Développement de moyens matériels pour la protection des couches basses du logiciel

Benoît Morgan, Éric Alata, Vincent Nicomette

Réunion du groupe RéSIST
LAAS-CNRS, INSA Toulouse

15 décembre 2014

Plan

Problématique

- Systèmes informatiques de plus en plus complexes
  - Plusieurs couches logicielles et matérielles empilées
- Cycle de vie complexe
  1. Démarrage
  2. Configuration par le BIOS localisé sur la machine
  3. Chargement du système d'exploitation
  3. Chargement du gestionnaire de machines virtuelles
     1. Démarrage
     2. Chargement du système d'exploitation
  4. Exécution d'applications
- Exécution supportée par des platformes matérielles (x86, PCI Express)

Problématique

- Les approches classiques nécessitent de faire confiance à ces piles logicielles et matérielles.
- Persistance d'erreurs de configuration ou de bugs logiciels et matériels
  malgré les bonnes pratiques de sécurité

$\Rightarrow$ Utilisation de ces erreurs pour modifier ou compromettre les couches

Comment s'assurer que l'intégrité d'un logiciel n'a pas été compromise

État de l'art

- Application du principe d'isolation
  - Utilisation des privilèges du processeur
  - Problème des attaques profitant du matériel
- Surveillance du noyau par un module plus privilégié
  - Hytux : Linux Kernel Module (inspiré par Blue Pill)
  - BMCS : KVM
  - HIMA : Xen
  - HyperTap
  - TinyChecker (nested virtualization)
- Identification d'activités malveillantes dans les machines virtuelles
  - SIMA
- Surveillance du système depuis un mode "exotique" du processeur
  - HyperCheck : SMM
- Instrumentation du composant avec un modèle de comportement
  - HyperSafe

État de l'art

- Identification d'un composant matériel compromis
  - Viper (depuis un module du noyau)
  - IOCheck (depuis le mode SMM)
- Identification de la compromission des couches basses depuis le matériel
  - Copilot

État de l'art

- Test du logiciel depuis un composant logiciel
- Test des logiciels bas niveau depuis un composant matériel
- Test des logiciels bas niveau depuis un mode "exotique" du processeur

$\Rightarrow$ Compromission du comportement d'un composant sans modifier son code

$\Rightarrow$ Difficulté du test d'un logiciel haut niveau depuis une couche basse

Projet SVC

- Secure Virtual Cloud
- Projet Investissement d'Avenir
- Développement d'un cloud sécurisé
- 10 entreprises et laboratoires
- LAAS-CNRS et IRIT
- 14 millions d'euros sur 3 ans

Le cloud computing

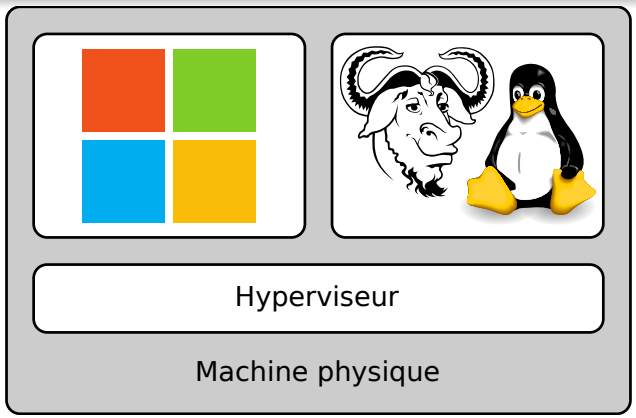
Intérêt du cloud

- Gain de performances
- Utilisation temporaire de logiciels
- Économies :
  - Achat
  - Maintenance
- Disponibilité

Problèmes de sécurité

- Confidentialité ?
- Intégrité ?

Utilisation de machines virtuelles



Utilité du gestionnaire de machines virtuelles

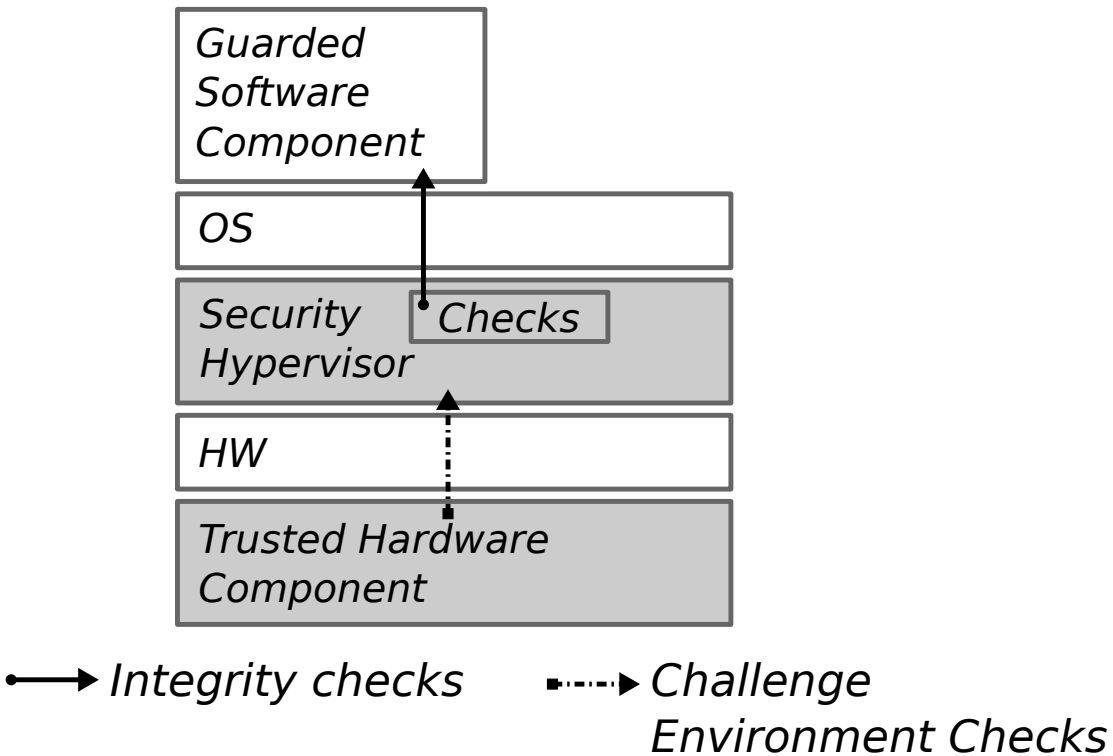
- Isolation spatiale
- Isolation temporelle
- Protection de la machine virtuelle

Corruption du Gestionnaire de Machines Virtuelles

- Perte de l'isolation spatiale, mémoire
- Perte de l'isolation temporelle, temps d'exécution

Plan

Architecture globale pour le test de composants logiciels



⇒ Enclave d'exécution de tests d'intégrité assistée par le matériel

Application au projet SVC

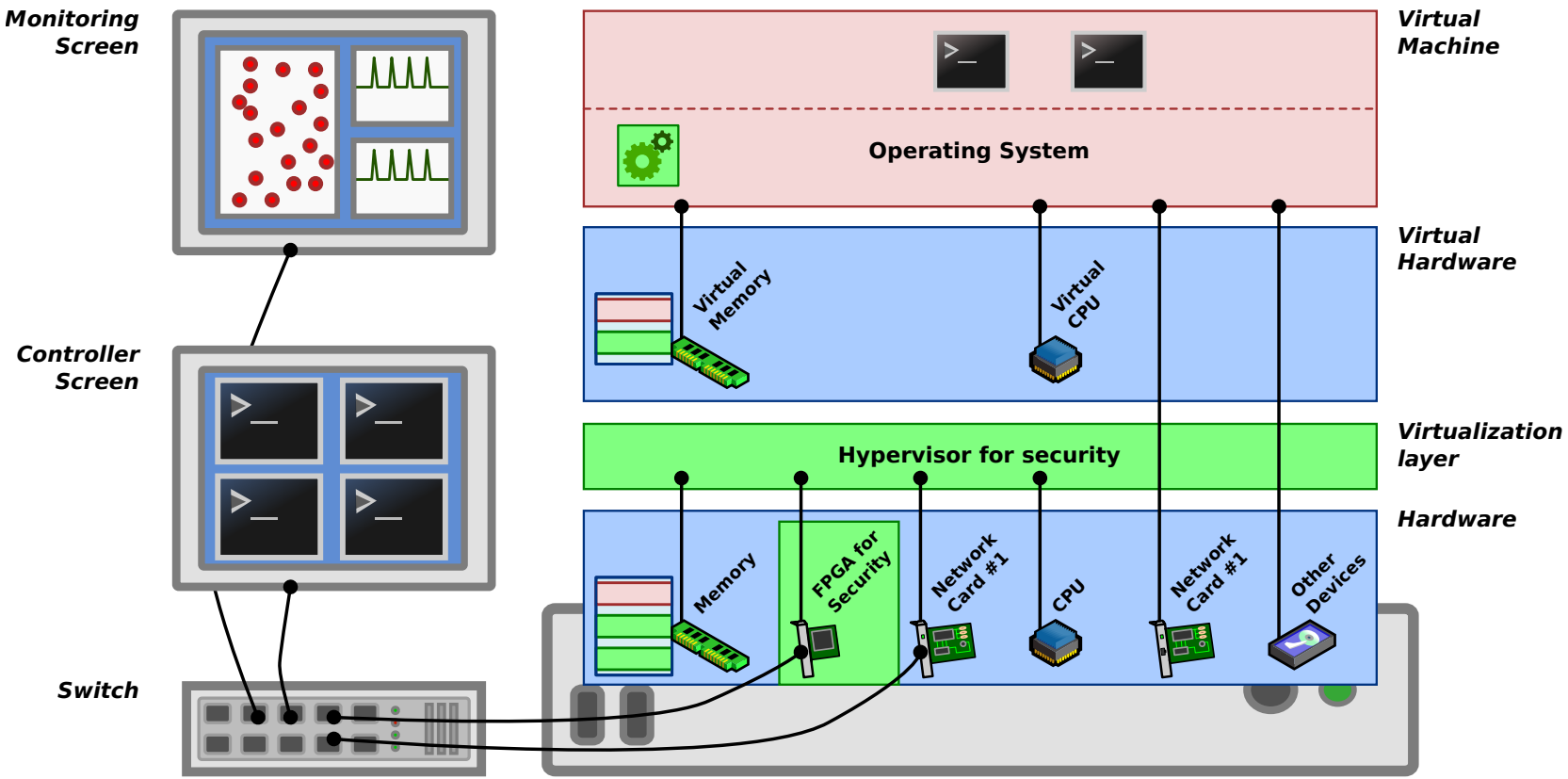
Étude de recherche dans le cadre du projet SVC

- Guarded Software Component : Hyperviseur VMware

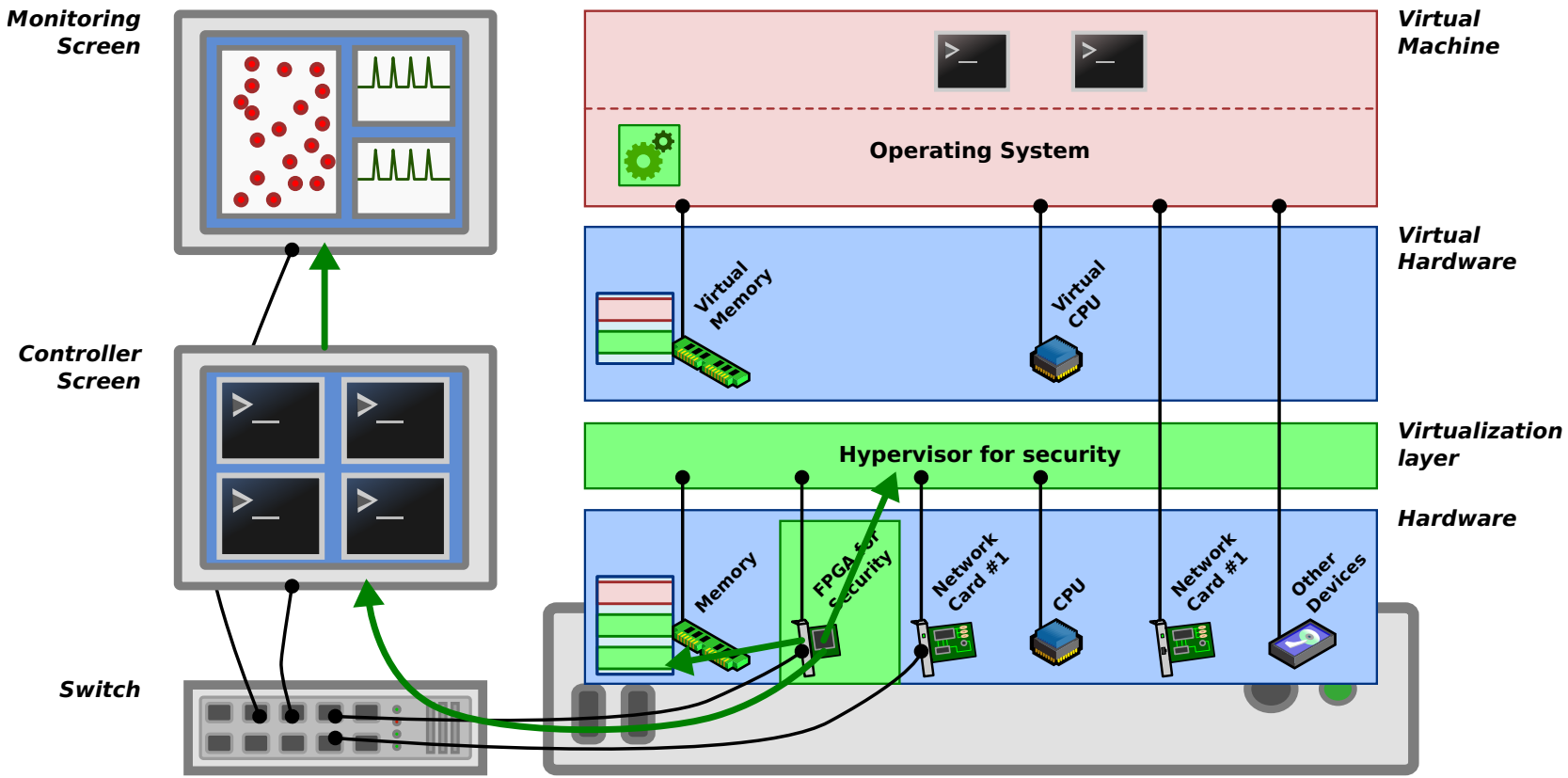
Objectifs préliminaires

- Guarded Software Component : Applications et drivers GNU / Linux
- Hyperviseur de sécurité
  - Nested ready
  - Mono core
  - Enclave : exécution des tests d'intégrité des guarded softwares
- Trusted Hardware Component
  - Compatible PCI Express
  - Indépendant du CPU
  - Challenges de l'enclave

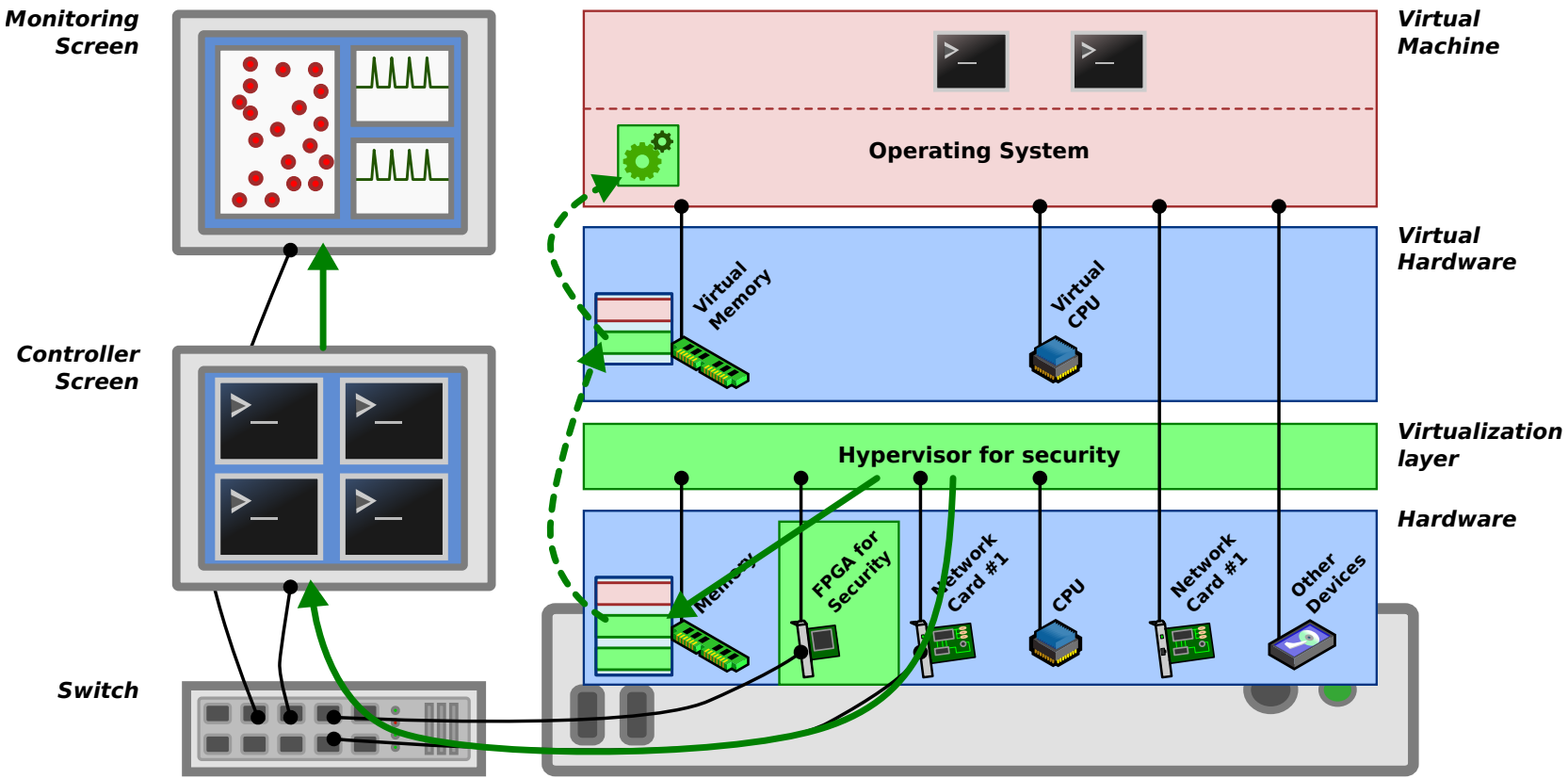
Architecture globale pour le test de composants logiciels



Protocole de tests : Étape 1

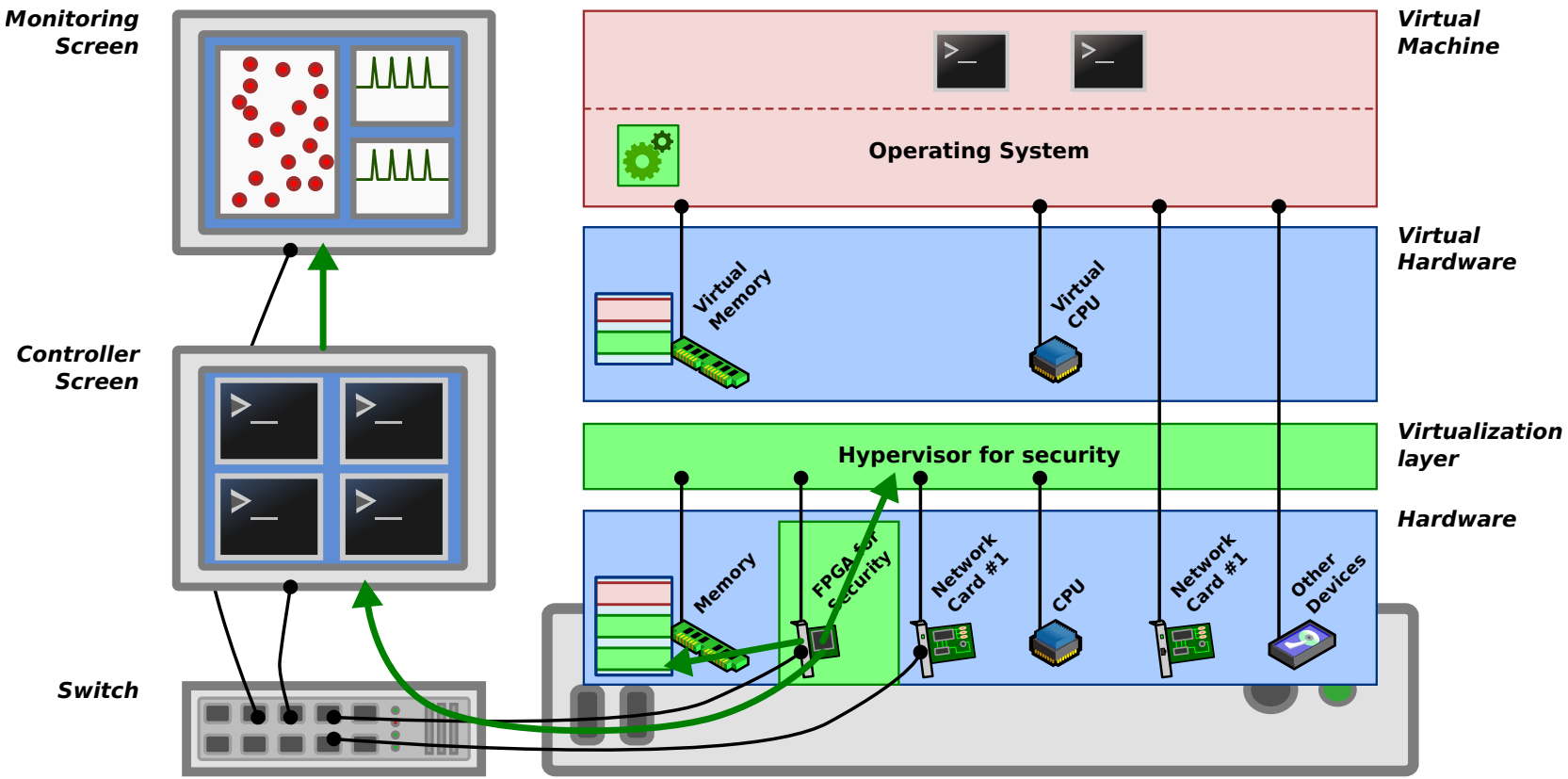


Protocole de tests : Étape 2



Plan

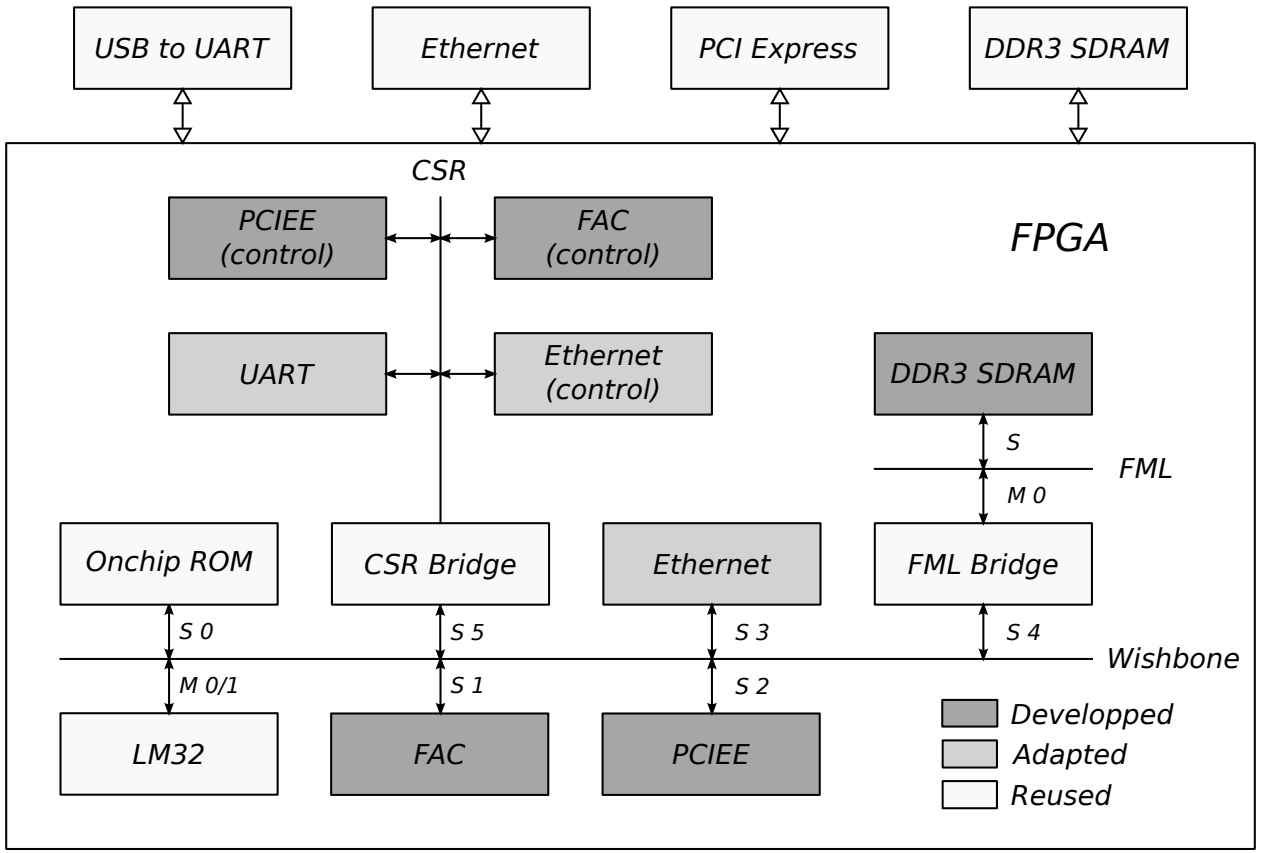
Protocole de tests : Étape 1



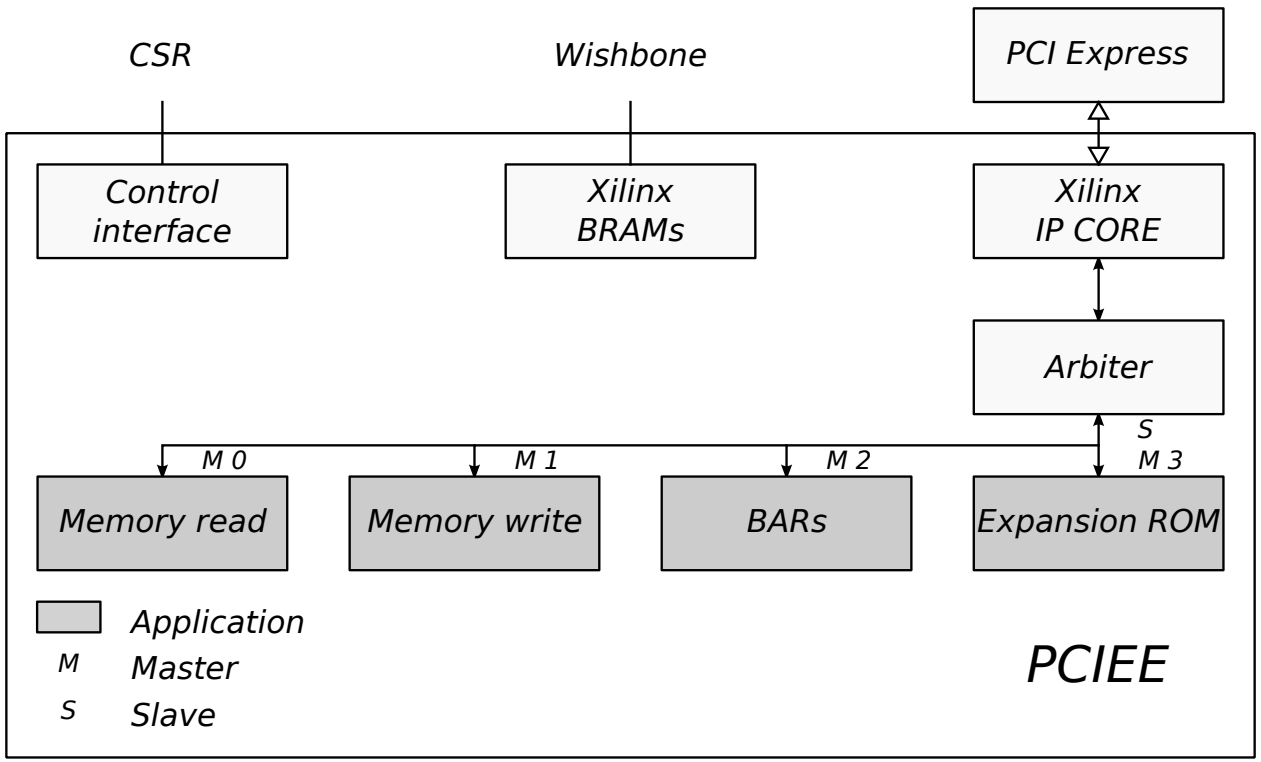
Choix

- FPGA
  - Prototypage rapide
  - Performances réalistes
- Basé sur Milkymist
  - System On Chip pour des effets vidéos temps réel

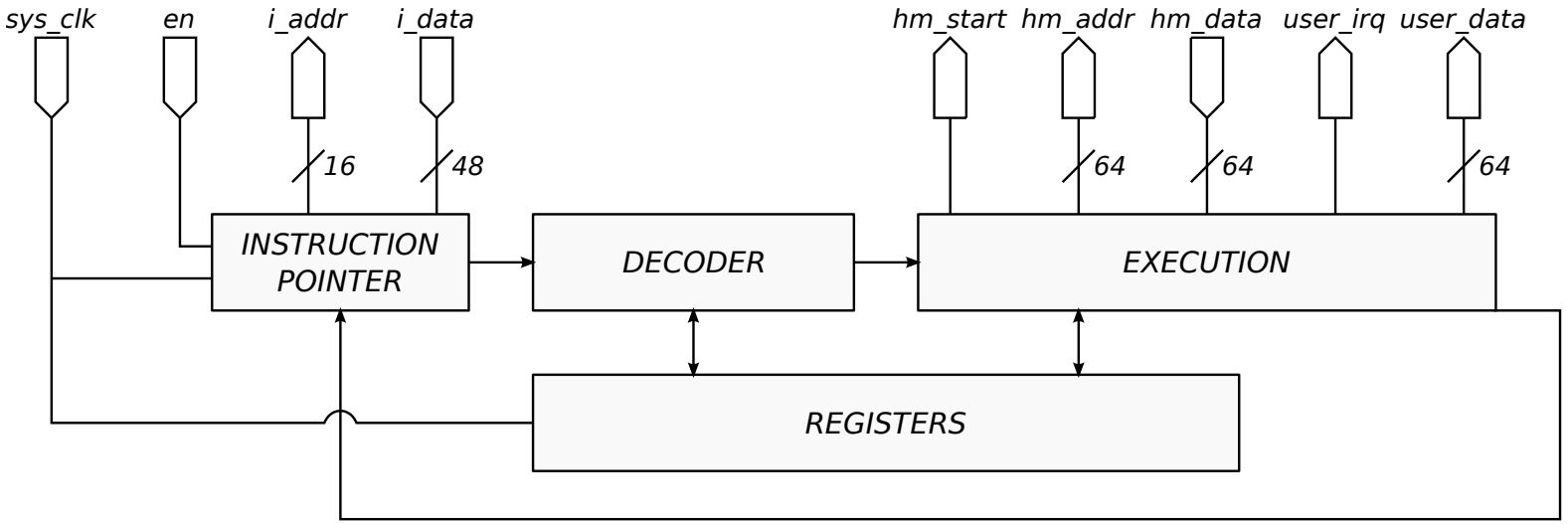
System On Chip



PCI Express Endpoint



Fast Automata Core



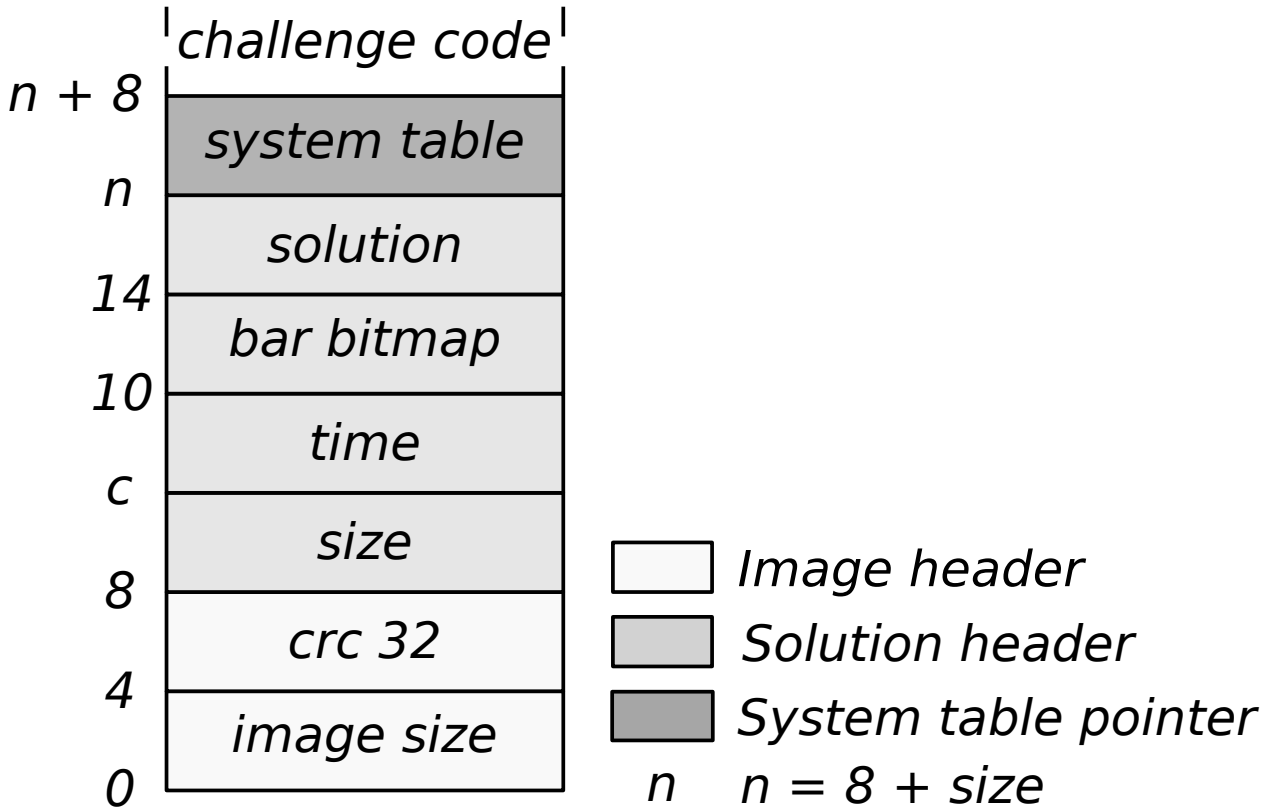
Code	Mnemonic	Description
0x1	mask	Jump if allowed / not allowed mask not applicable
0x2	equ	Jump if not equal
0x3	low	Jump if not lower
0x4	add	Add two registers
0x4	xor	Xor two registers
0x5	hamm	Hamming distance
0xc	int	Interruption
0xd	mload	Memory load
0xe	load	Immediate load
0xf	jmp	Jump

Mode Challenge

1. Démarrage depuis le BIOS intégré dans la ROM
2. Initialisation des composants
3. Téléchargement du BIOS "Mode challenge"

1. Téléchargement du challenge courant dans l'expansion ROM
2. Attente d'un période d'exécution
3. Fin de lecture de l'expansion ROM
4. Attente de l'écriture de la solution
5. Vérification en temps et en valeur
6. Tests d'environnement avec le FAC

Format du challenge



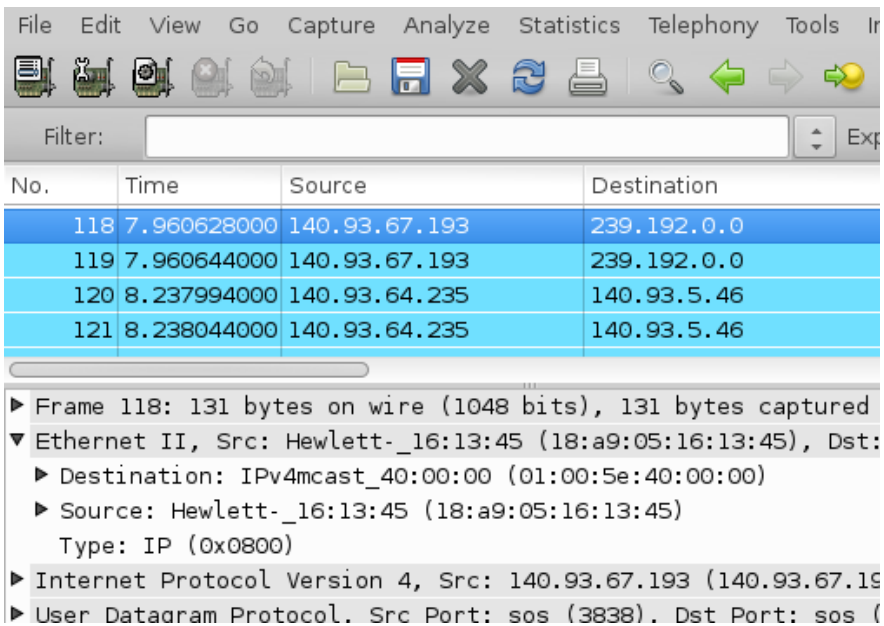
Outils développés

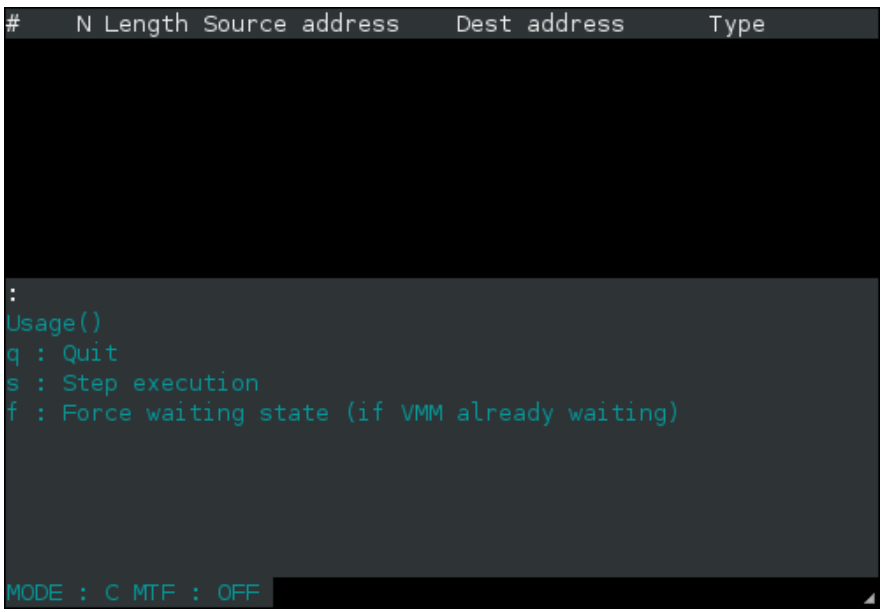
- Assembleur pour le FAC (lex & yacc)

```
_start:
  loadd r31, $0x9 // hamm < 9
  loadd r30, $0xff8 // 512 entries
  loadd r29, $0x8 // Increment
  loadd r28, $0xffffffff // Hamming mask
  loadd r28_1, $0xfffff000 // Hamming mask
  loadd r27, $0x0 // i
  loadd r20, $loop // lood start
  loadd r21, $endloop // lood end
  loadd r10, $0x1 // ok
  loadd r11, $0x2 // ko
  loadd r1, $0x0 // entry i
  loadd r2, $0x0 // entry i + 1
  loadd r3, $0x0 // computed hamm
// for (i = 0; i < 0x1000 - 8; i = i + 8)
loop:
  infd r27, r30, r21 // if not (i < 0xff8) => fin
  mloadq r1, r27 // r1 = entry i
  addd r27, r27, r29 // i = i + 8
  mloadq r2, r27 // r1 = entry i
  hammq r3, r2, r1, r28 // r3 = hamm
  // if (hamm <= 9)
  infd r31, r3, r20 // no error we go back to loop
  intq r11 // KO
  intq r0 // end
endloop:
  intq r10 // OK
  intq r0 // end
_end:
```

Outils développés

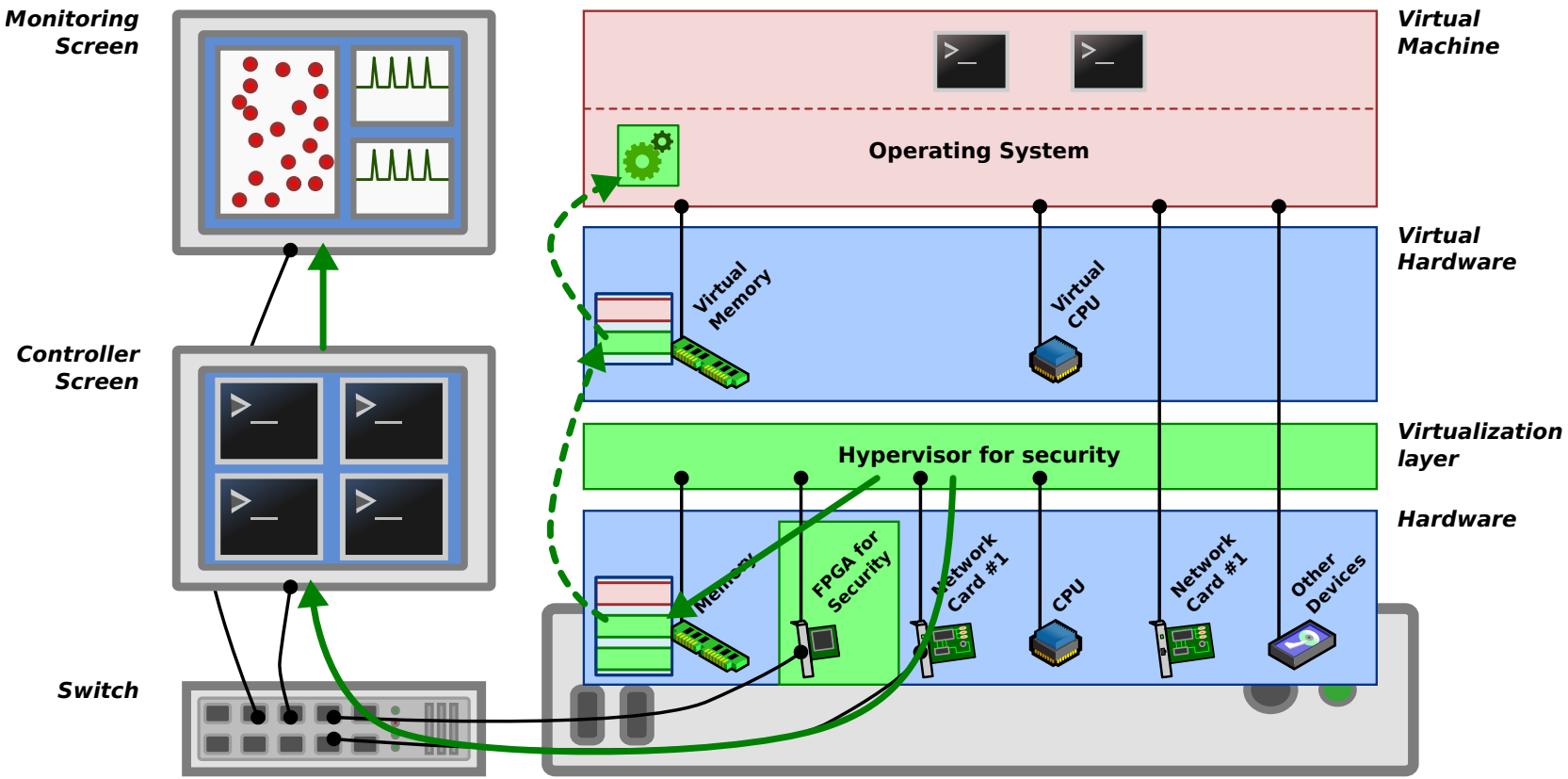
- Binutils pour la génération du challenge (gcc)
- Interface de contrôle et debug





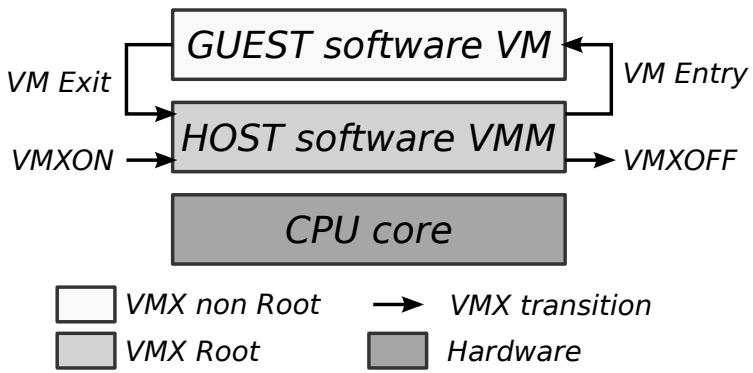
Plan

Protocole de tests : Étape 2



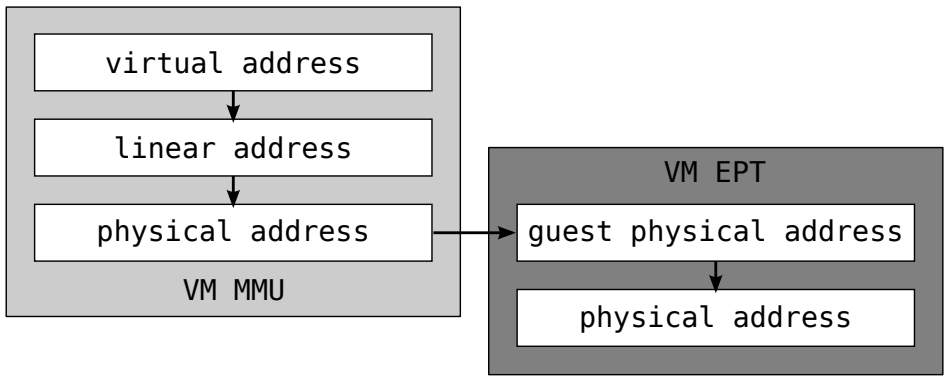
Assistance matérielle à la virtualisation : VT-x

- Virtual Machine Monitor : VMM
  - Host
  - Privilégié
  - Contrôle
- Virtual Machine : VM
  - Guest
  - Contrôlé par le VMM



- Virtual Machine Control Structure, VMCS
  - Définit et contrôle une machine virtuelle
  - Créée par le VMM
  - États : Host, Guest
  - Contrôles d'exécution pour le déroutement de l'exécution du Guest

Virtualisation de la MMU : EPT



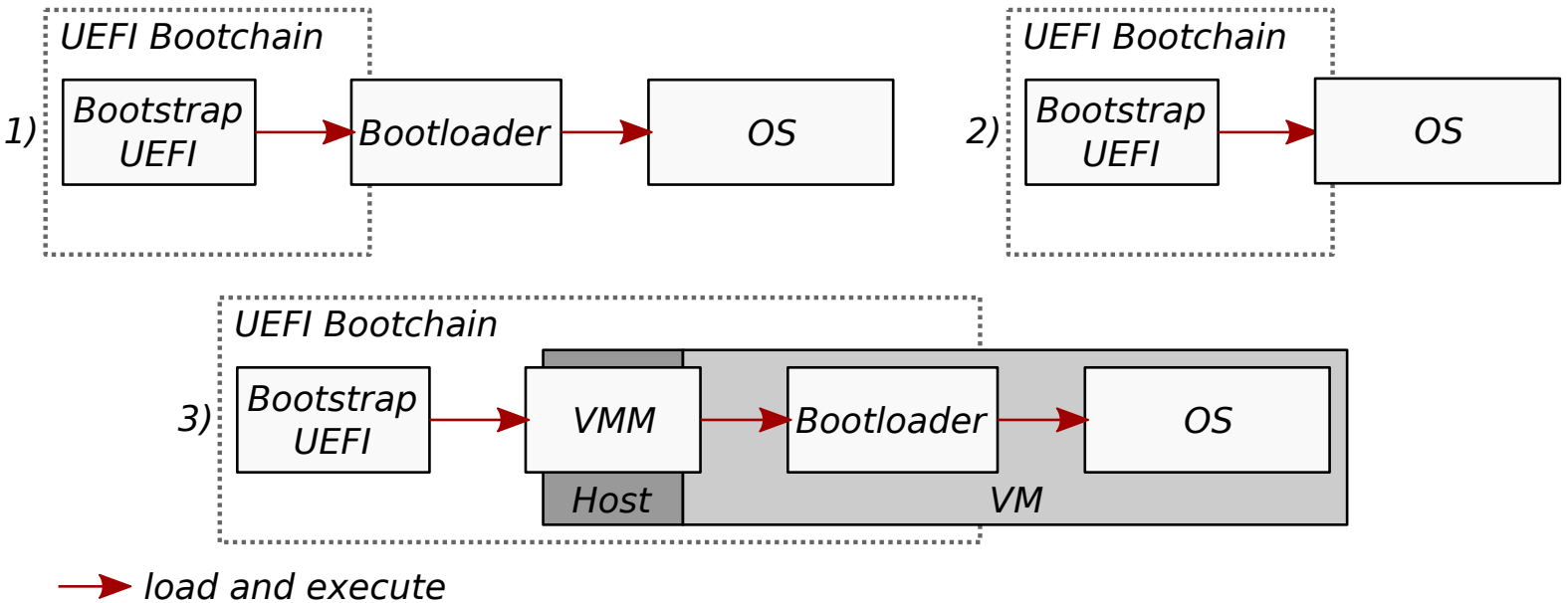
Définition

- Extended Page Table (EPT)
- Contrôle transparent supplémentaire

Unified Extensible Firmware Interface

- Interface standardisée entre BIOS et OS
- Extensible : modèle de drivers
- (Boot)loader intégré
  - Image PE32+
  - Applications (Shell)
  - Bootservice drivers
  - Runtime drivers
- Stockage de variables en flash (boot order)

Séquence de démarrage UEFI



Protection de l'hyperviseur de sécurité

- Configuration d'EPT
  - Protection de l'espace mémoire
  - Protection de l'espace de configuration PCI Express de la carte réseau
  - Protection des registres de configuration PCI Express de la carte réseau (BARs)
- Virtualisation des port d'entrée / sortie d'accès à l'espace de configuration PCI Express (0xcf8, 0xcfc)
- Virtualisation de l'accès à certains registres (MSRs, CRs, XCRs)

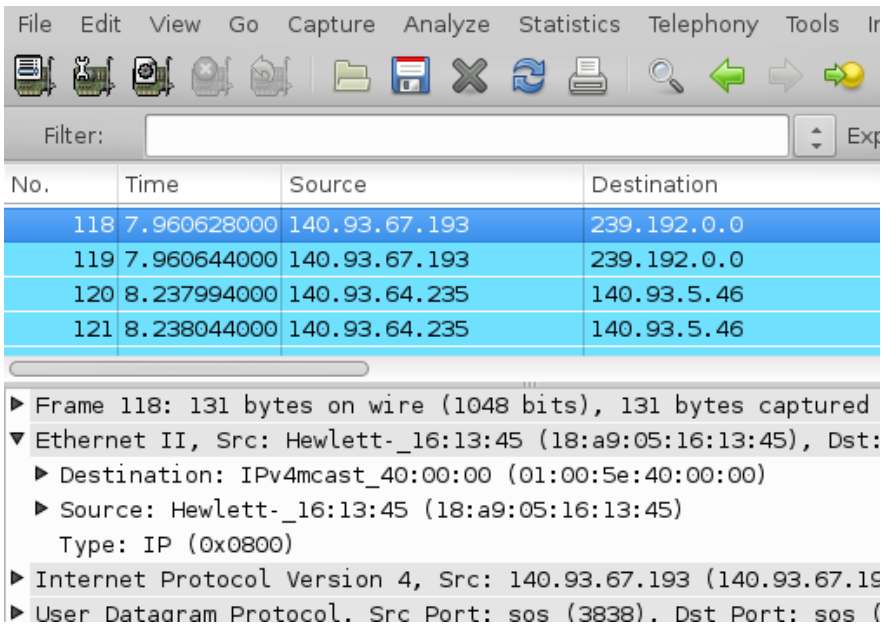
Enclave d'exécution

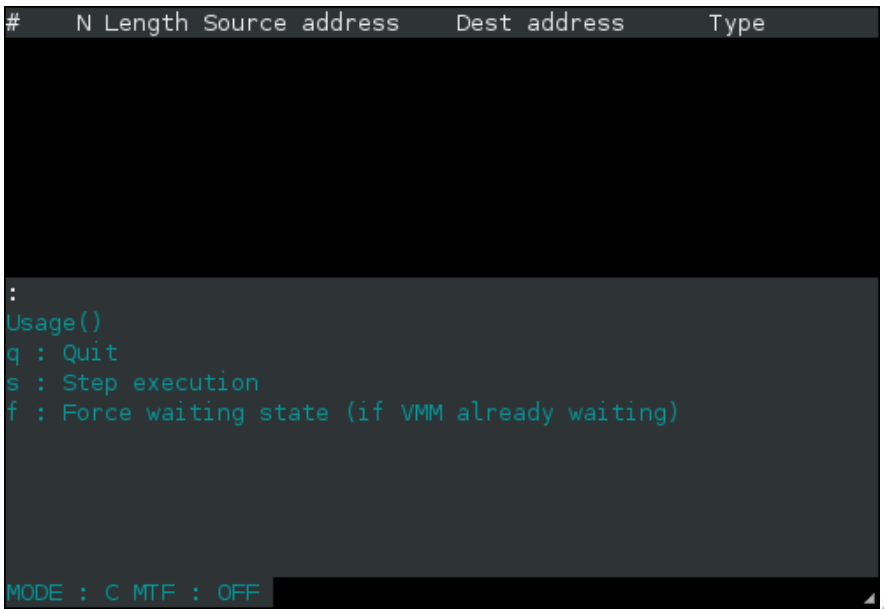
- Configuration du VMX preemption timer
- Chargement du code des tests d'intégrité des Guarded Software Components
  - Code embarqué
  - Driver UEFI

1. Interruption par le VMX preemption timer
2. Exécution périodique du challenge
   - Code (Expansion ROM)
   - Réponse (BARs)
3. Exécution des tests d'intégrité des Guarded Software Components

Outils développés

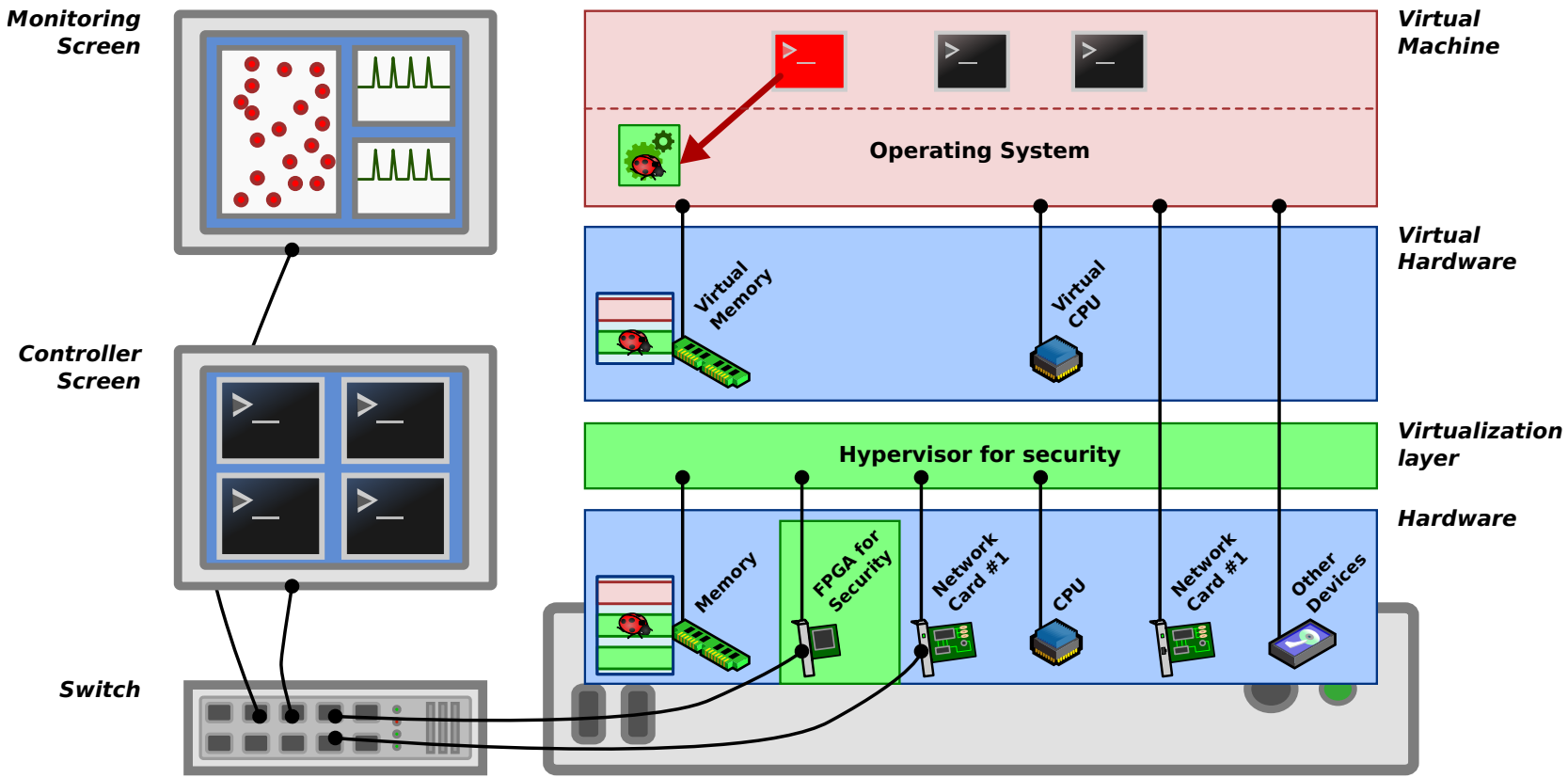
- Driver Ethernet, périphérique de confiance, platform flash
- Contrôle et debug Client / Serveur
- Extension Wireshark : dissecteur protocolaire
- API paravirtualisée d'interaction avec l'enclave (debug)



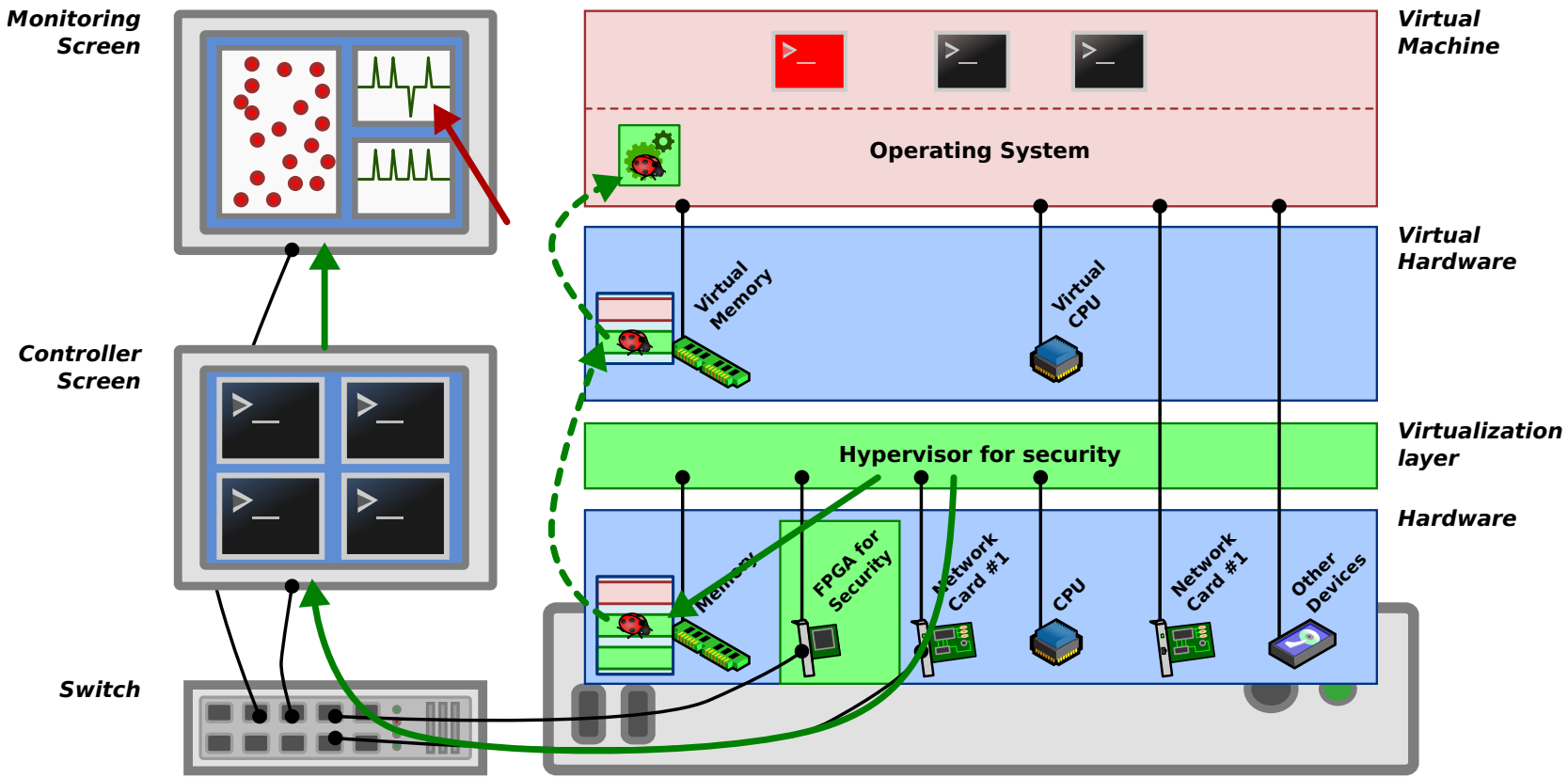


Plan

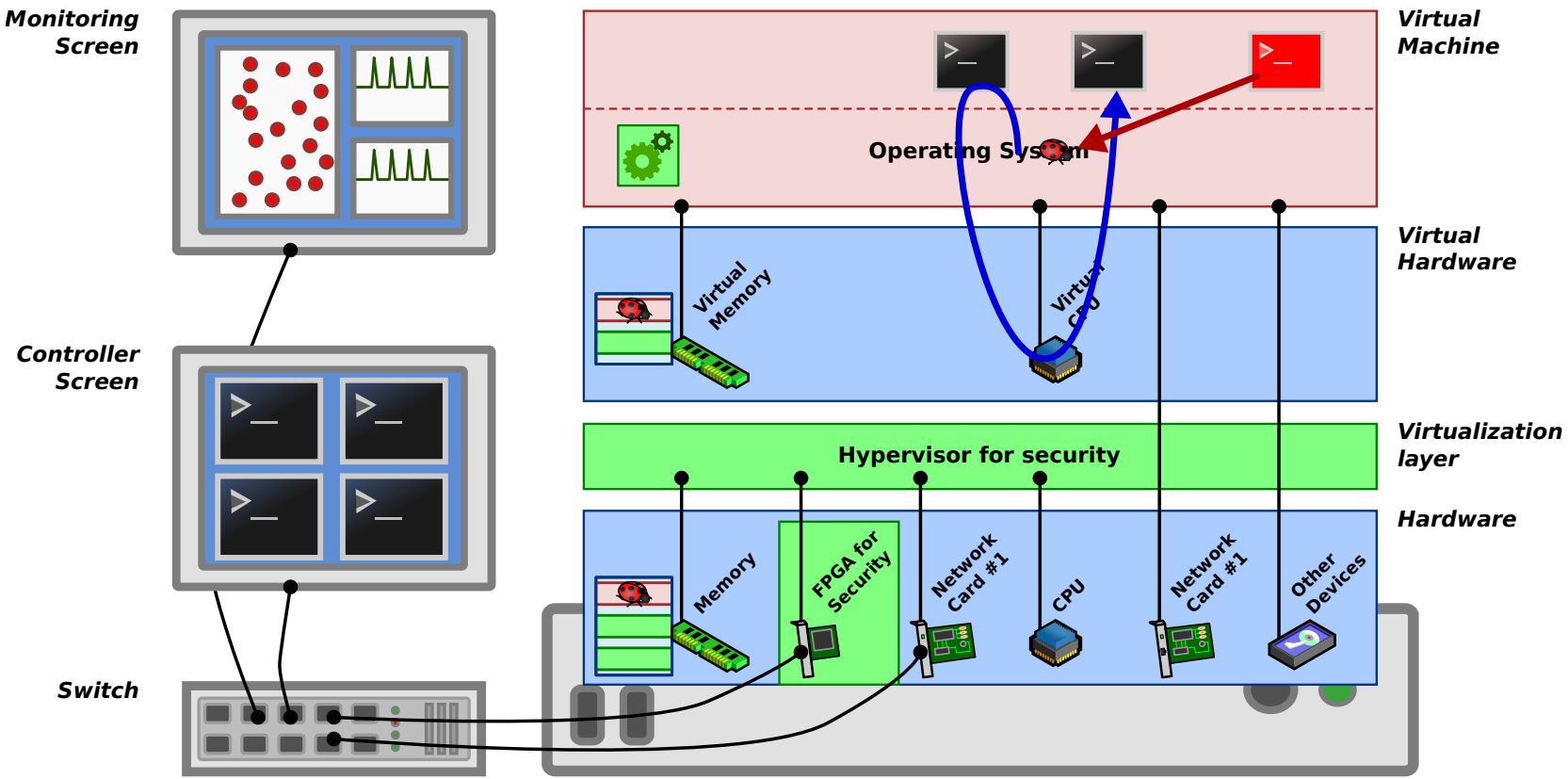
Corruption d'un driver e1000e



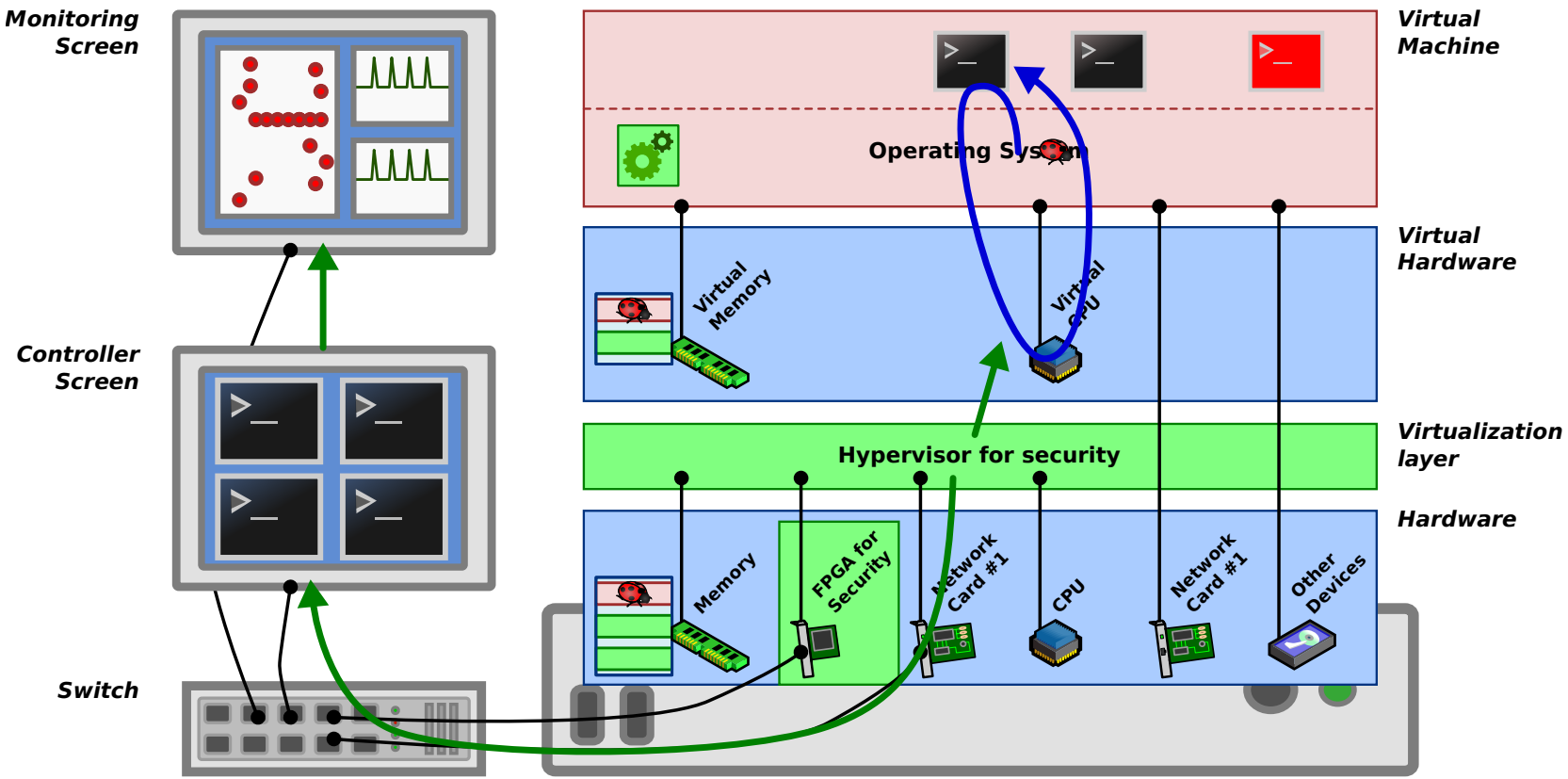
Corruption d'un driver e1000e



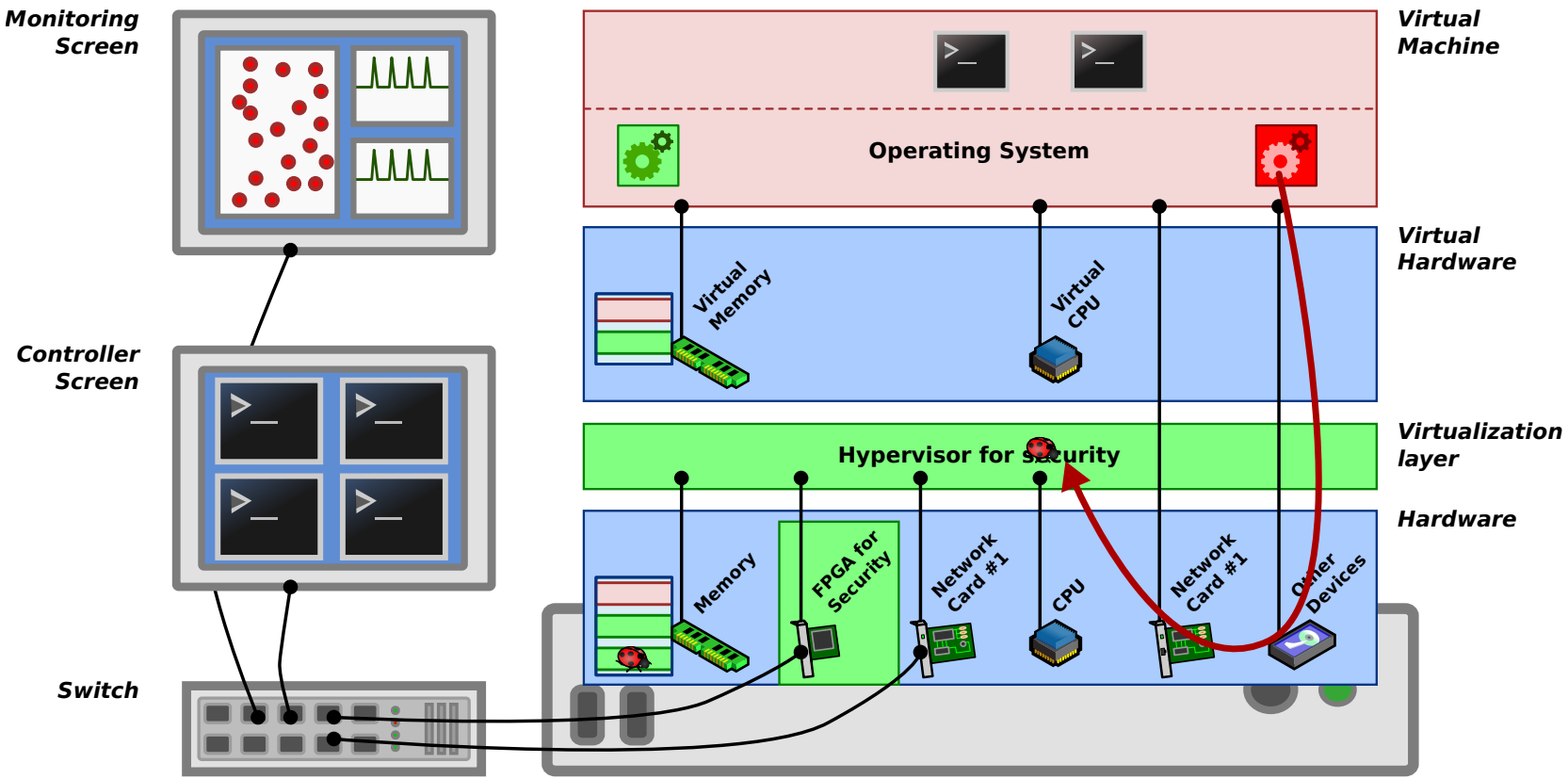
Corruption de l'ordonnancement



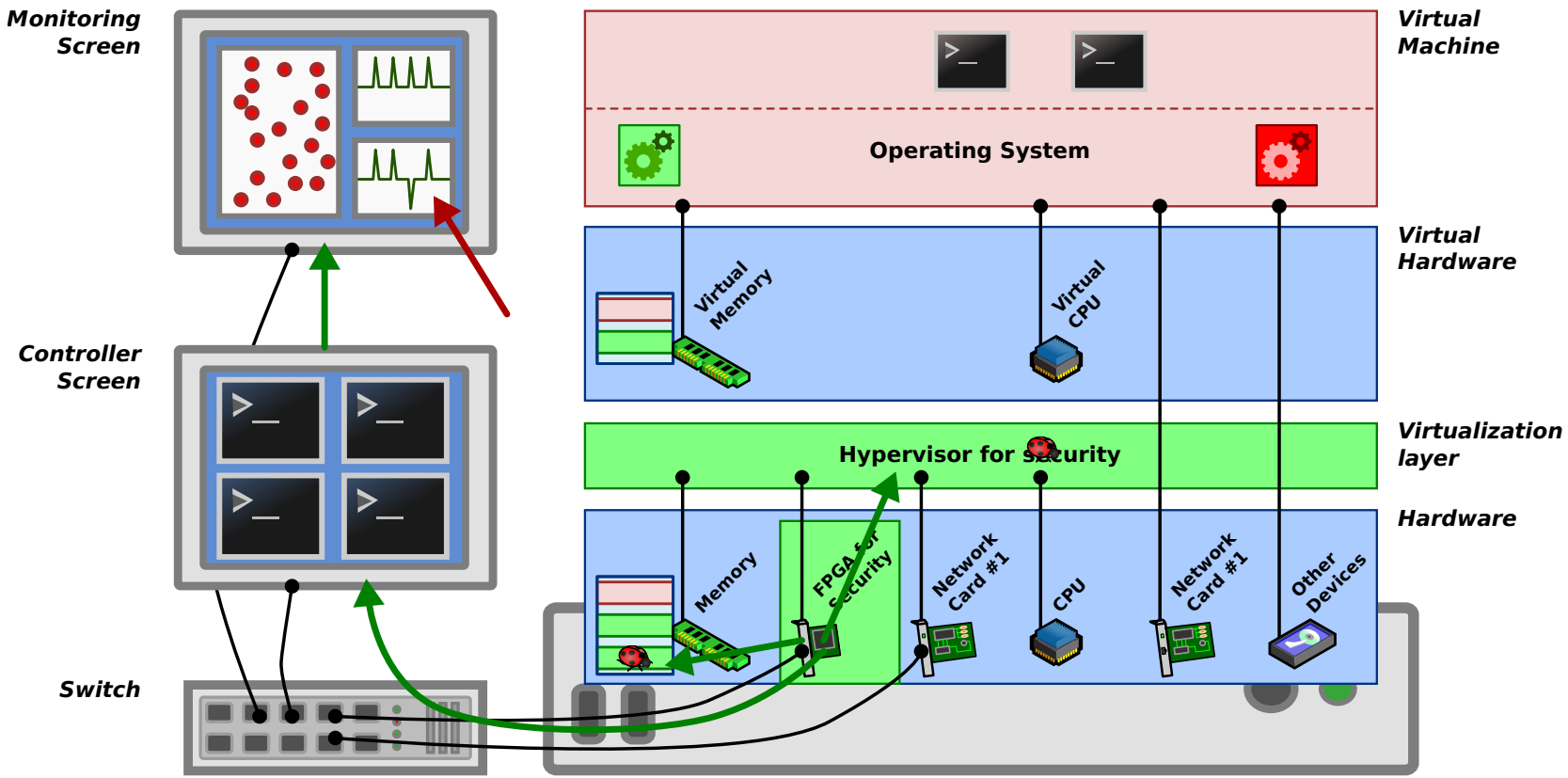
Corruption de l'ordonnancement



Corruption de l'hyperviseur de sécurité



Corruption de l'hyperviseur de sécurité



Travail en cours...

Développement de moyens matériels pour la protection des couches basses du logiciel

Benoît Morgan, Éric Alata, Vincent Nicomette

Réunion du groupe RéSIST
LAAS-CNRS, INSA Toulouse

15 décembre 2014