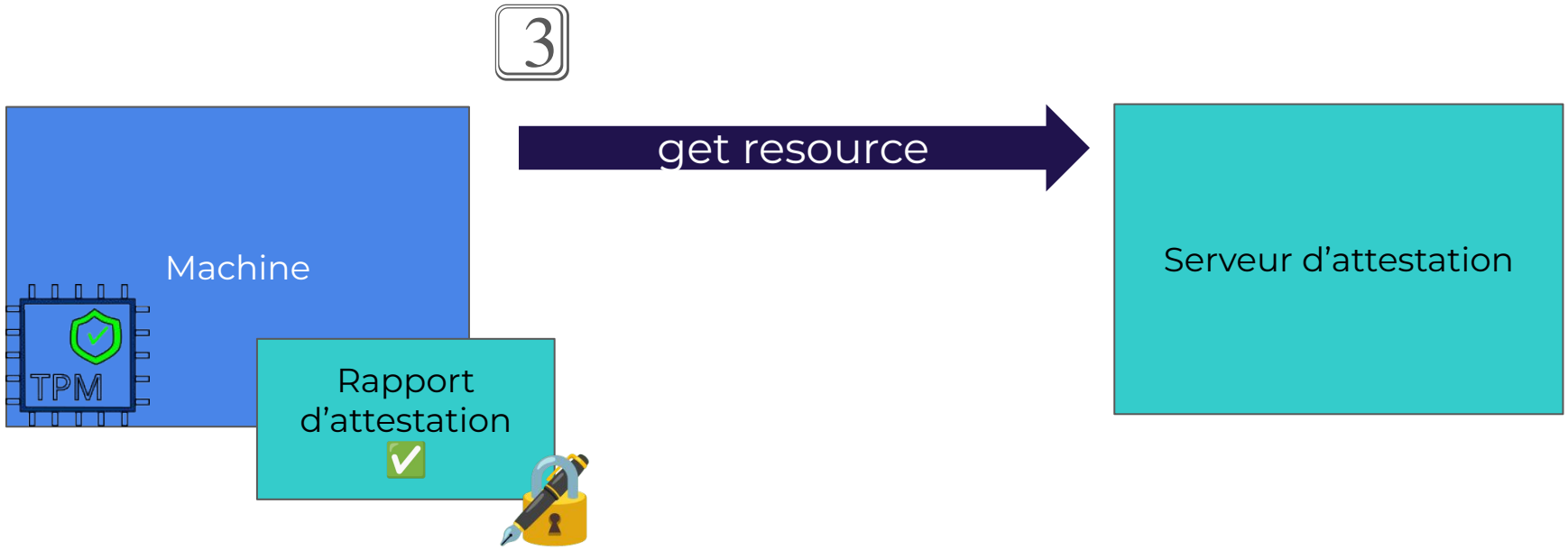
```
package policy
import rego.v1
default executables := 33

tpm_pcrs_valid if {
  input.tpm.pcr04 in query_reference_value("tpm_pcr4")
  input.tpm.pcr07 in query_reference_value("tpm_pcr7")
  input.tpm.pcr14 in query_reference_value("tpm_pcr14")

  pcr8 = expected_pcrs8[UUID]
  pcr8 == input.tpm.pcrs08
}
executables := 3 if tpm_pcrs_valid

default configuration := 0
default hardware := 0
```



Attestation des machines confidentielles



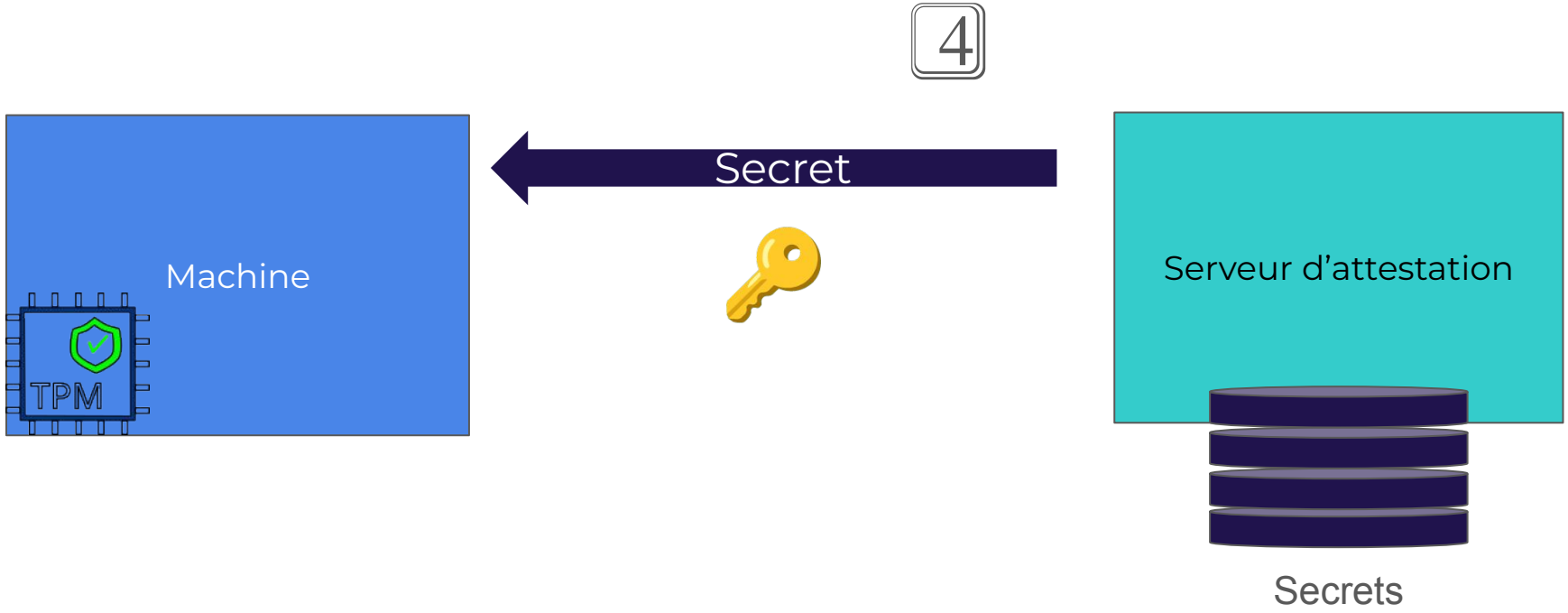
Attestation des machines confidentielles



Politique de sécurité des ressources

```
package policy
import rego.v1

default allow := false

allow if {
  input["submods"]["cpu0"]["ear.status"] == "affirming"

  uuid := split(input.resource_path, "/")[1]
  input.initdata.uuid = uuid
}
```

Attestation des machines confidentielles



Intégration pour un noeud du cluster

Intégration pour chiffrer les noeuds du cluster

- Chiffrement des disques avec **LUKS**
- Réalisé lors du **premier démarrage**
- Le client Trustee réalisé l'attestation à distance
  - puis récupère le **secret LUKS**
- Chiffrement/déchiffrement intégré avec un **plugin Clevis** (Pin)
- Coordonné par **Ignition**
  - configure la machine lors du premier démarrage

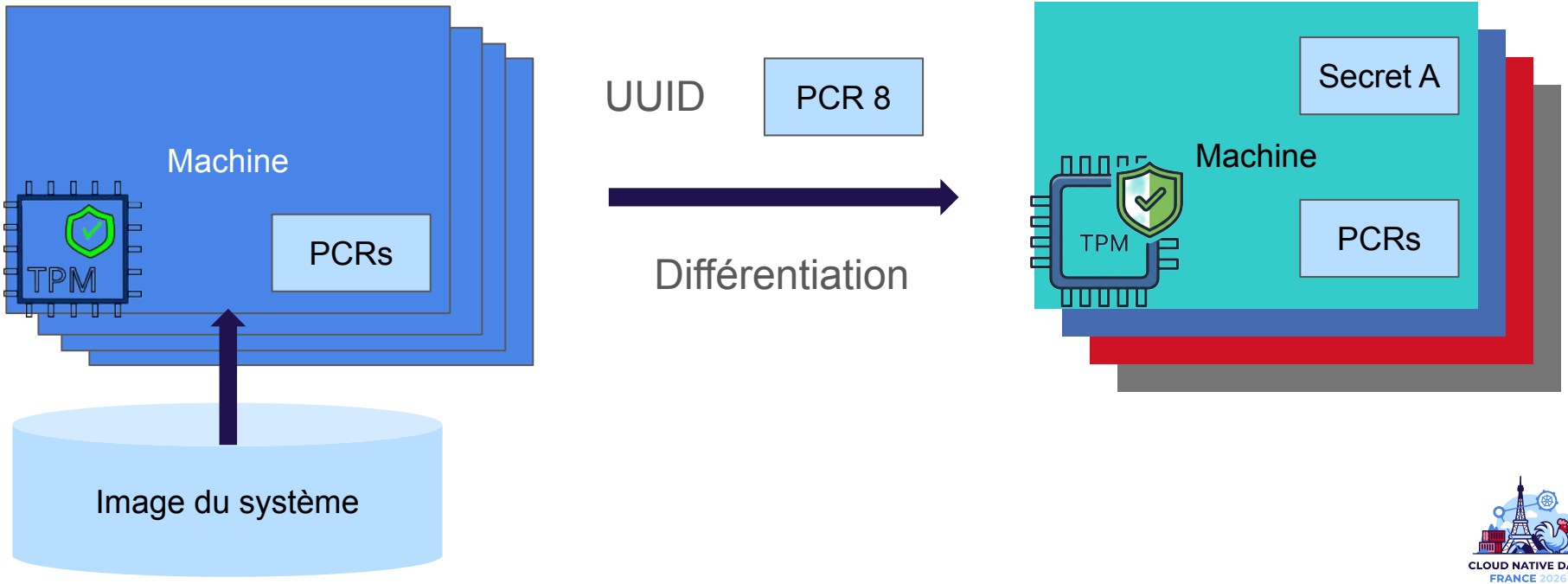
Automatisation avec un opérateur

Combiner le tout et l'automatiser pour un cluster

- **Opérateur** pour Kubernetes
- Déploiement du serveur d’attestation à distance (**Trustee**)
- Gestion des **politiques de sécurité**
- Automatisation du calcul des **valeurs de référence**
- **Approbation** des images du système pour le cluster

Trusted Compute Base (TCB) et secret

- Même image pour chaque noeud (TCB) -> même registres PCR
- Différencier chaque machine pour y assigner un secret unique



Politique de sécurité d'attestation

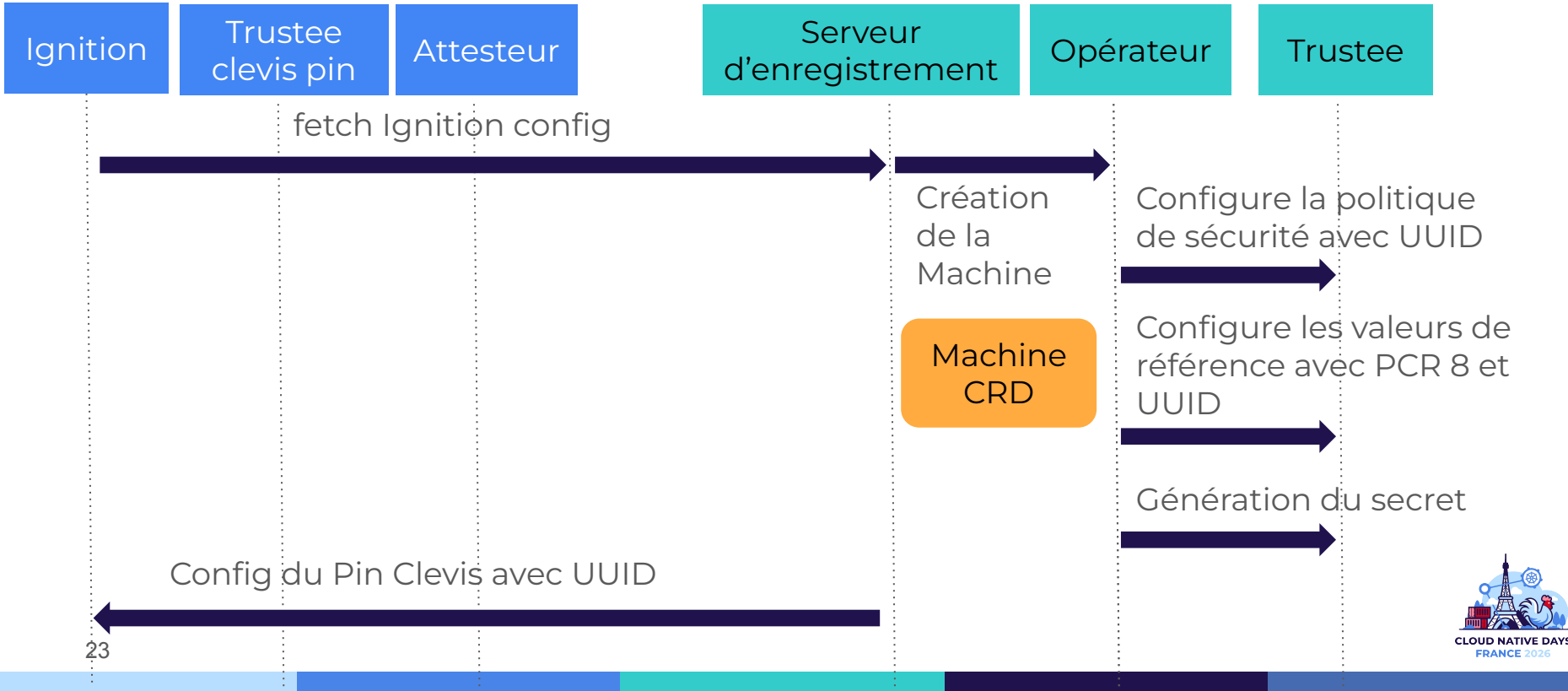
```
package policy
import rego.v1
default executables := 33

tpm_pcrs_valid if {
  input.tpm.pcr04 in query_reference_value("tpm_pcr4")
  input.tpm.pcr07 in query_reference_value("tpm_pcr7")
  input.tpm.pcr14 in query_reference_value("tpm_pcr14")

  pcr8 = expected_pcrs8[UUID]
  pcr8 == input.tpm.pcrs08
}
executables := 3 if tpm_pcrs_valid

default configuration := 0
default hardware := 0
```

Politique de sécurité des ressources

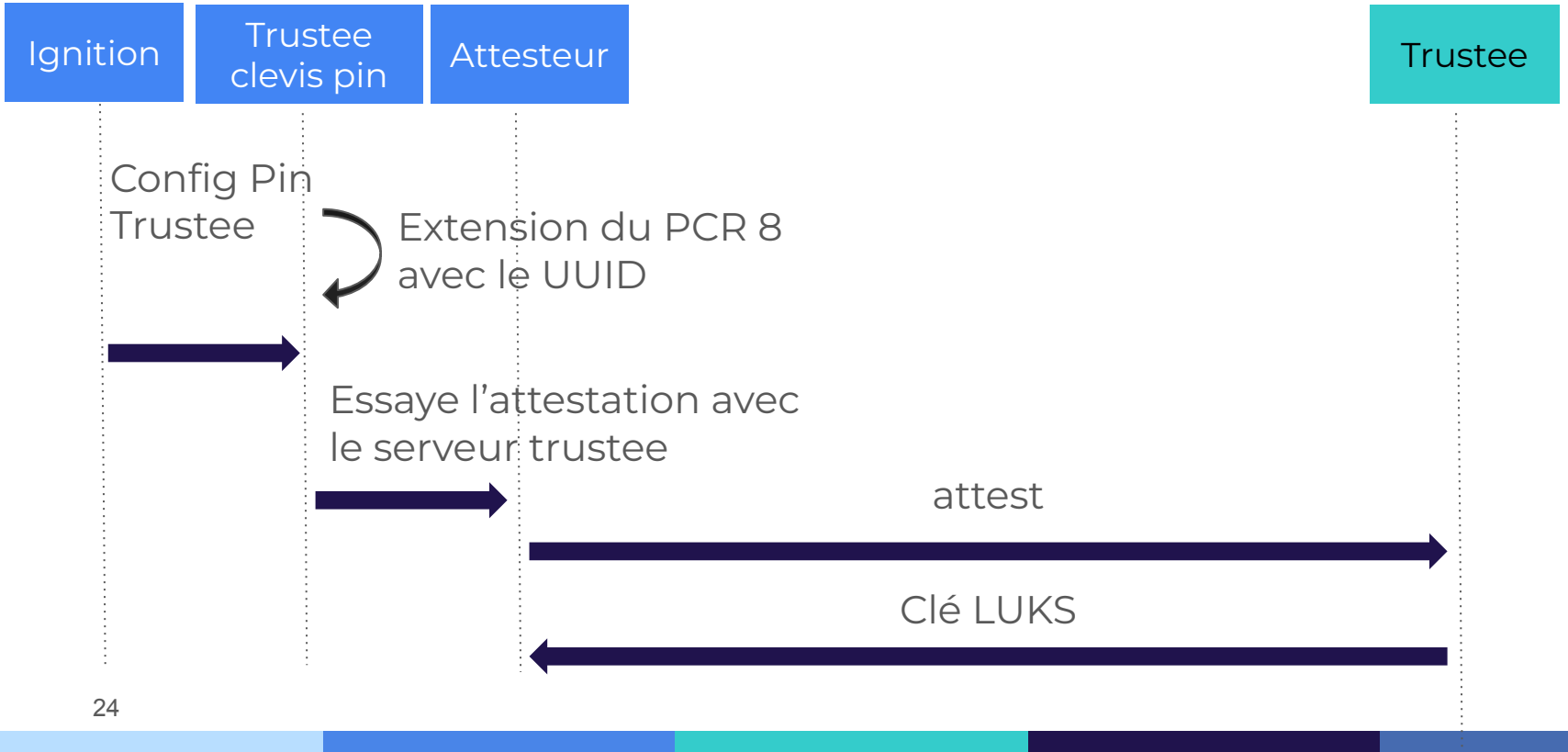
```
package policy
import rego.v1

default allow := false

allow if {
  input["submods"]["cpu0"]["ear.status"] == "affirming"

  uuid := split(input.resource_path, "/")[1]
  input.initdata.uuid = uuid
}
```

Déroulement complet

Premier démarrage d'une machine confidentielle



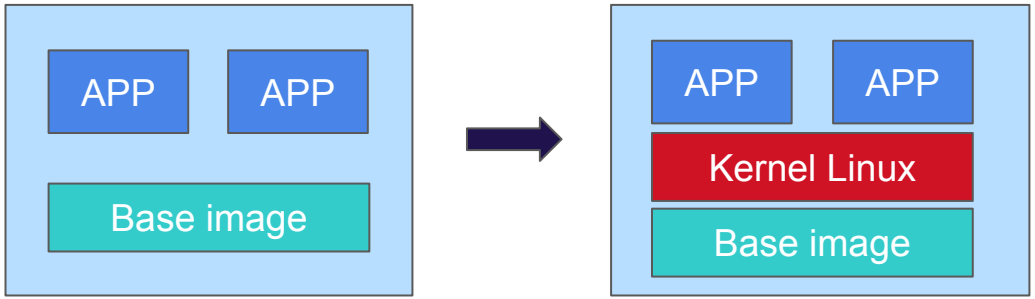
Premier démarrage d'une machine confidentielle



Calcul des valeurs de référence

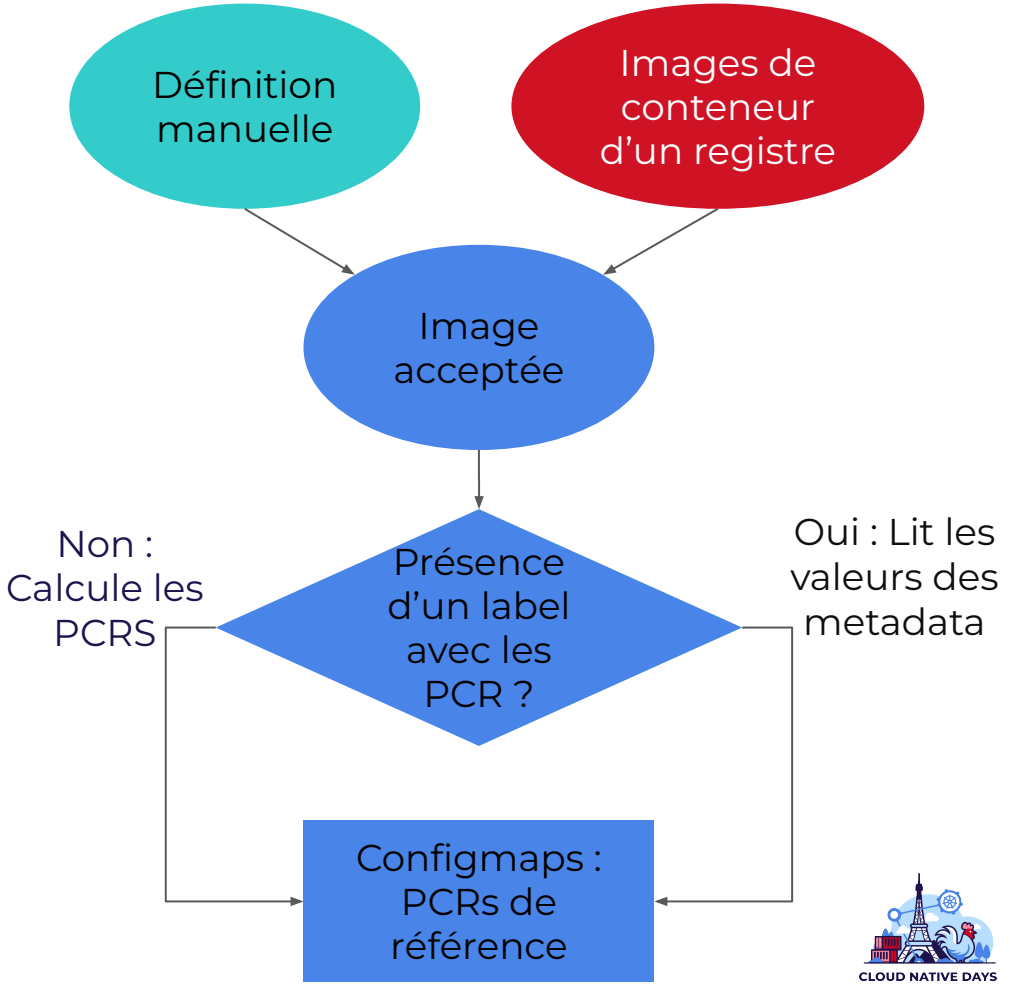
Conteneurs bootables (Bootable Containers)

- Besoin d'**automatiser le calcul** des valeurs de référence
- **Conteneurs bootable** ("bootable containers" ou bootc) :
  - OS distribué à l'aide d'une **image de conteneur OCI**
- Facilite le calcul des valeurs de référence pour un OS
  - Image et format unique pour toutes les plateformes cloud



Approbation des images de l'OS

```
apiVersion:
confidential-containers.io/v1alpha1
kind: ApprovedImage
metadata:
  name: my-fedora-coreos-image
  namespace: confidential-clusters
spec:
  reference:
quay.io/my-registry/fcos:stable
```



Démonstration

Vision future et conclusion

Un regard sur l'avenir

- Intégration avec **Cluster API** (CAPI) pour une meilleure gestion des machines (https://cluster-api.sigs.k8s.io/)
- **Protection** de la configuration de Ignition avec une double attestation à distance
- Configuration et attestation à distance du **premier noeud** du cluster (bootstrap du cluster)

Le mot de la fin

L'opérateur :

- **gère** toute la complexité du calcul des valeurs de référence et des politiques de sécurité
- **déploie** Trustee et coordonne le démarrage des machines confidentielles
- est **responsable** de la vision globale du cluster et de l'état des noeuds

Ainsi la bergerie est en sécurité des loups numériques

https://github.com/trusted-execution-clusters

Pour nous contacter :

Alice Frosi

afrosi@redhat.com

Timothée Ravier

Mastodon : siosm@floss.social

Avis :

Merci