Virtualisation et sécurité

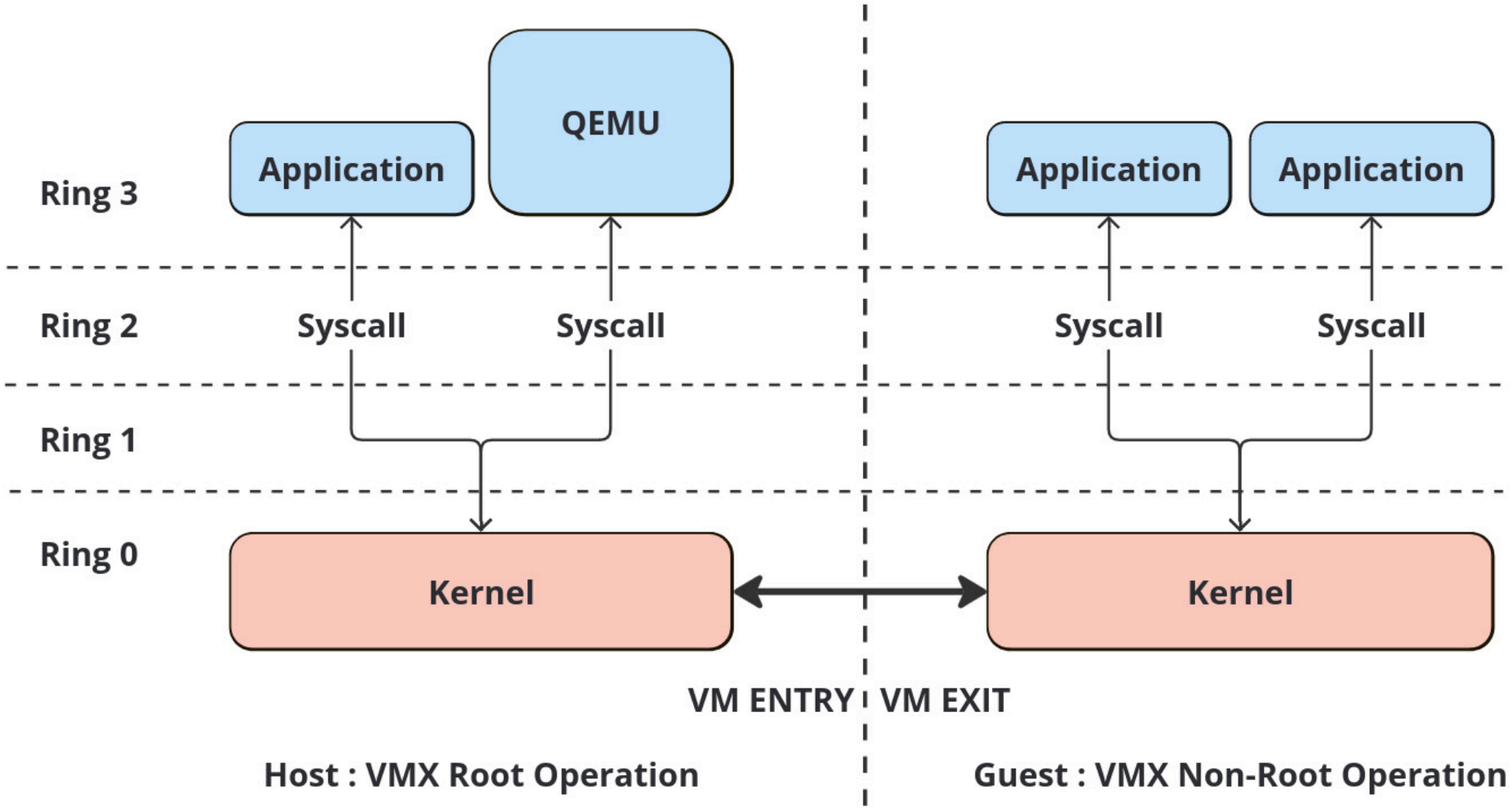
Timothée Ravier
siosm@floss.social - tim.siosm.fr/cours - github.com/travier

5e année cycle ingénieur, filière STI
Option Sécurité des Systèmes Embarqués et du Cloud (2SEC)

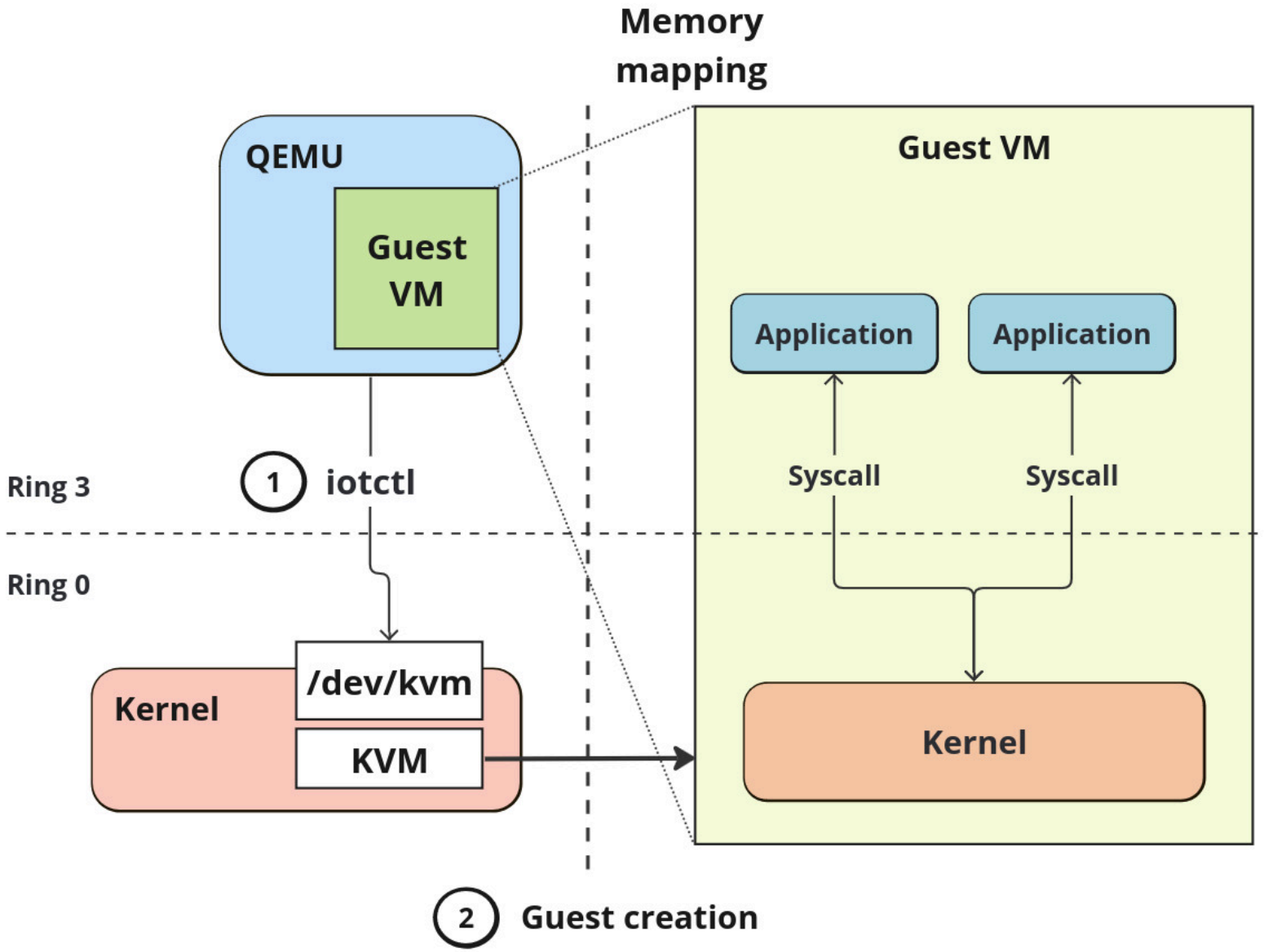
2024 - 2025

Virtualisation avec KVM/QEMU

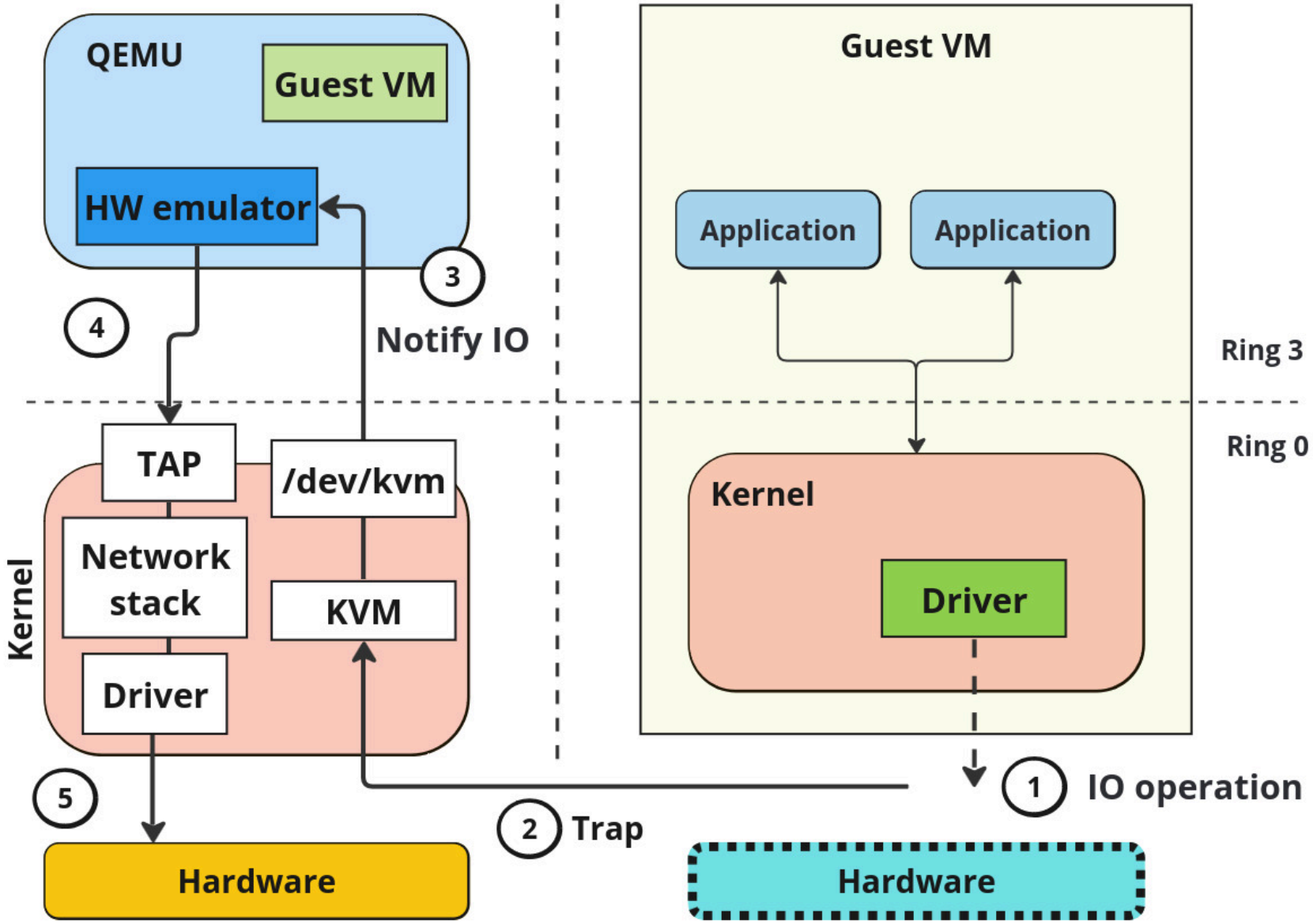
Principe général : KVM & VT-x (Intel)



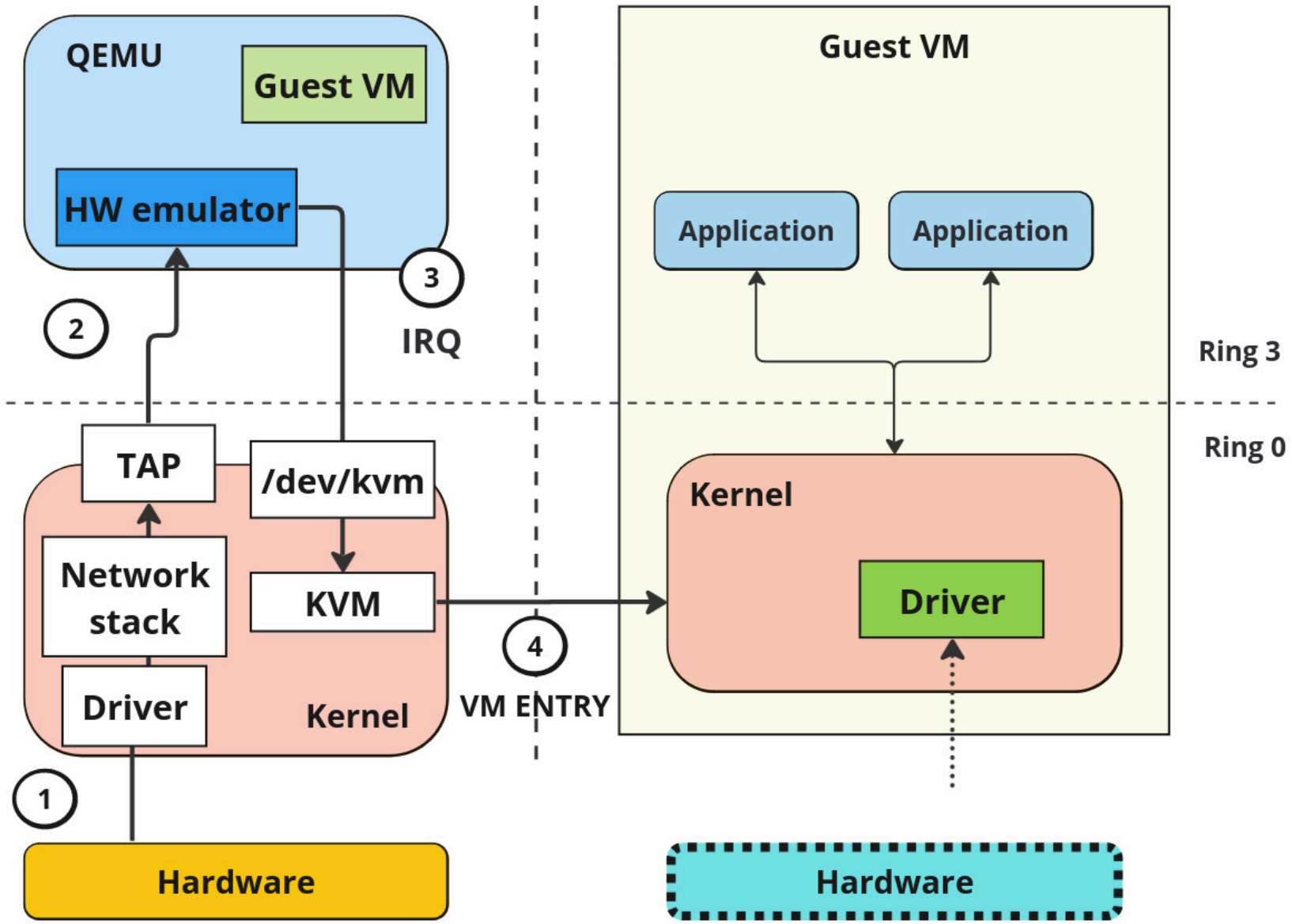
Création d'une VM avec QEMU



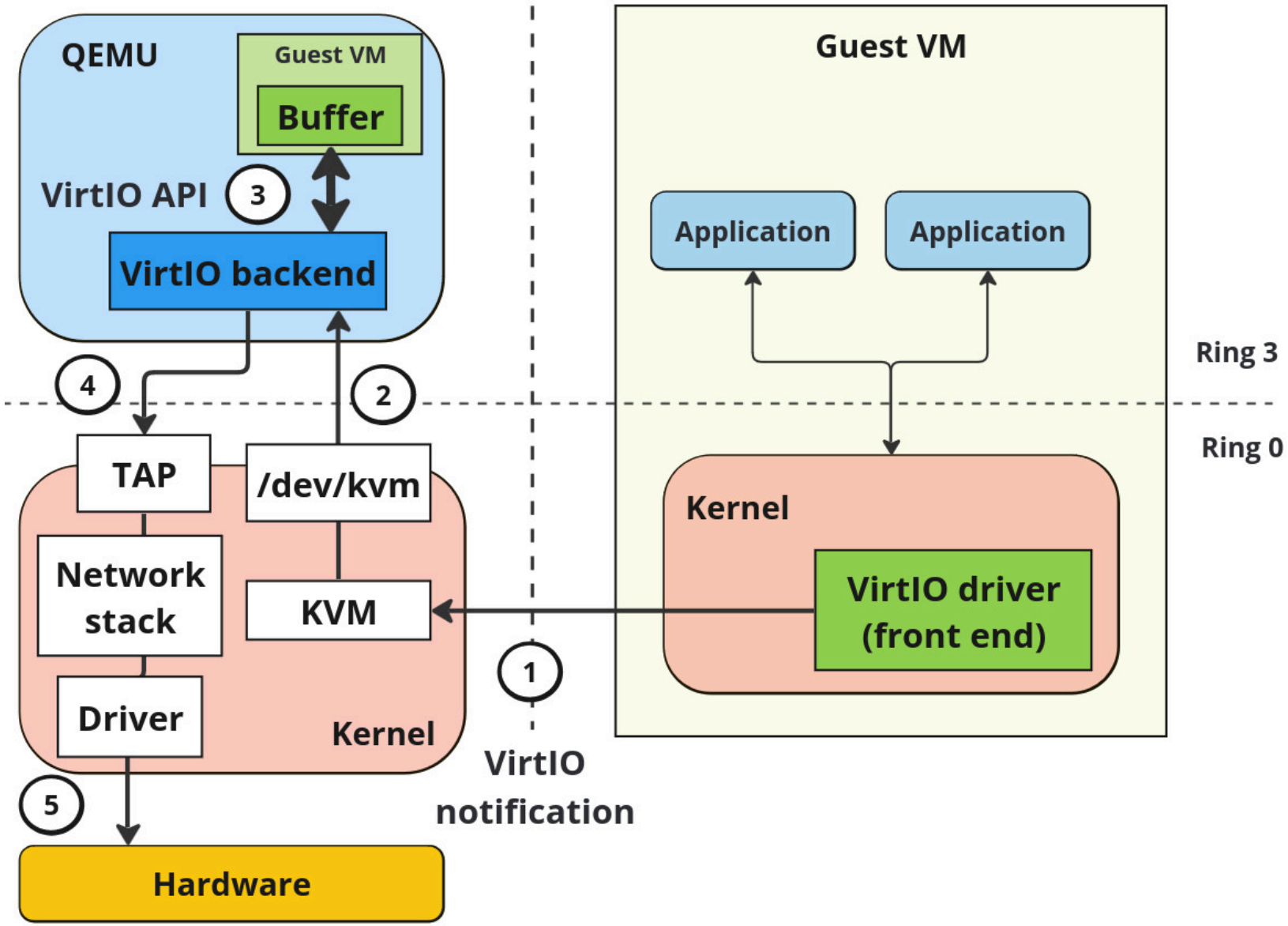
Matériel émulé (QEMU) : envoi



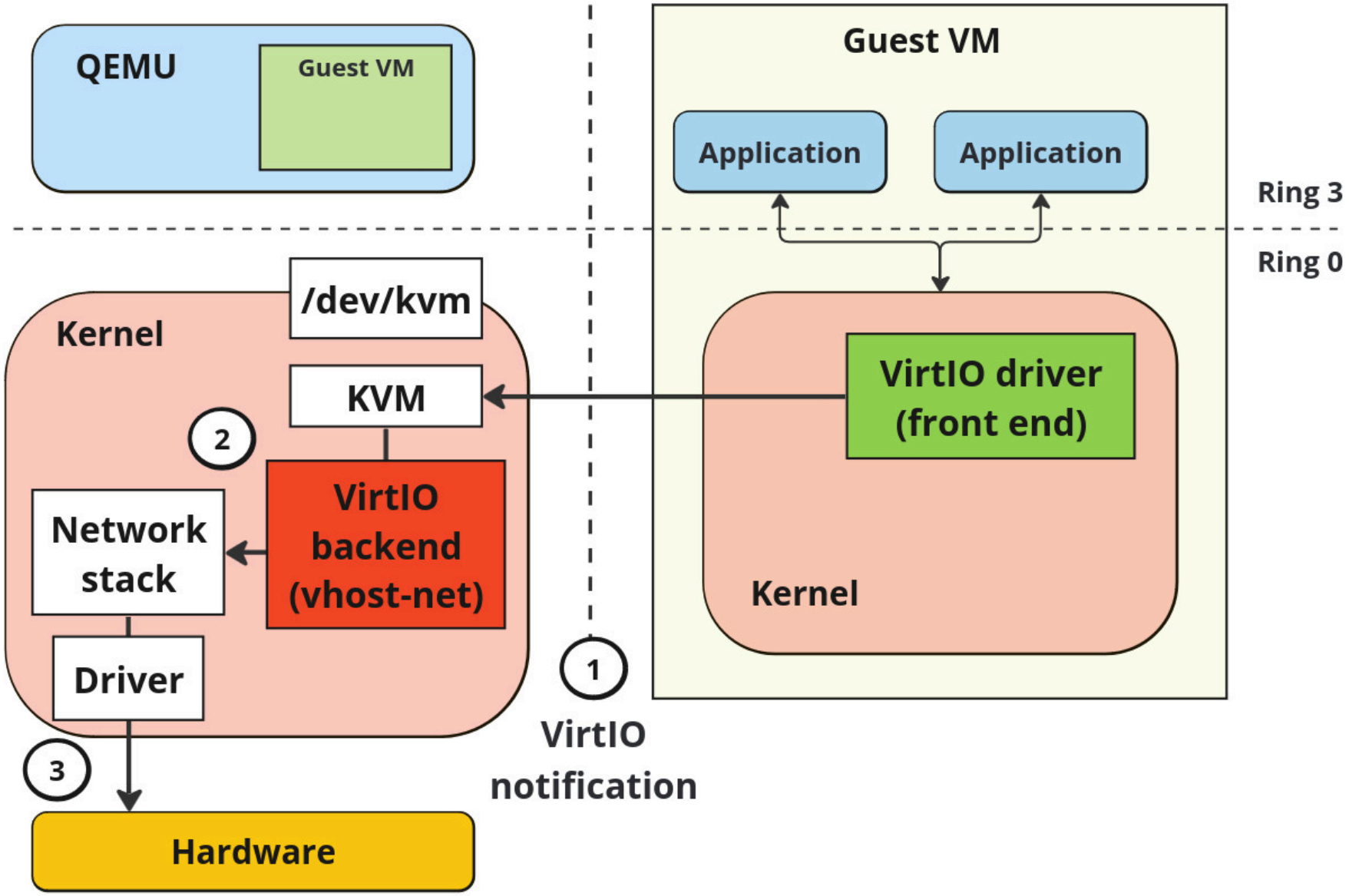
Matériel émulé (QEMU) : réception



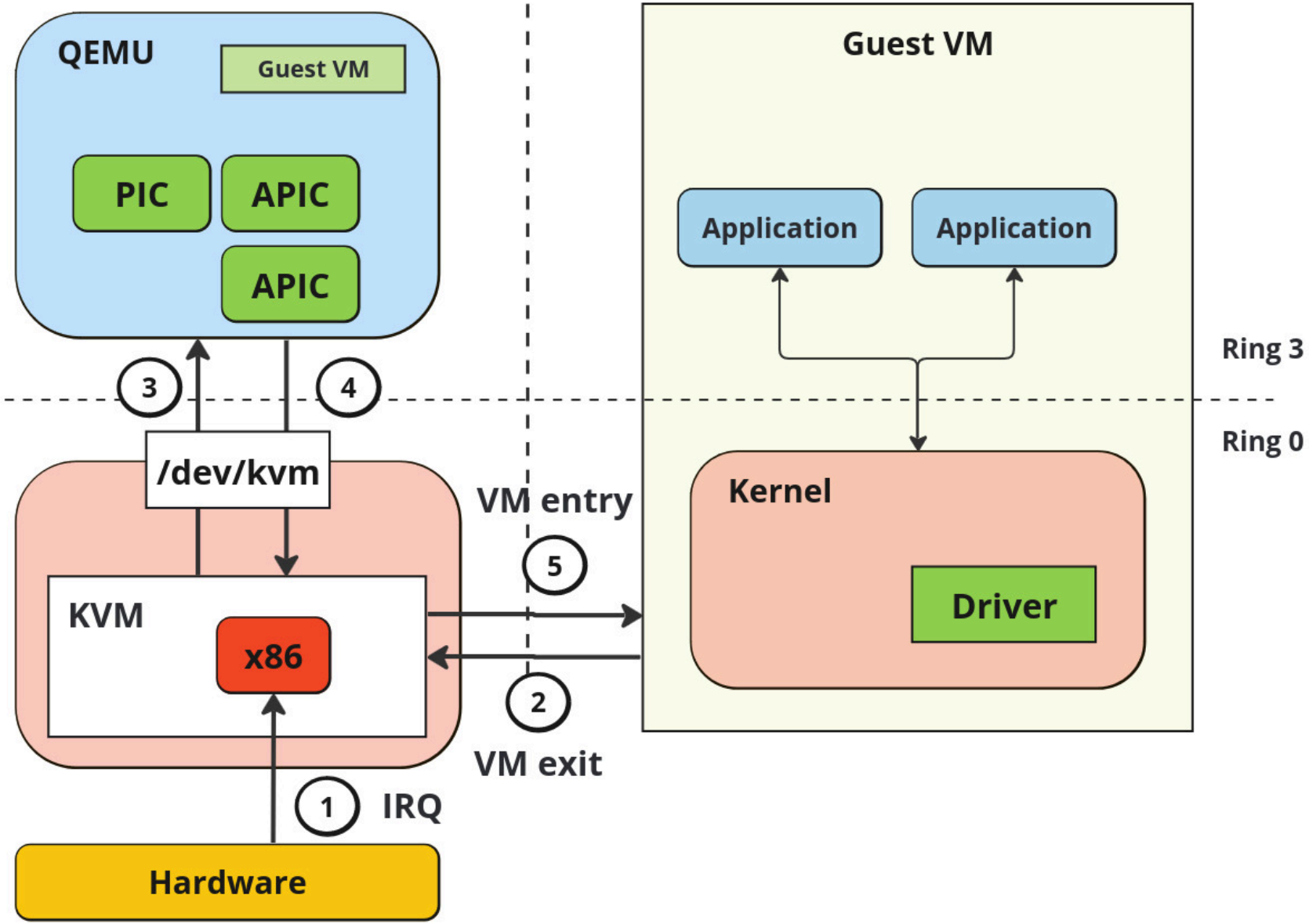
Matériel virtuel : VirtIO (QEMU)



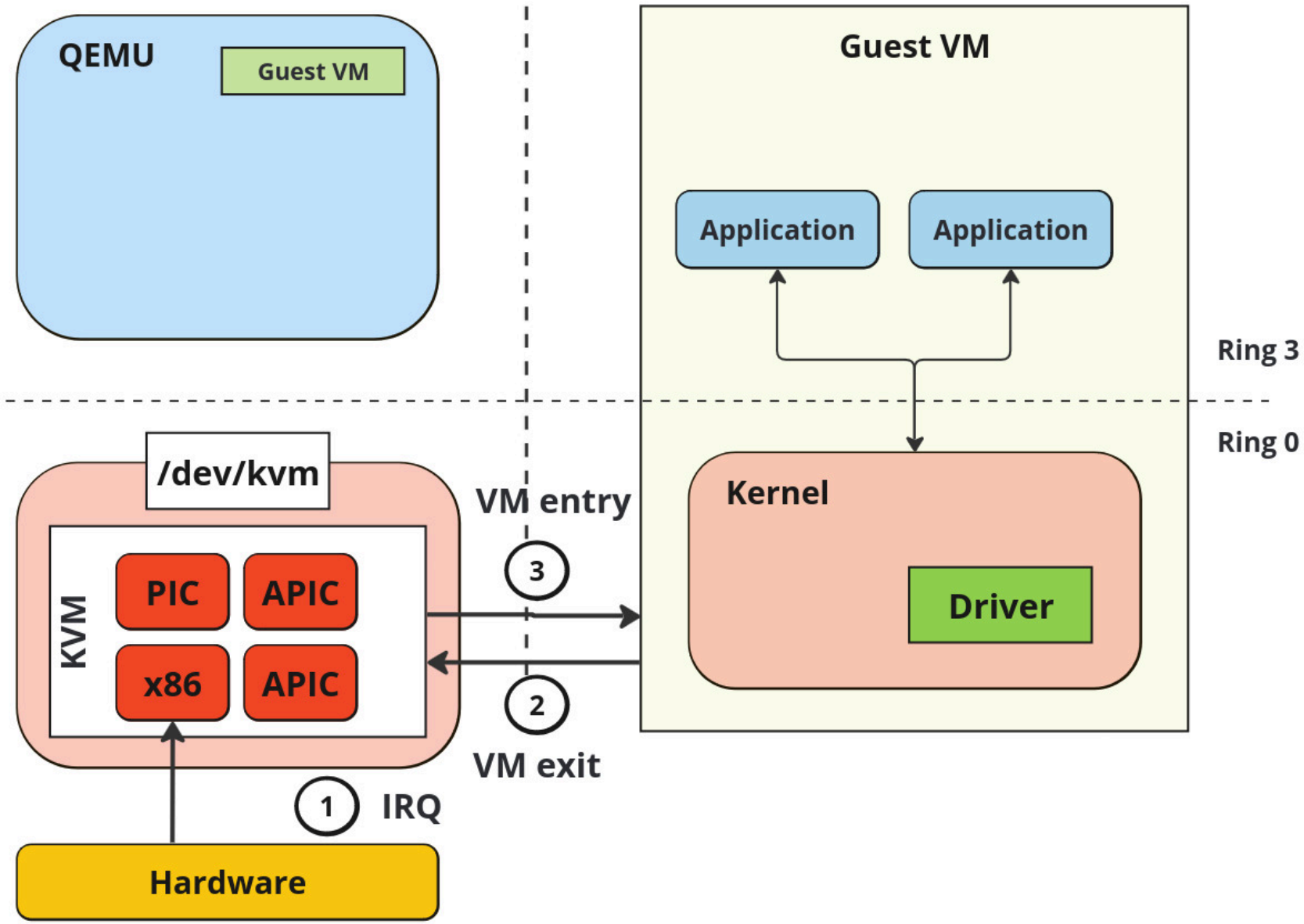
Matériel virtuel : VirtIO (Kernel)



Composants émulés (QEMU)



Composants émulés (Kernel)



Scénarios d'attaque et vulnérabilités

Virtualisation == sécurité ?

La virtualisation n'est pas un composant de sécurité !

Virtualisation == sécurité ?

Plus d'isolation, mais :

- Matériel plus compliqué
- Plus de versions de systèmes (noyau, espace utilisateur)
- Plus de réseaux à gérer
- Plus d'interfaces d'administration

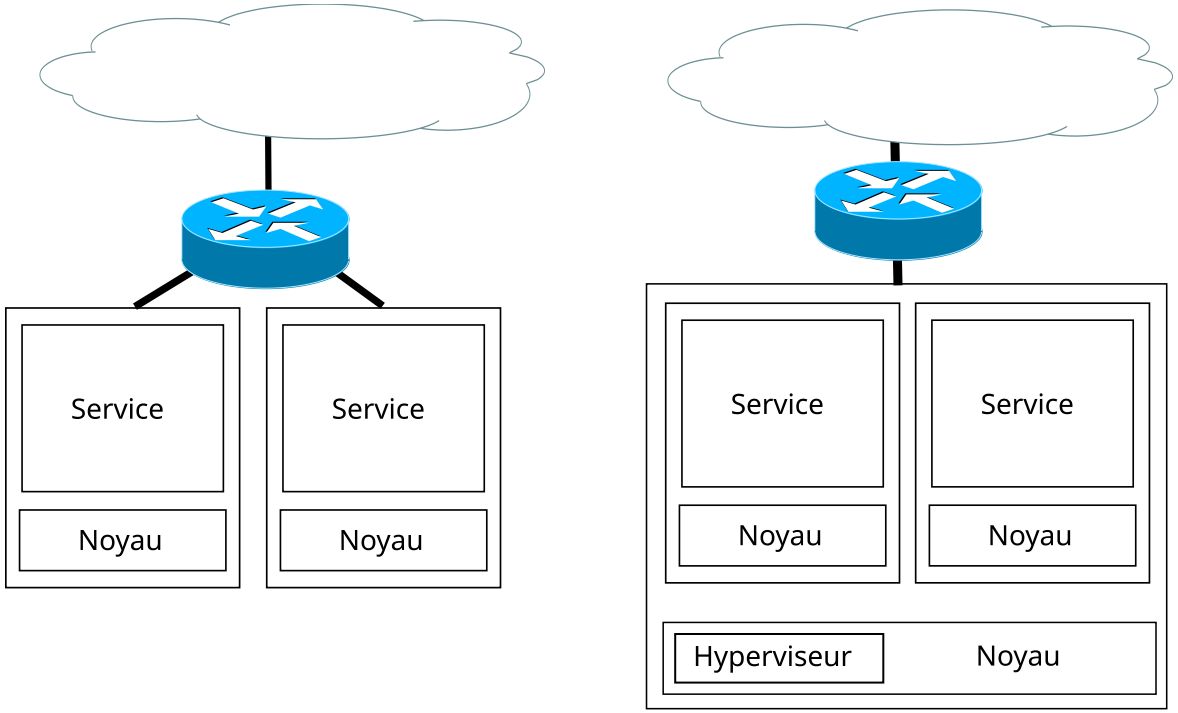
Virtualisation == sécurité ?

- La virtualisation ne dispense donc pas des autres solutions pour assurer la sécurité
- Au contraire, elle en ajoute à appliquer

Virtualisation == sécurité ?

Pour s'en convaincre, il faut comparer :

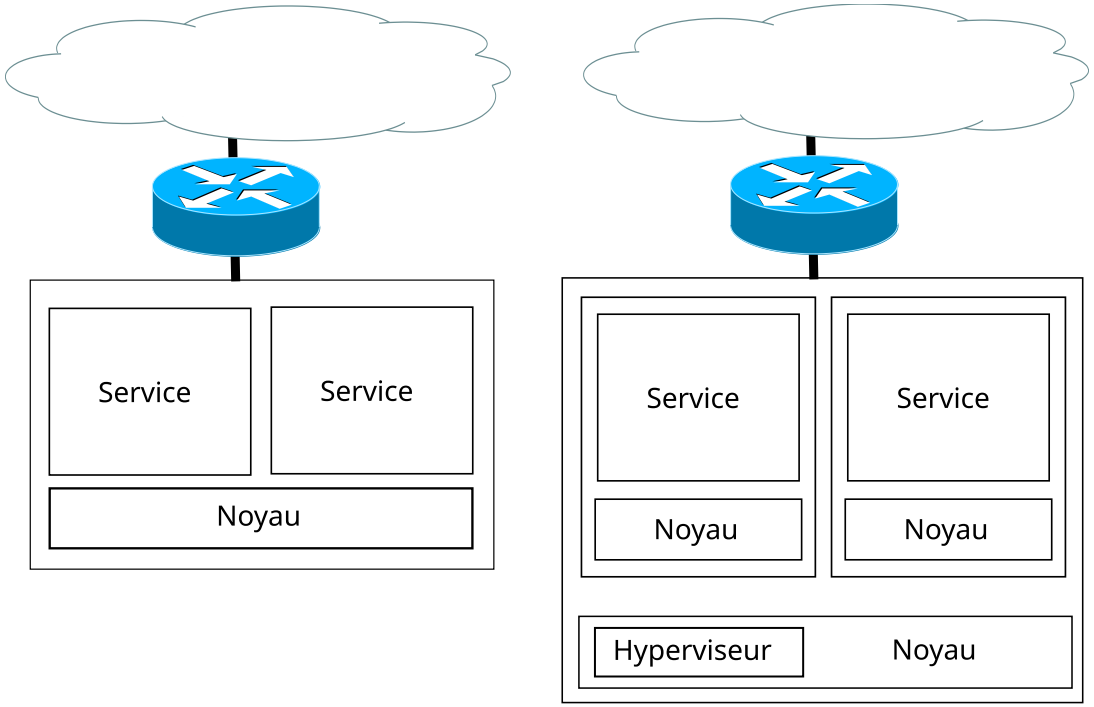
- Services sur des machines physiquement distinctes vs machines virtuelles



Virtualisation == sécurité?

Pour s'en convaincre, il faut comparer :

- Services sur une même machine vs services dans une ou plusieurs machines virtuelles



Scénario d'attaque

Scénario d'attaque : Machine virtuelle

- La machine virtuelle (le système, les services ou les données) sont visés
- Principalement des attaques à distance similaires aux attaques classiques pour les machines physiques
- Les machines virtuelles ne sont pas exemptées des contraintes de sécurité appliquées aux machines physique

Scénario d'attaque : Autres VMs

- Corrompre ou espionner les autres machines virtuelles sur le même hyperviseur

Scénario d'attaque : Système hôte

- Sortir de la machine virtuelle pour attaquer le système hôte et les autres machines virtuelles
- Attaquer ou corrompre l'hyperviseur :
  - faille dans l'hyperviseur ou les drivers virtuels
  - faille dans le système hôte

Scénario d'attaque : Virtualisation à la volée

- Virtualiser à la volée un système non-virtualisé sans visibilité pour l'utilisateur
- Théoriquement possible (Blue Pill)
- Invisibilité contestée en pratique : il est très difficile de masquer à un système qu'il est virtualisé

Virtualisation et sécurité

Eléments à durcir

- Sécurité de l'hyperviseur :
  - Matériel / Noyau / OS
  - Gestionnaire de machines virtuelles (VMM)
  - Emulateur / virtualiseur
  - Isolation entre les machines virtuelles
- Sécurité des machines virtuelles :
  - Noyau / OS
  - Applications
  - Mise à jour
- Sécurité des images de machines virtuelles :
  - Distribution et intégrité
  - Distribution des secrets

Matériel

- Matériel sécurisé ?
- Canaux cachés
- Matériel de confiance
- Cf. chapitre sur la sécurité des conteneurs.

Matériel : support de la virtualisation

Surface d'attaque supplémentaire :

- Fonctions de virtualisation dans les processeurs
- CPU et MMU : SVM/AMD-V (AMD), VT-x (Intel) :
  - The Intel SYSRET privilege escalation
- IOMMU : AMD-Vi (AMD), VT-d (Intel)

IOMMU & PCI passthrouh

- IOMMU intéressante dans tous les cas : contrôle d'accès entre prériphériques PCI-Express
- PCI passthrough et SR-IOV à éviter :
  - firmware du matériel accessible ?
  - interfaces cachées ?
  - quelle isolation entre les différents contextes par le matériel ?
  - exemple : cartes graphiques : XDC2012: Graphics stack security
- Utiliser VFIO

Network Function Virtualization, Packet Processing Performance of Virtualized Platforms with Linux and Intel Architecture

Noyau Linux

- Point commun entre toutes les machines virtuelles
- Hyperviseur : module KVM : (~30k SLOC)
  - `/dev/kvm` : interface `iotcl`
  - 2 hypercalls : `KVM_HC_VAPIC_POLL_IRQ` ($\Leftrightarrow$ `VM_EXIT`) et `KVM_HC_KICK_CPU`
- Drivers spécifiques :
  - vhost-net
  - vhost-scsi

Durcissement KVM

Eviter :

- les drivers dans le noyau : pas de vhost-net & vhost-scsi
- les émulateurs matériels dans le noyau : pas de PIC, APIC, IOAPIC :
  - à désactiver avec des options pour QEMU
  - KVM Security Improvements, Andrew Honig, KVM Forum 2014
  - Performant Security Hardening, Steve Rutherford, KVM Forum 2016

Eviter Xen

- De très nombreuses vulnérabilités : https://xenbits.xen.org/xsa/
- Dernier Cloud provider majeur : Amazon AWS
- Migration en cours vers KVM (Nitro)

Ducissement du noyau Linux

- Cf. chapitre sur la sécurité des conteneurs

Système d'exploitation

- Le système d'exploitation hôte de l'hyperviseur est un système classique
- Cf. chapitre sur la sécurité des conteneurs

Gestionnaire de machines virtuelles (VMM)

- VMM généralement très privilégié :
  - Gestion du réseaux, des images disques, des secrets, etc.
- Contrôle des accès au VMM :
  - Accès au VMM $\Leftrightarrow$ `root`
  - Situation similaire à Docker
- Exemple pour libvirt :
  - Accès à la socket libvirt en local
  - Accès à distance en TCP

Durcissement de l'émulateur / virtualiseur

- Emulateur : QEMU (~500k SLOC)
- Processus en espace utilisateur :
- Durcissement : Réduction de la surface d'attaque :
  - Utiliser le moins possible d'émulateurs de matériel
  - Options de compilation et durcissement
- Intégration du support de Rust en cours

Alternatives à QEMU ?

- The Chrome OS Virtual Machine Monitor : Maintenu par Google pour Chrome OS
- Firecracker : Maintenu par AWS pour Lambda et Fargate
  - Attacking Firecracker: AWS' microVM Monitor Written in Rust
- cloud-hypervisor : Maintenu par Intel, virtualisateur x64, périphériques virtio
- Point commun de ces alternatives : Rust (rust-vmm)

Cloud providers ?

- Google Compute Engine :
  - 7 ways we harden our KVM hypervisor at Google Cloud: security in plaintext
- Amazon Web Services Nitro :
  - AWS EC2 Virtualization 2017: Introducing Nitro
  - The Security Design of the AWS Nitro System

Isolation entre les machines virtuelles

- Confinement des machines virtuelles $\Rightarrow$ confinement de l'émulateur / virtualiseur
- Réduction de la surface d'attaque du noyau : seccomp-bpf
- Isolation avec SELinux :
  - Confinement des instances de QEMU à l'aide des catégories (sVirt)
  - Ensembles de catégories associées aux images disques des machines virtuelles
- Voir chapitre sur la sécurité des conteneurs

Confidential Computing

- Protéger les machines virtuelles de l'hôte
- Intel TDX, AMD SEV-SNP et IBM Z Secure Execution
- Chiffrement de la mémoire des machines virtuelles
- Co-processeur chargé de la gestion des clés
- Chiffrement des disques nécessaire
- Nécessite des TPM virtuels (vTPM), Secure Boot, Remote Attestation
- Disponible à divers degrés sur GCP, Azure, AWS Nitro, IBM Cloud

Sécurité des machines virtuelles

- Système d'exploitation complet
- Contraintes similaires : mise à jour, durcissement, etc.
- Voir chapitre sur la sécurité des conteneurs

Sécurité des machines virtuelles : `root`

- Protection du compte `root` et du noyau impérative
- La quasi totalité des vulnérabilités dans les hyperviseurs nécessitent d'être `root` (Ring 0) dans la machine virtuelle
- Séparation de privilèges et défense en profondeur nécessaire

Entropie dans une machine virtuelle

- Beaucoup de composants matériels émulés dans une VM
- Comportement très peu aléatoire
- Entropie de mauvaise qualité
- Une seule solution : VirtIO-RNG
- Attention aux « fausses bonnes idées » (haveged, pollinate, etc.)

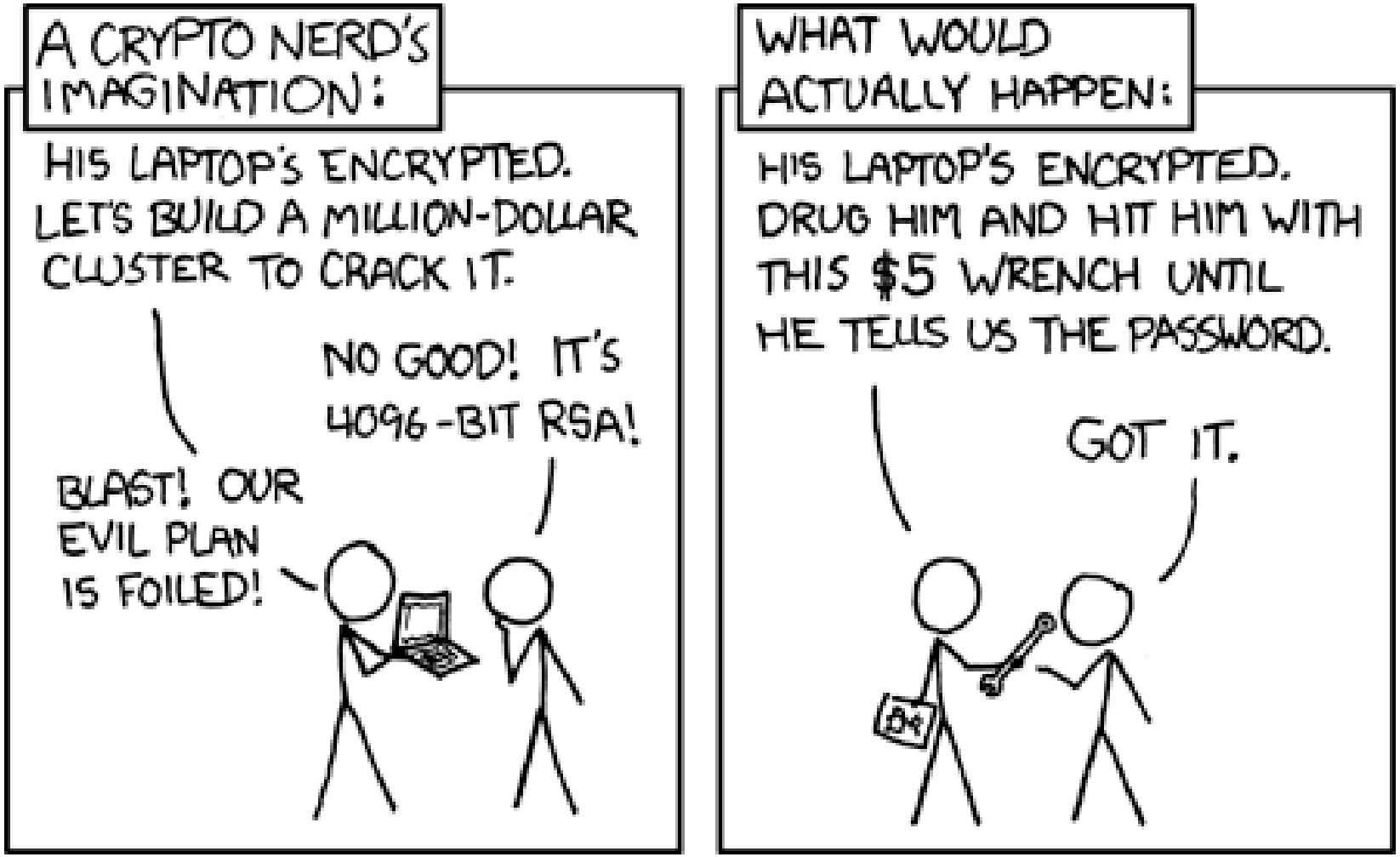
Images de machines virtuelles

- Vérifier l'origine et l'intégrité des images de machines virtuelles
- Considérer les images de machines virtuelles comme publiques
- Utiliser d'autres mécanismes pour provisionner des secrets :
  - cloud-init, Ignition, user-data, etc.

Hôte et système de fichier

- Il est très fortement déconseillé de monter directement le système de fichiers d'une machine virtuelle sur la machine hôte :
  - A reminder why you should never mount guest disk images on the host OS
  - `mount` détermine quel module doit être chargé en fonction du système de fichiers : il peut donc être trompé
  - Une faille dans un driver de système de fichier ou dans la couche VFS du noyau Linux peut compromettre toute la machine
- Il faut donc utiliser d'autres outils qui le font indirectement avec FUSE : `libguestfs`

Surtout ne pas oublier l'analyse des risques !



http://xkcd.com/538/

Suite

Sécurité des infrastructures