Architecture de sécurité dynamique pour systèmes multiprocesseurs intégrés sur puce

Joël Porquet

Présenté par Franck Wajsbürt

Université Pierre et Marie Curie - Paris VI

30 mai 2012

UPMC
SORBONNE UNIVERSITÉS

Résultat de la thèse de Joël Porquet

Thèse CIFRE soutenue le 13 décembre 2010

Partenariat entre :

- Université Pierre et Marie Curie,
  - Laboratoire d'informatique de Paris 6,
    - Département « Systèmes sur Puce »
- STMicroelectronics,
  - Division « Advanced System Technology »,
    - Équipe « Security Roadmap »

Encadrement :

- Directeur de thèse : Prof. Alain Greiner
- Encadrant industriel : Christian Schwarz

Plan

Plan

Systèmes embarqués orientés multimédia

Set-top box (STB)

Smartphone

Double enjeu

- Sécurité
- Flexibilité

Systèmes embarqués orientés multimédia

Sous-systèmes logiciels autonomes

- Interface utilisateur
- Sous-systèmes sécurisés :
  - *STB* : Accès Conditionnel (*CA*), Magnétoscope, etc.
  - *Smartphone* : Accès réseau (*Baseband*), etc.
- Piles logicielles hétérogènes :
  - Applications utilisateur,
  - Systèmes d'exploitations spécialisés (*RT-OS*),
  - Systèmes d'exploitations généralistes (*GP-OS*),
  - etc.

Systèmes embarqués orientés multimédia

Sous-systèmes logiciels autonomes

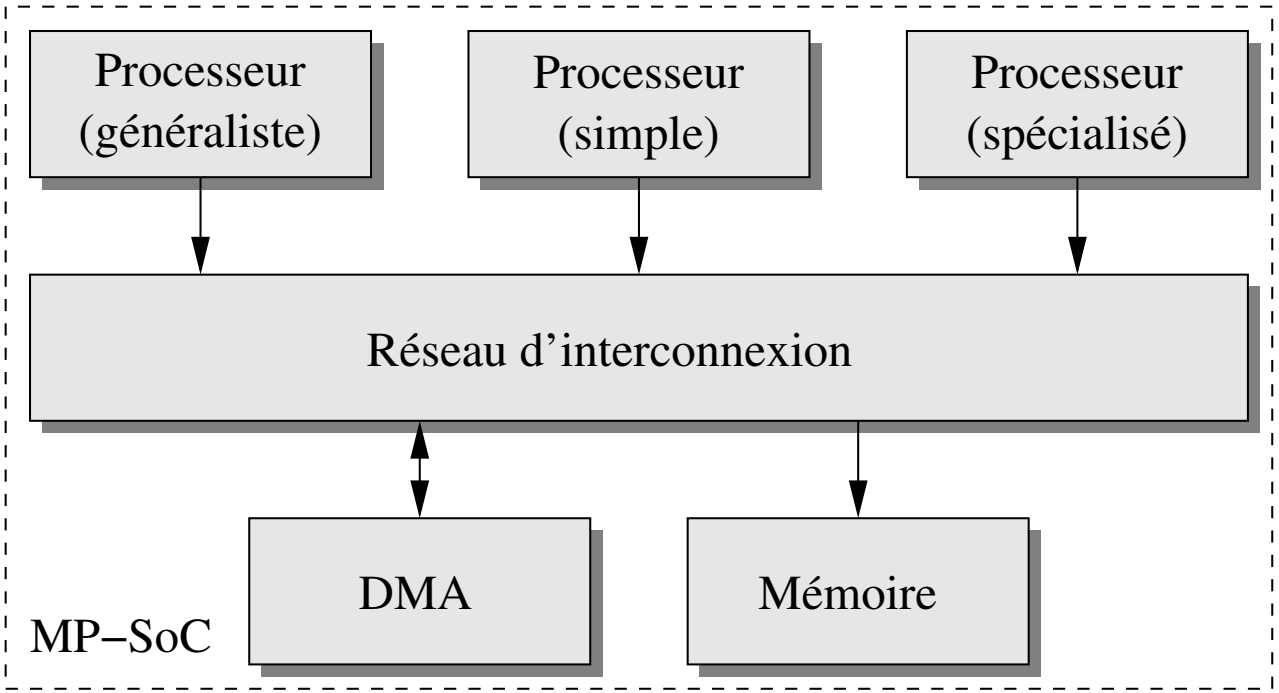
- Interface utilisateur
- Sous-systèmes sécurisés :
  - *STB* : Accès Conditionnel (*CA*), Magnétoscope, etc.
  - *Smartphone* : Accès réseau (*Baseband*), etc.
- Piles logicielles hétérogènes :
  - Applications utilisateur,
  - Systèmes d'exploitations spécialisés (*RT-OS*),
  - Systèmes d'exploitations généralistes (*GP-OS*),
  - etc.

Plateforme matérielle complexe

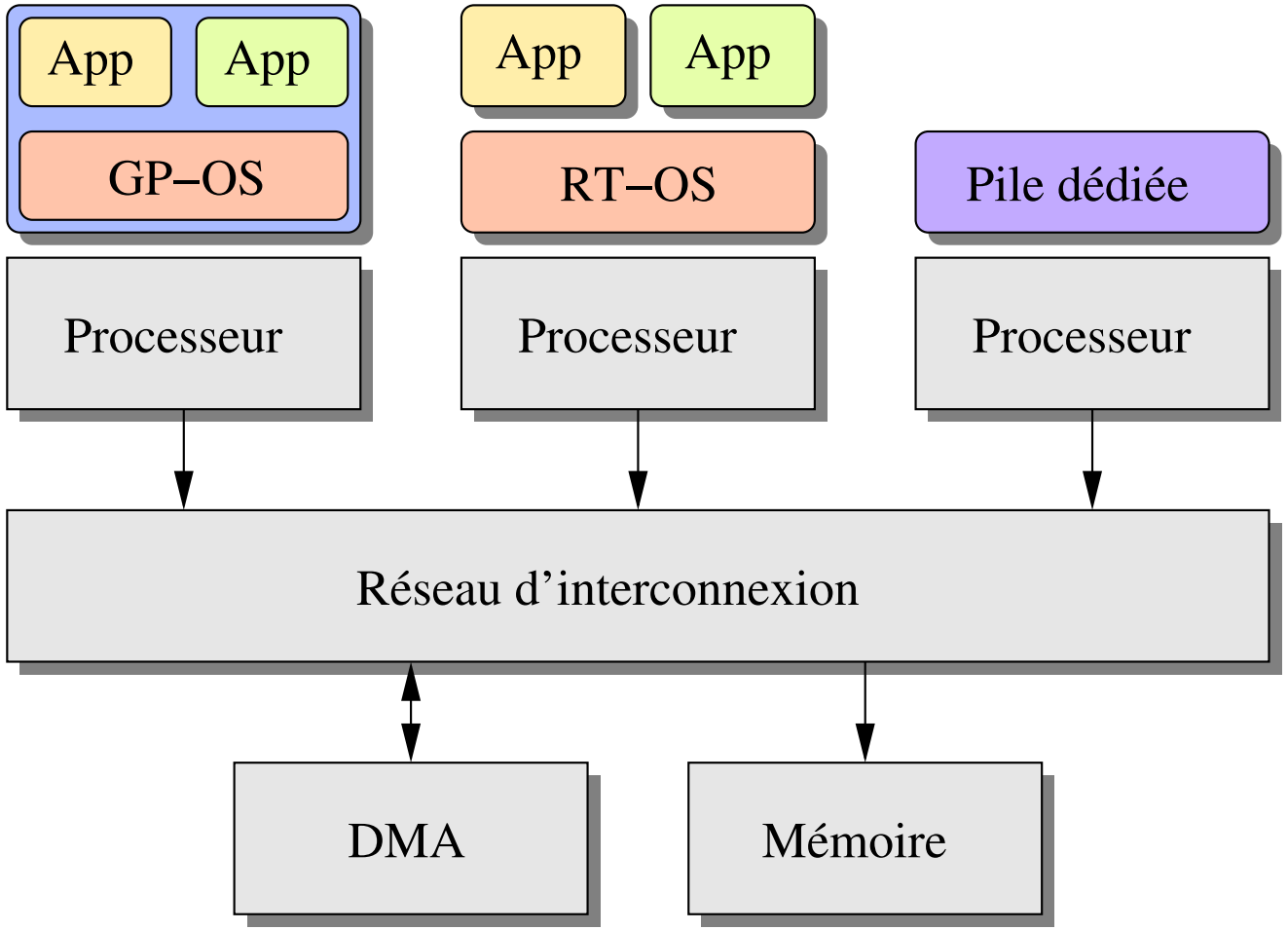
- Système multiprocesseur intégré sur puce (*MP-SoC*)
- Multiprocesseur hétérogène
- Espace d'adressage partagé

Plateforme matérielle de référence

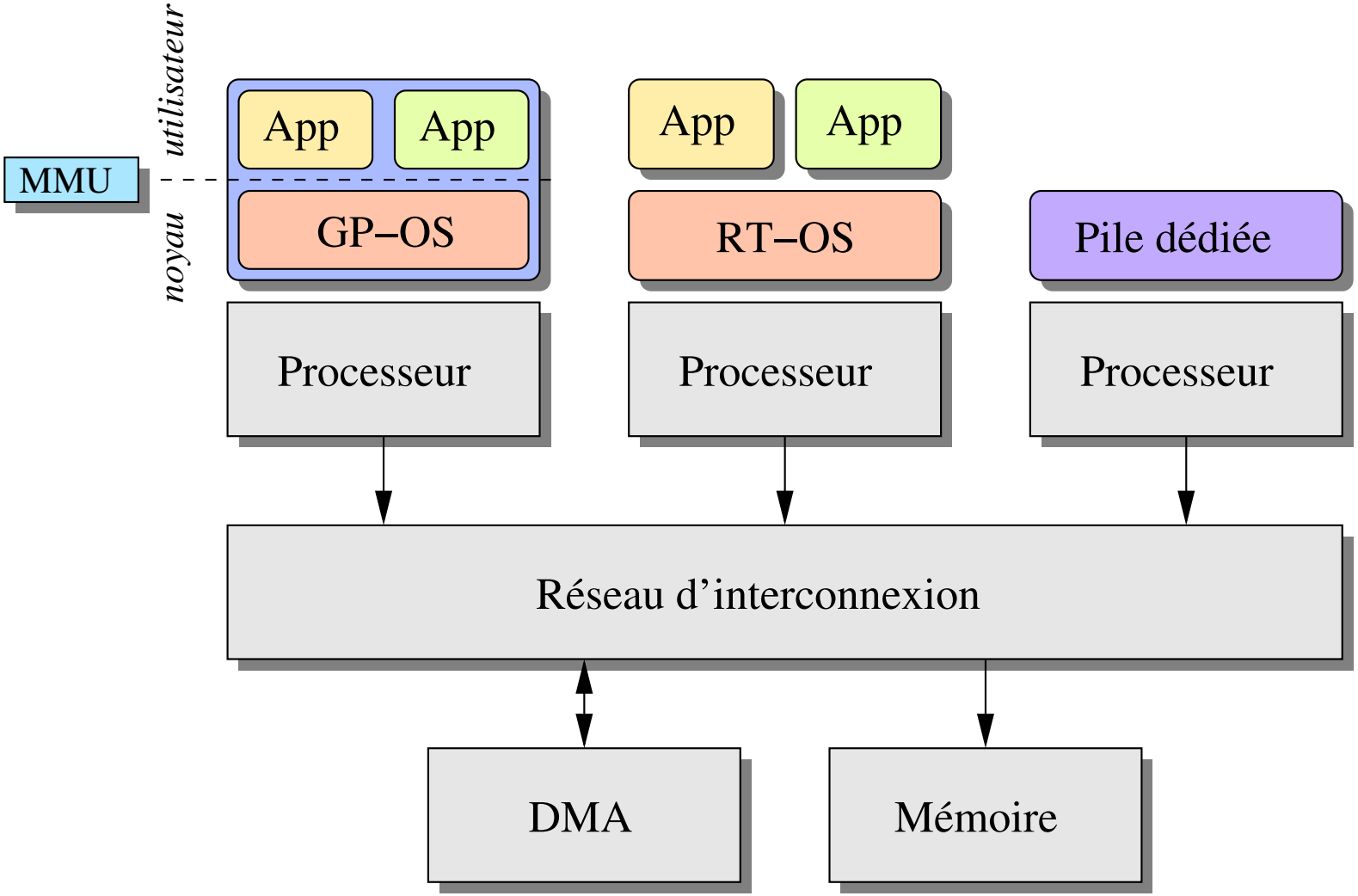
- Multiprocesseur hétérogène
- Périphérique(s) à capacité DMA (*Direct Memory Access*)
- Espace d'adressage partagé



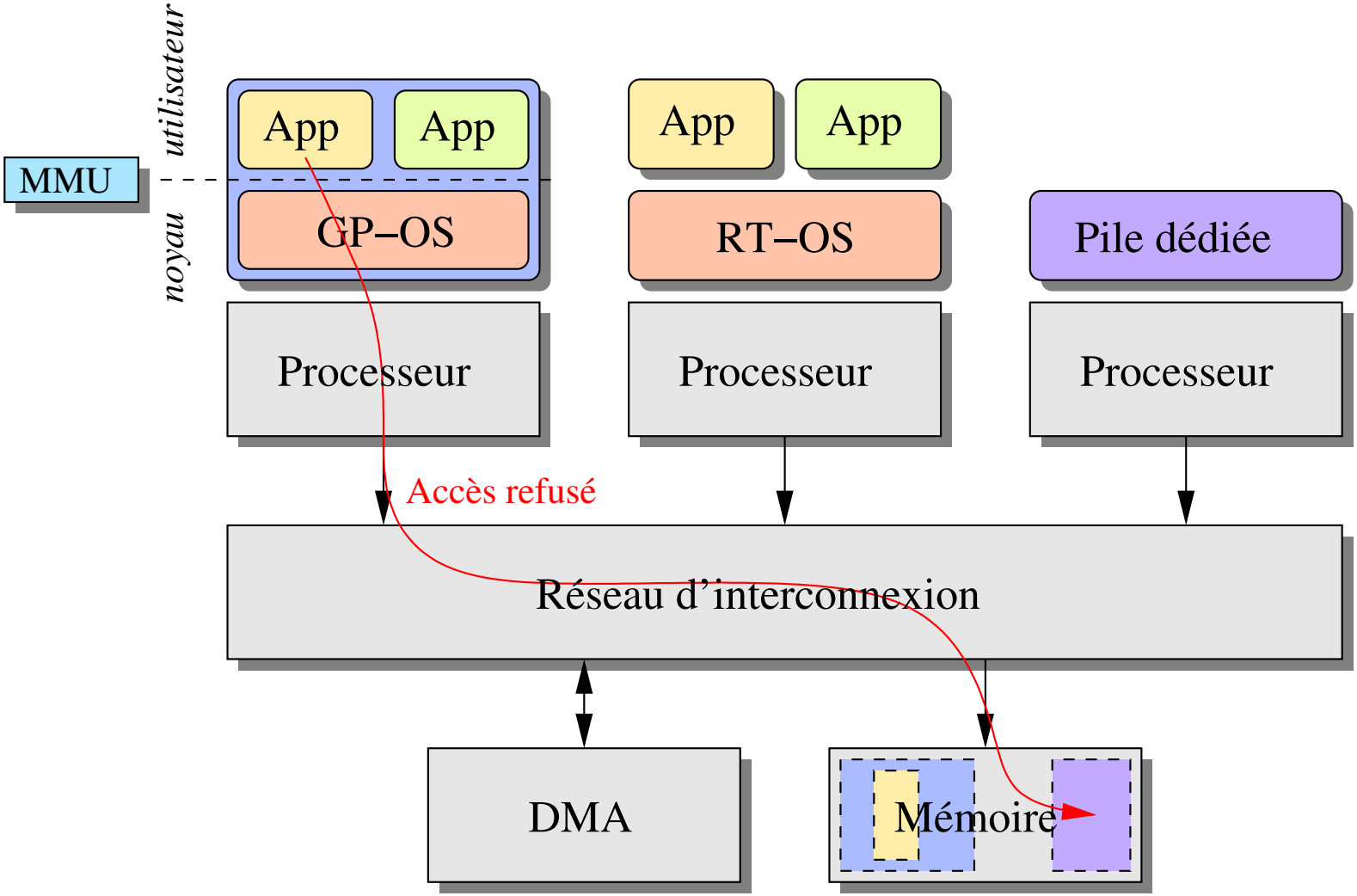
Co-hébergement



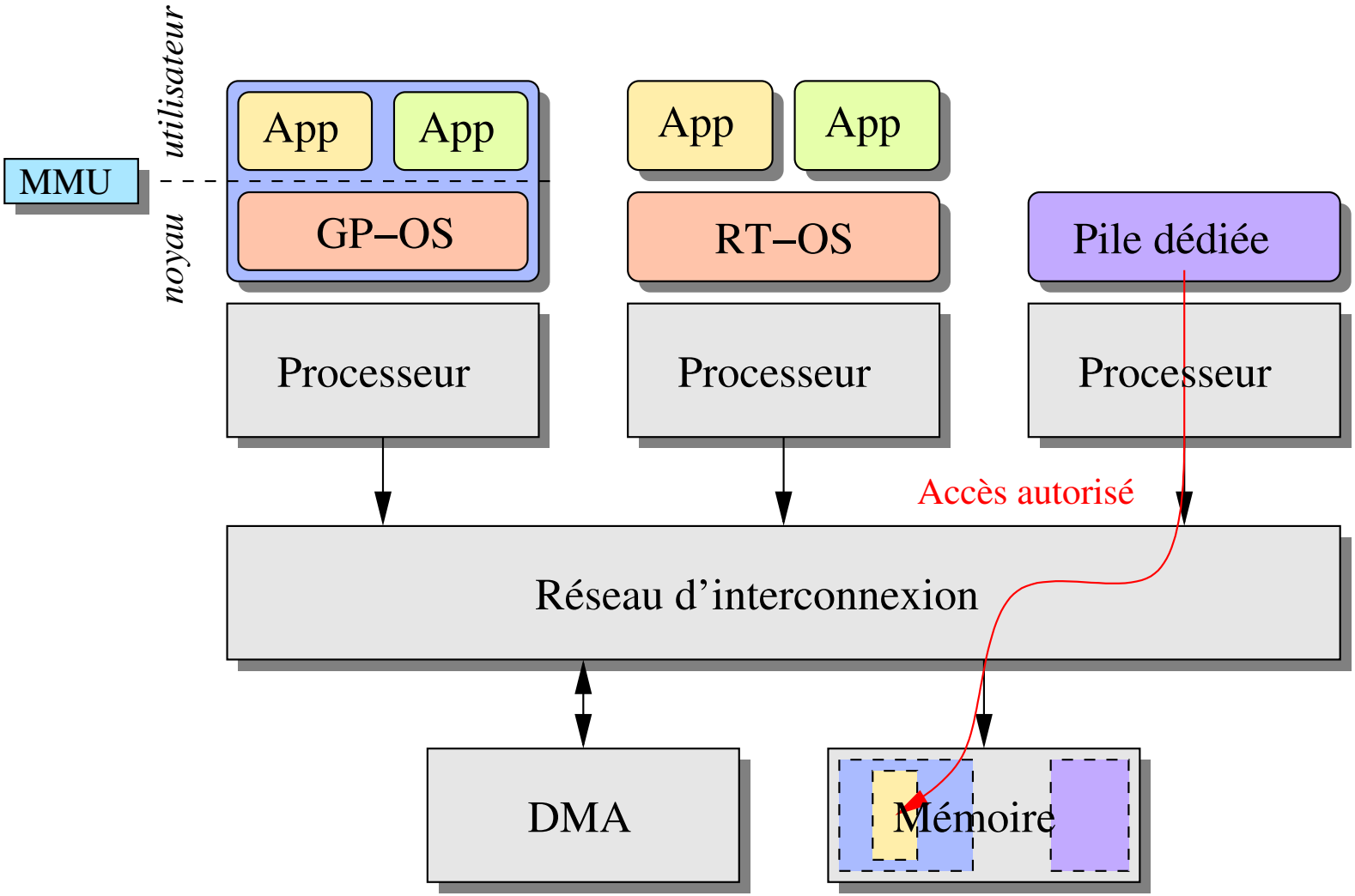
Co-hébergement : isolation par mémoire virtuelle



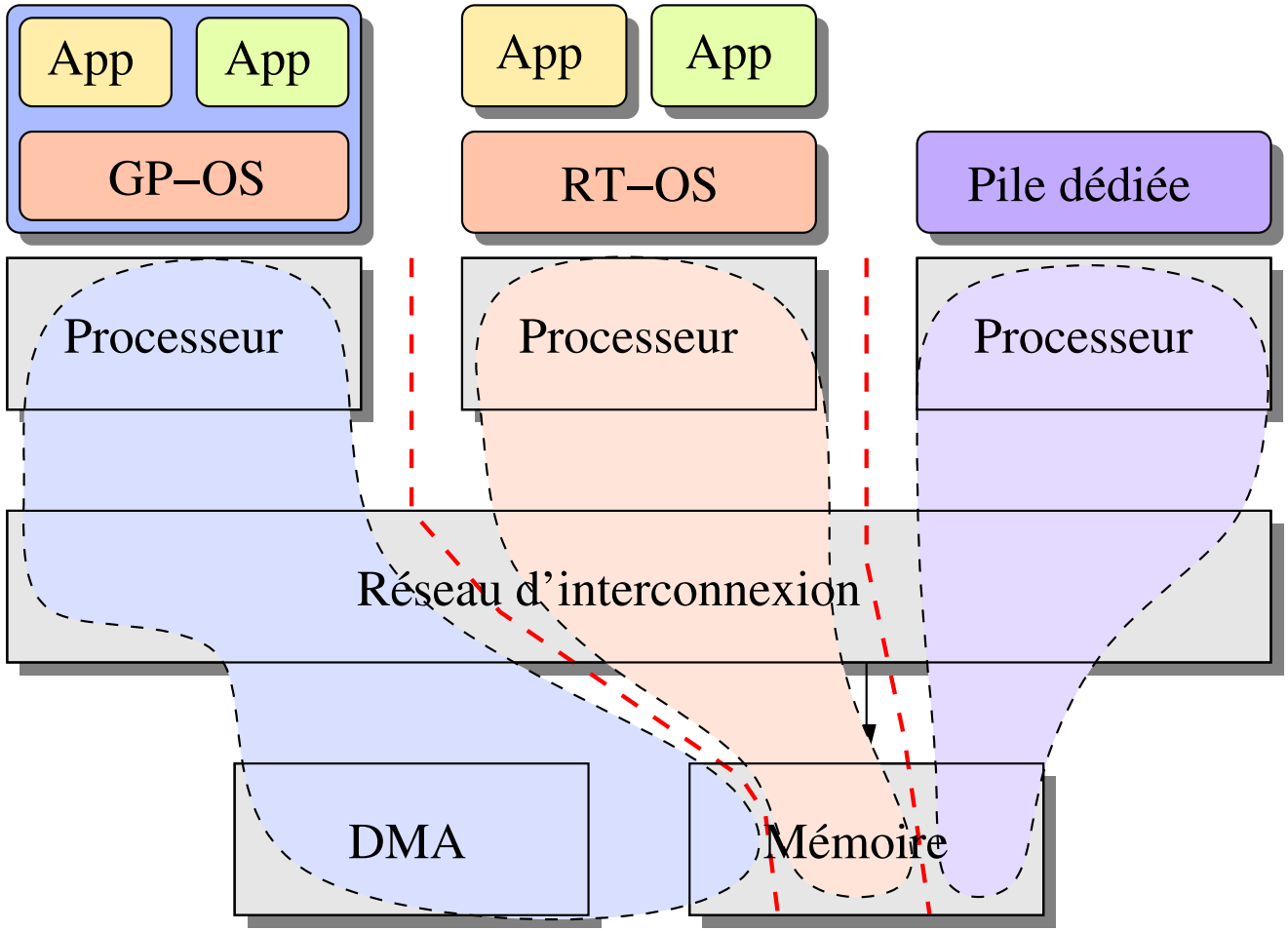
Co-hébergement : isolation par mémoire virtuelle



Co-hébergement : isolation par mémoire virtuelle



Co-hébergement : isolation par partitionnement statique



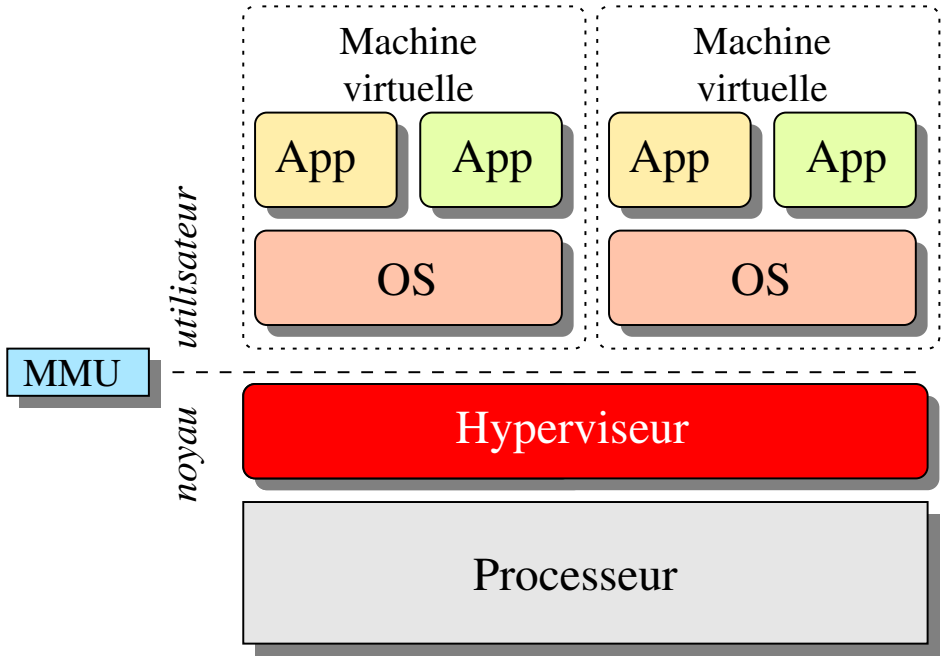
Problématique

Co-hébergement sécurisé, flexible et dynamique

- Cohérent avec les aspects MP-SoC
- Partage de l'espace d'adressage, des périphériques
- Estimation des coûts

Plan

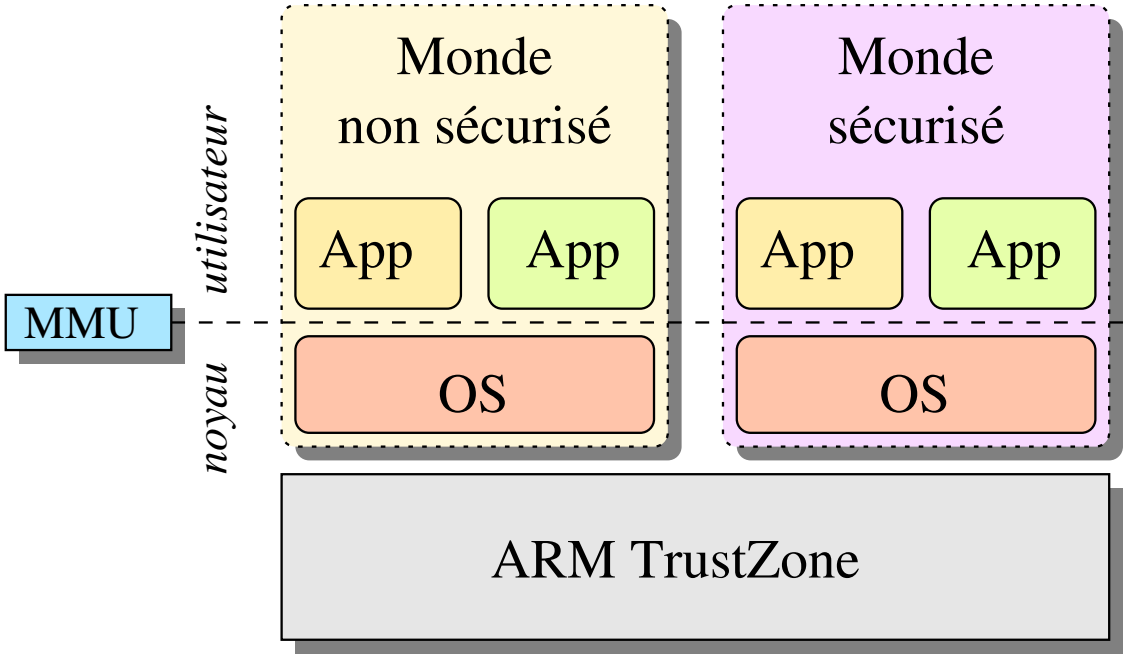
Virtualisation



Exécution des piles logicielles autonomes dans des environnements virtuels

- Dans l'embarqué : virtualisation logicielle
- Efficace pour un co-hébergement sur une base commune
- Pas adapté pour les plateformes hétérogènes

ARM TrustZone



Virtualisation « binaire »

- Ne supporte qu'une seule pile logicielle autonome

Architectures sécurisées

Définition

Protection de données intégrée au réseau d'interconnexion

Réseau sur puce (NoC)

Architectures sécurisées

Définition

Protection de données intégrée au réseau d'interconnexion

Réseau sur puce (NoC)

2006 : μSpider, UBS

- Granularité au niveau composant physique (initiateur et cible)

Architectures sécurisées

Définition

Protection de données intégrée au réseau d'interconnexion

Réseau sur puce (NoC)

2006 : μSpider, UBS

- Granularité au niveau composant physique (initiateur et cible)

2007 : DPU (*Data Protection Unit*), Polimi/STM

- Segmentation mémoire
- Granularité au niveau composant physique (initiateur)
- Avec quelques critères additionnels pour les processeurs (par exemple, distinction des modes d'exécution)

Architectures sécurisées

Définition

Protection de données intégrée au réseau d'interconnexion

Réseau sur puce (NoC)

2006 : μSpider, UBS

- Granularité au niveau composant physique (initiateur et cible)

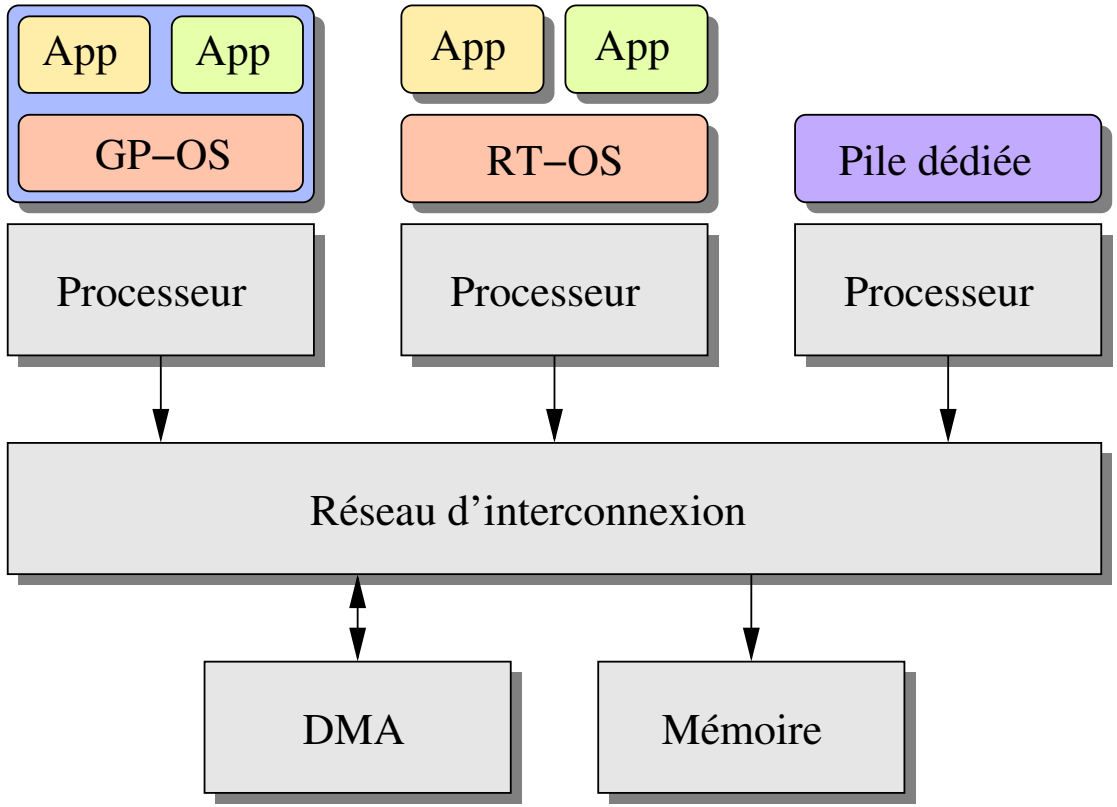
2007 : DPU (*Data Protection Unit*), Polimi/STM

- Segmentation mémoire
- Granularité au niveau composant physique (initiateur)
- Avec quelques critères additionnels pour les processeurs (par exemple, distinction des modes d'exécution)

Manque de flexibilité : partage des ressources processeur et mémoire

Plan

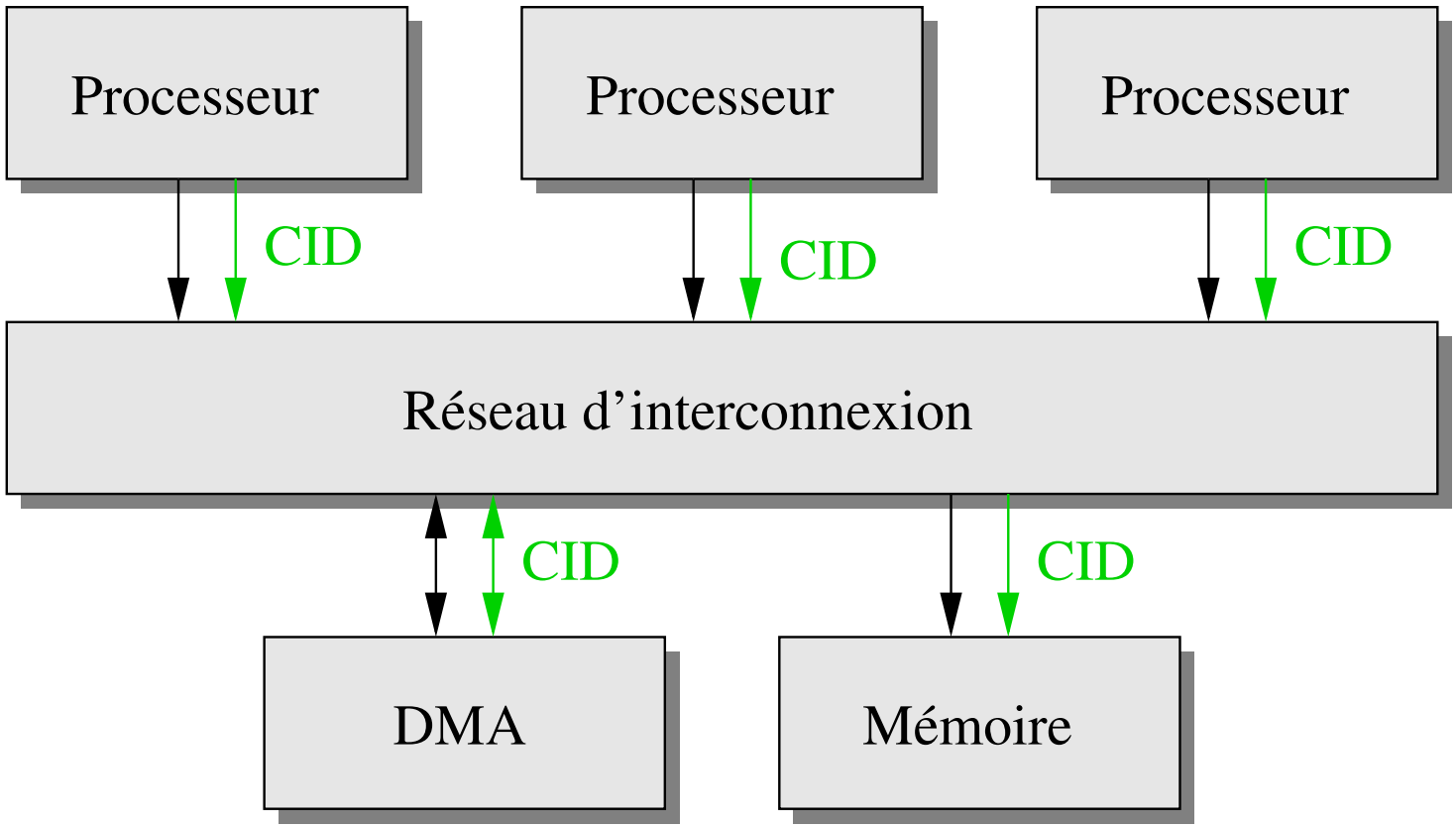
Introduction



Co-hébergement sécurisé

- Compartiment : conteneur sécurisé d'une entité logique
- Principes : identification et protection

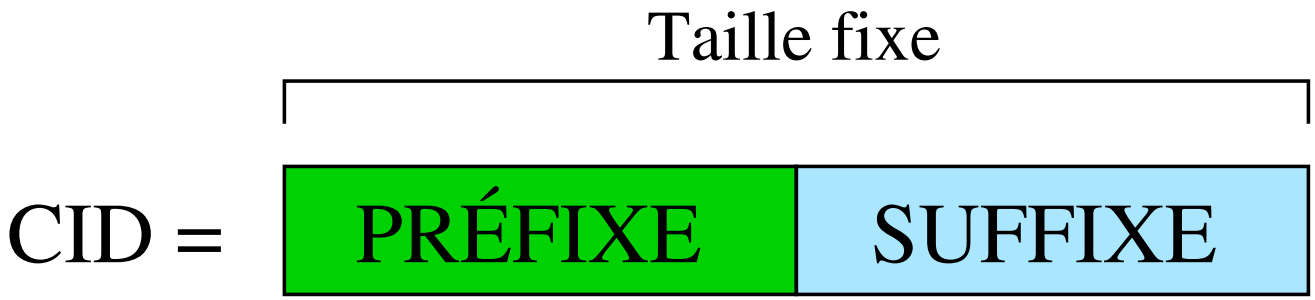
Identification - Propagation dans le réseau



Nouveau signal dans le protocole transactionnel

CID : *Compartment IDentifier*

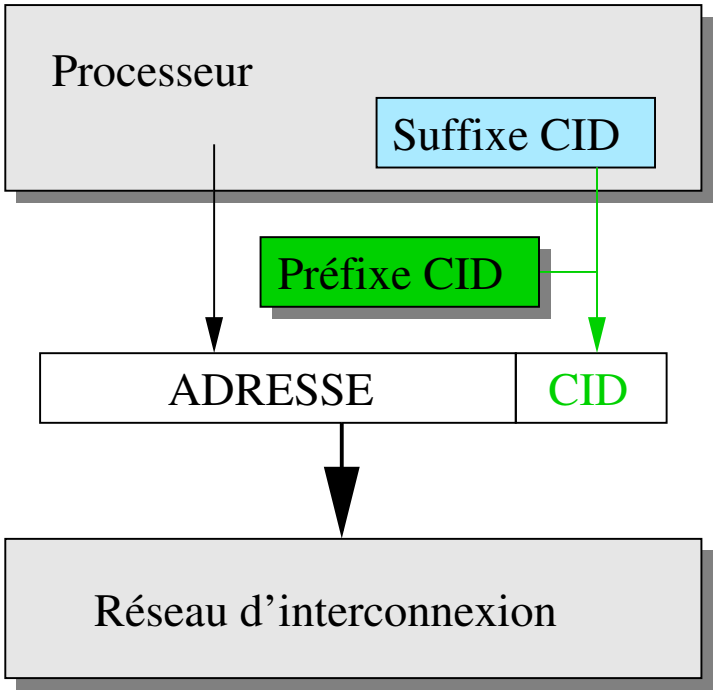
Identification - Processeurs



Structuration hiérarchique du CID

- Préfixe : défini par un agent de confiance global
  $\Rightarrow$ (GTA, *Global Trusted Agent*)
- Suffixe : défini par des agents de confiance locaux
  $\Rightarrow$ (LTA, *Local Trusted Agent*)

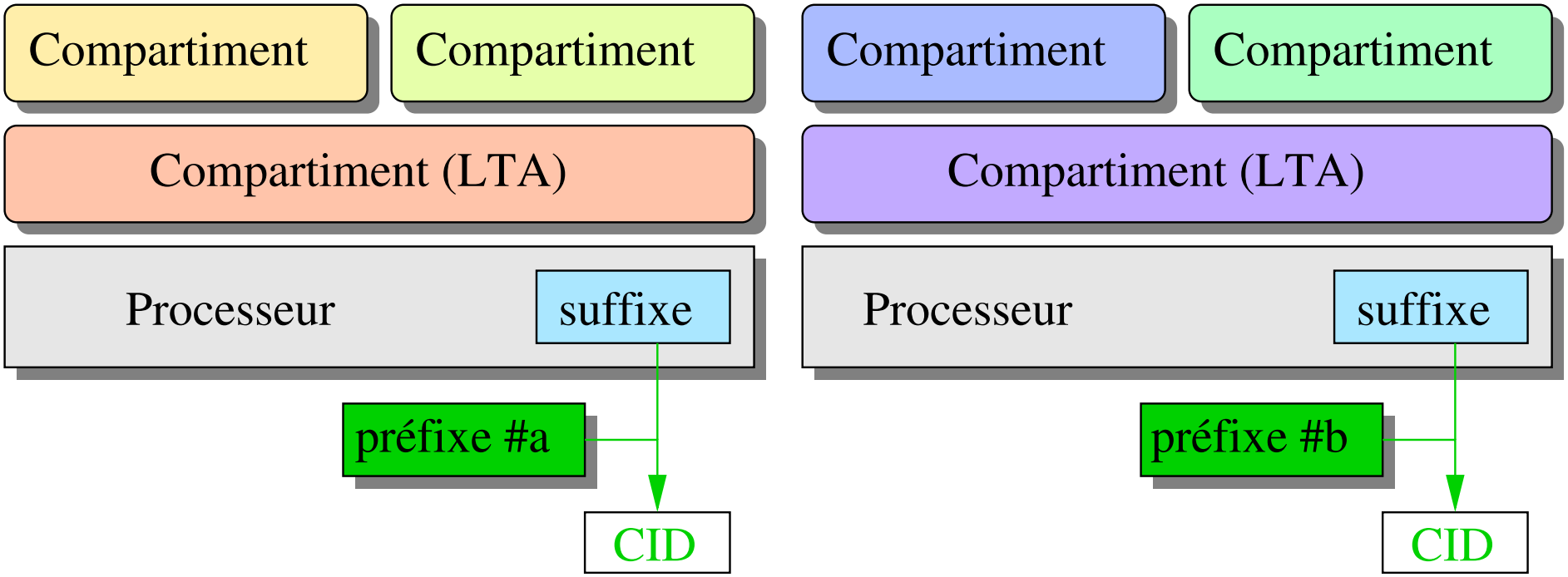
Identification - Processeur généraliste



Compatibilité matérielle au « multi-compartiment »

- Préfixe : registre externe
- Suffixe : nouveau registre de contrôle interne

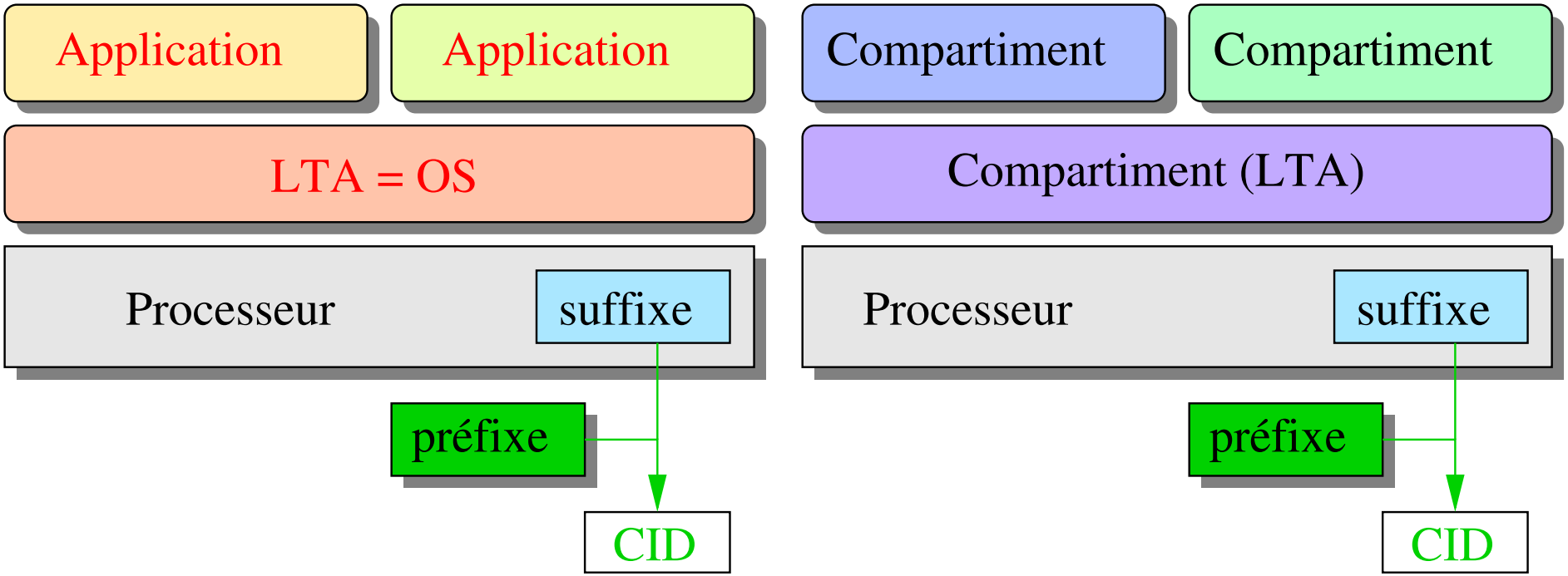
Identification - Processeur généraliste



Compatibilité logicielle au multi-compartiment

- Agent de confiance local (LTA)
- Partage du processeur, gestion du registre de suffixe du CID

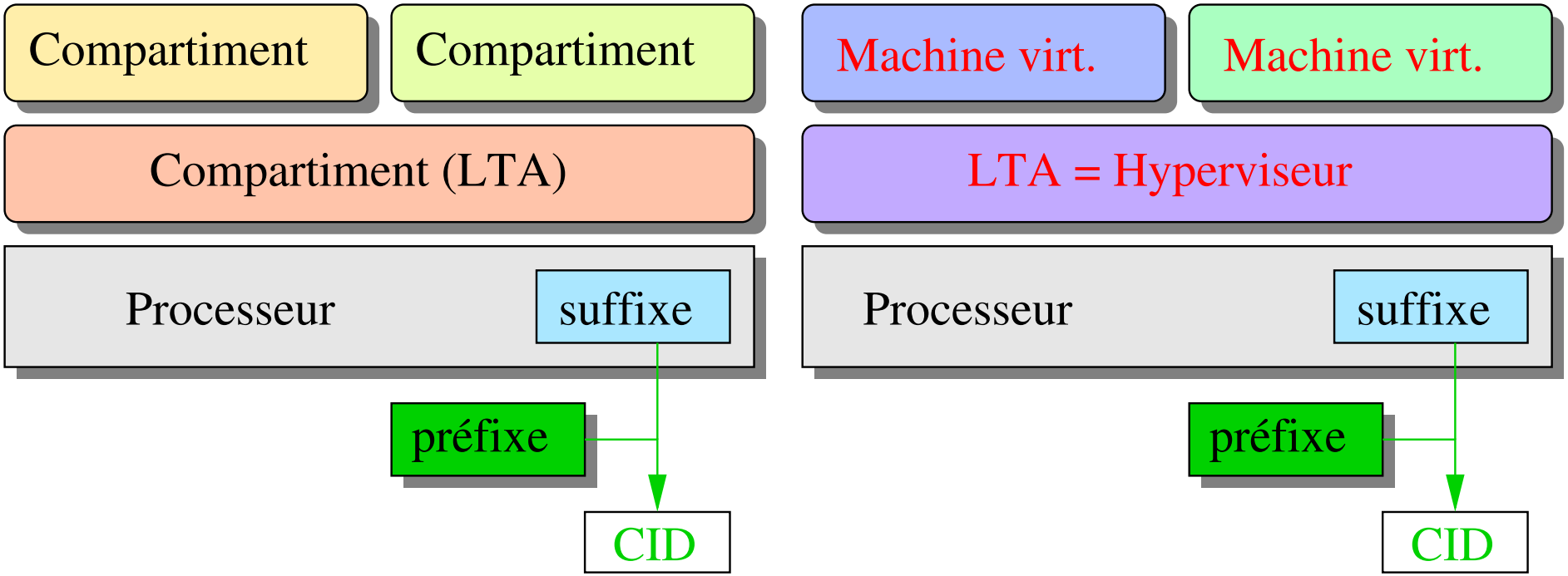
Identification - Processeur généraliste



Compatibilité

Système d'exploitation et applications

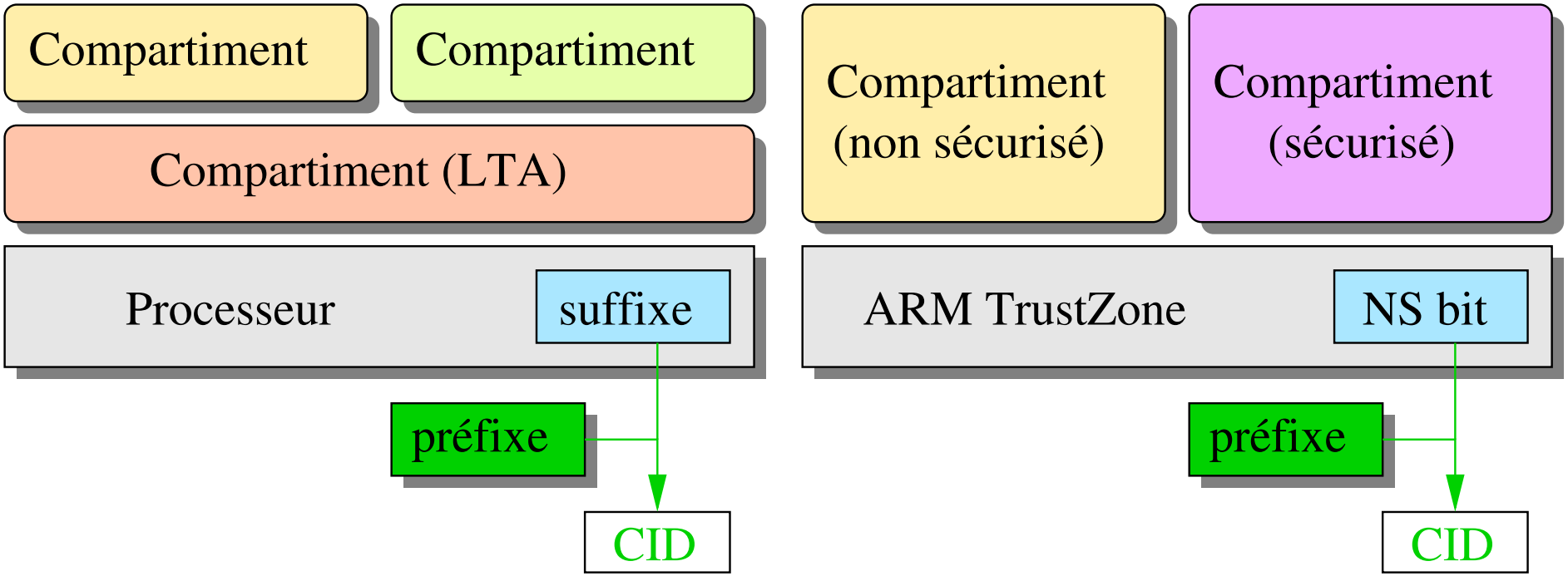
Identification - Processeur généraliste



Compatibilité

Hyperviseur et machines virtuelles

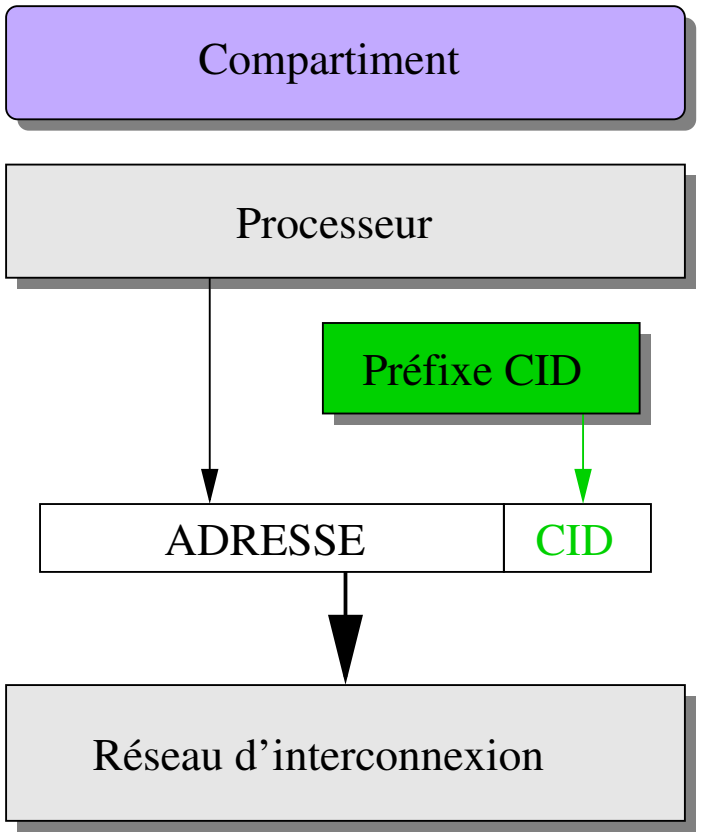
Identification - Processeur généraliste



Compatibilité ARM TrustZone

Cas particulier de processeur multi-compartiment

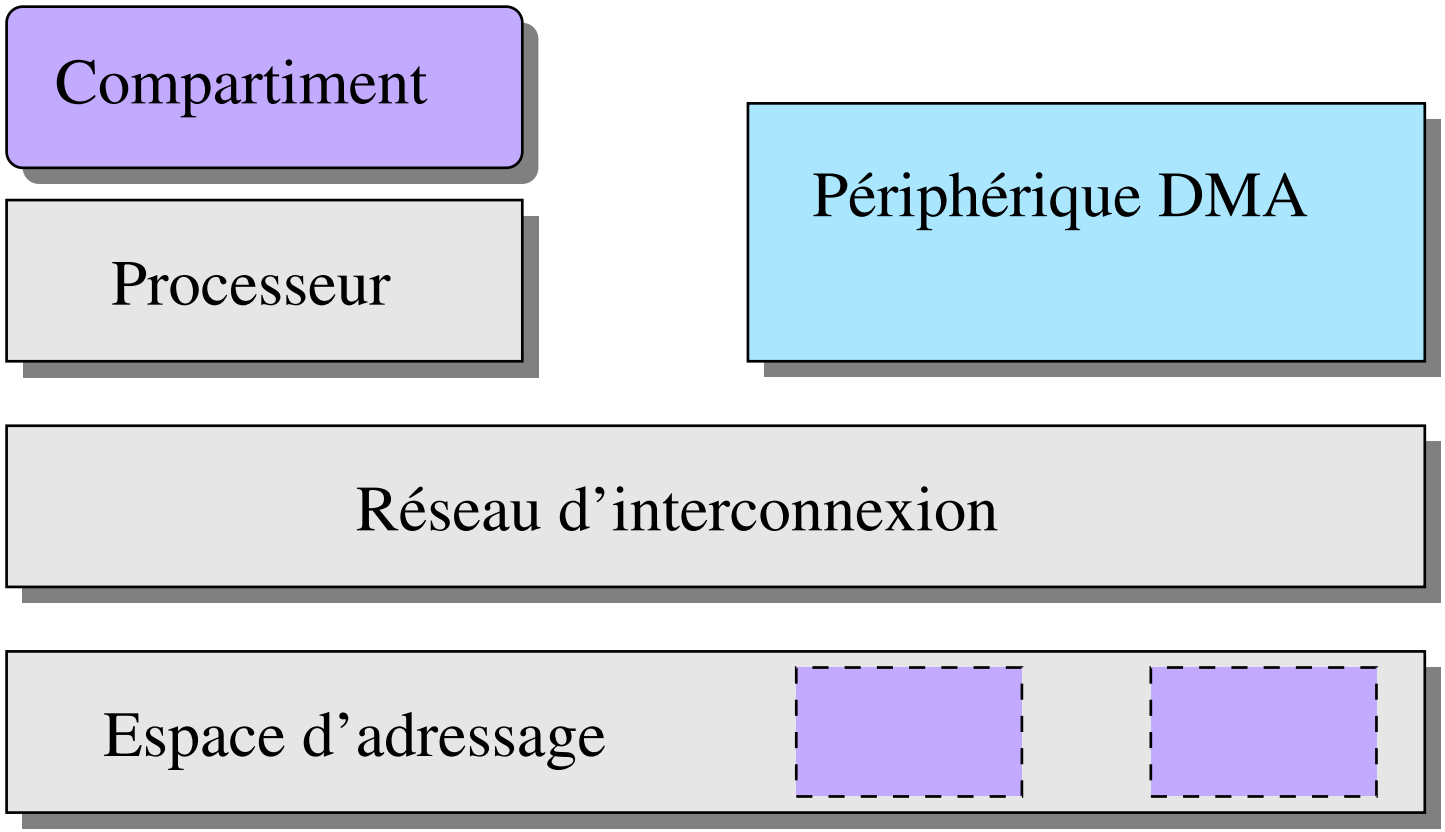
Identification - Processeur spécialisé



Compatibilité au multi-compartiment

- Uniquement un registre externe, configuré par le GTA

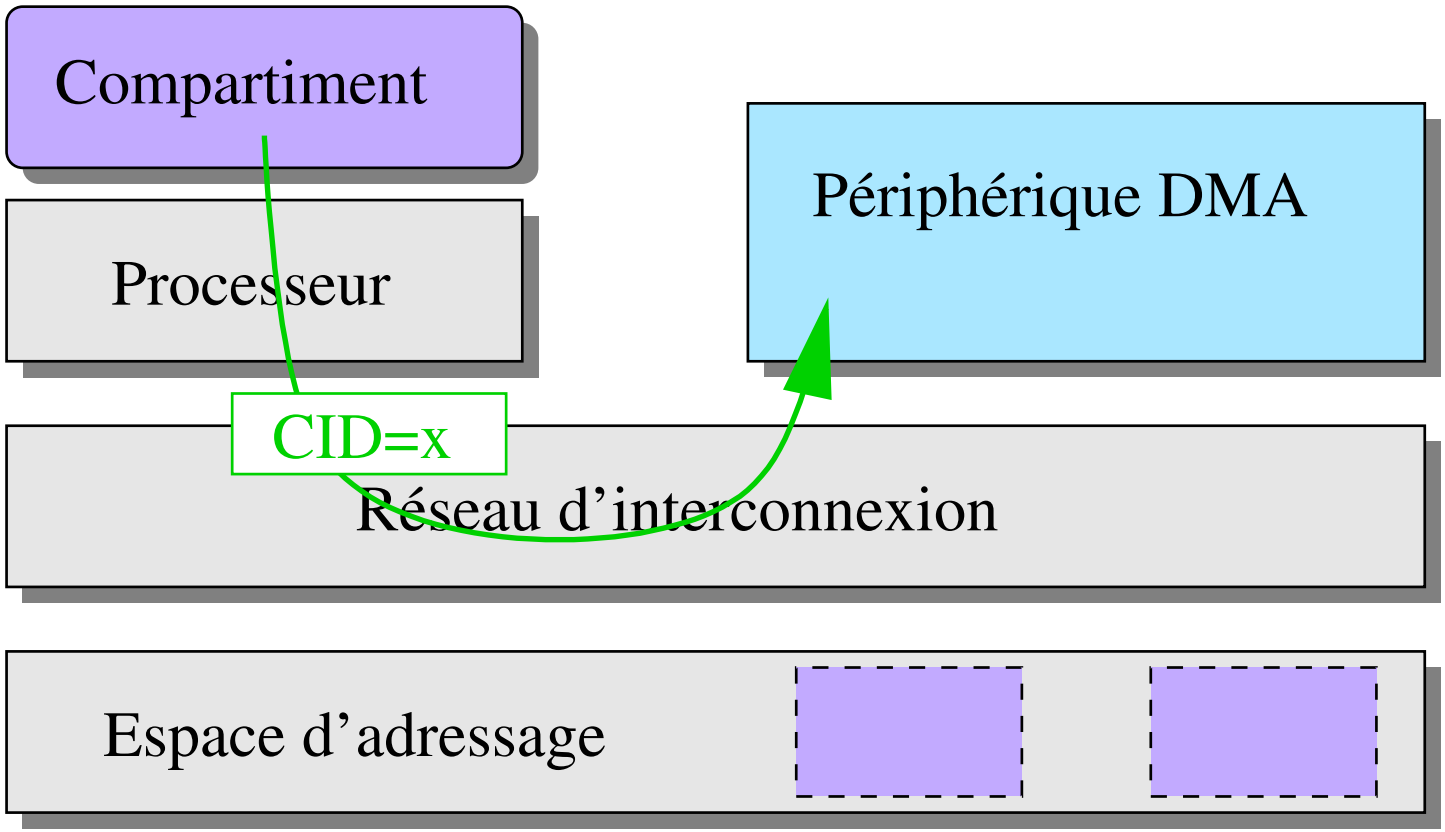
Identification - DMA de type mémoire à mémoire



Principe d'héritage du CID

- Héritage des droits d'accès du compartiment « commanditaire »

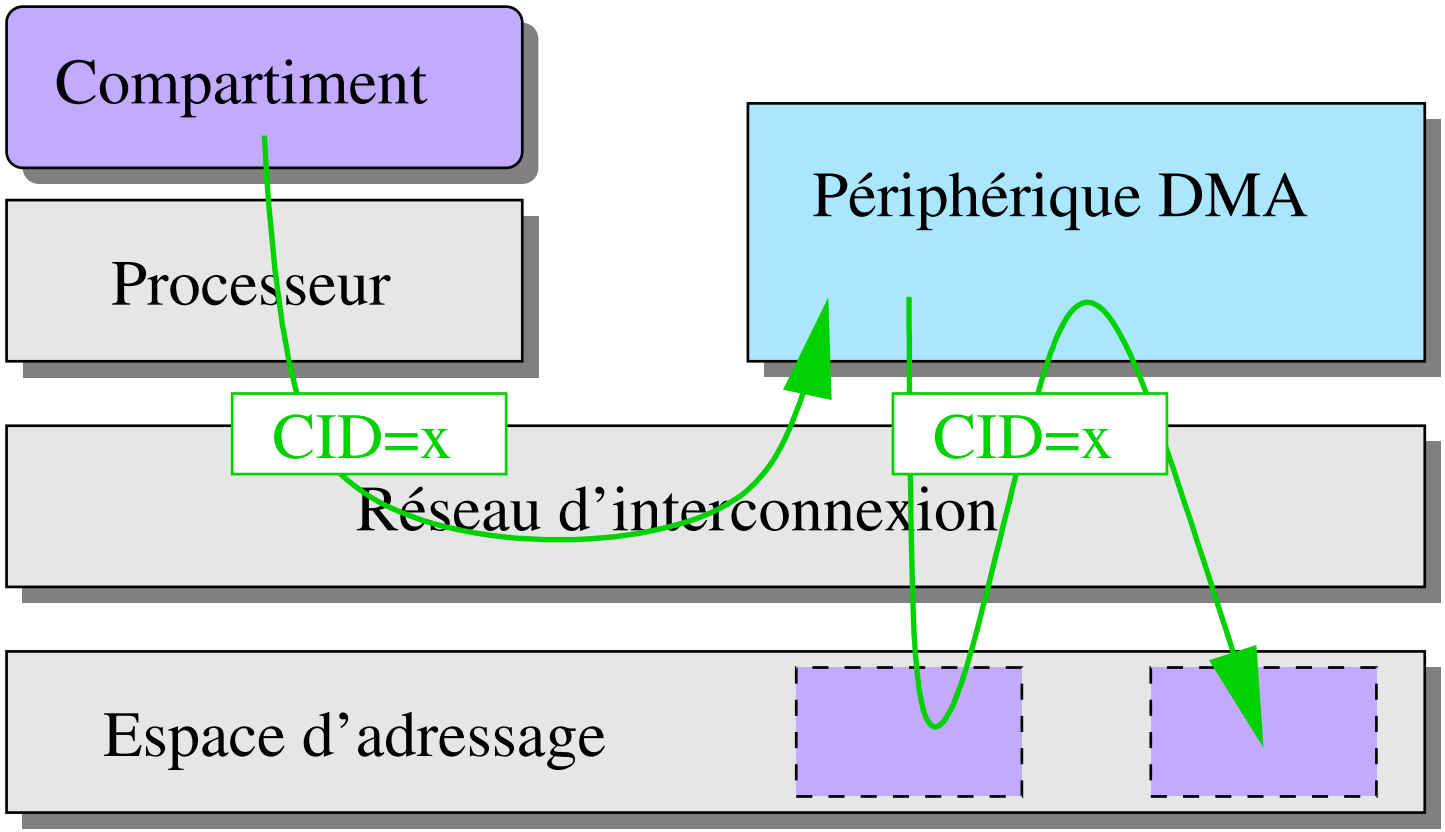
Identification - DMA de type mémoire à mémoire



Principe d'héritage du CID

- Héritage des droits d'accès du compartiment « commanditaire »

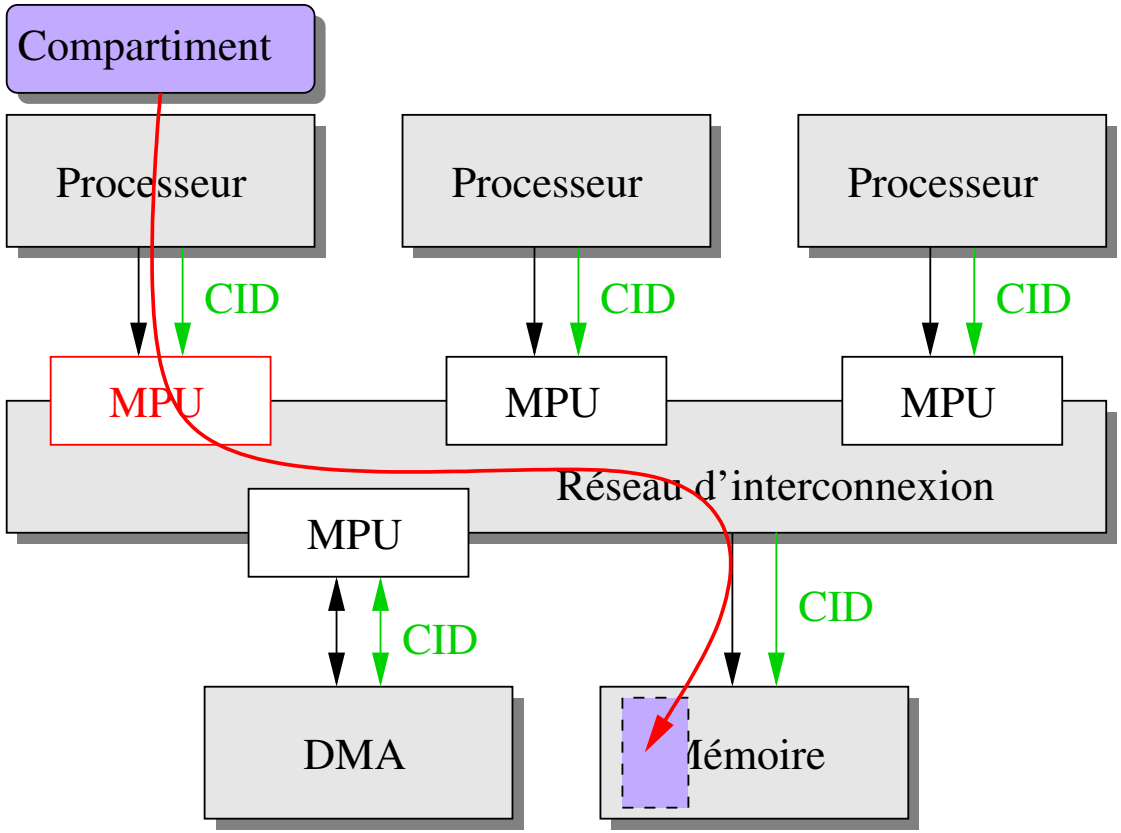
Identification - DMA de type mémoire à mémoire



Principe d'héritage du CID

- Héritage des droits d'accès du compartiment « commanditaire »

Protection - Intégration au réseau



Modules matériels : MPU (*Memory Protection Unit*)

- Fonction de « pare-feu »
- Table de permission : *(adresse,cid) = droits d'accès*

Protection - Intégration au réseau

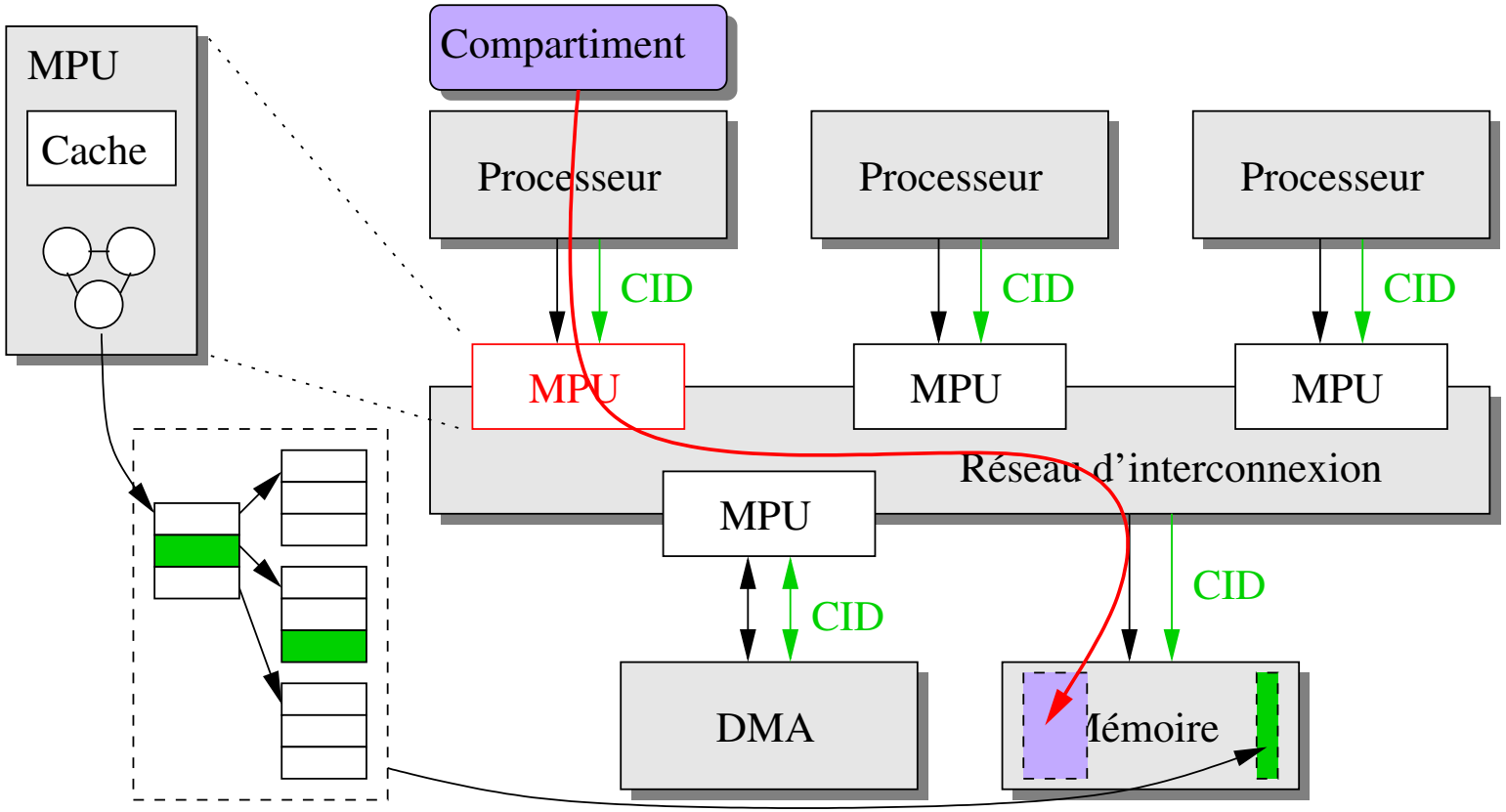
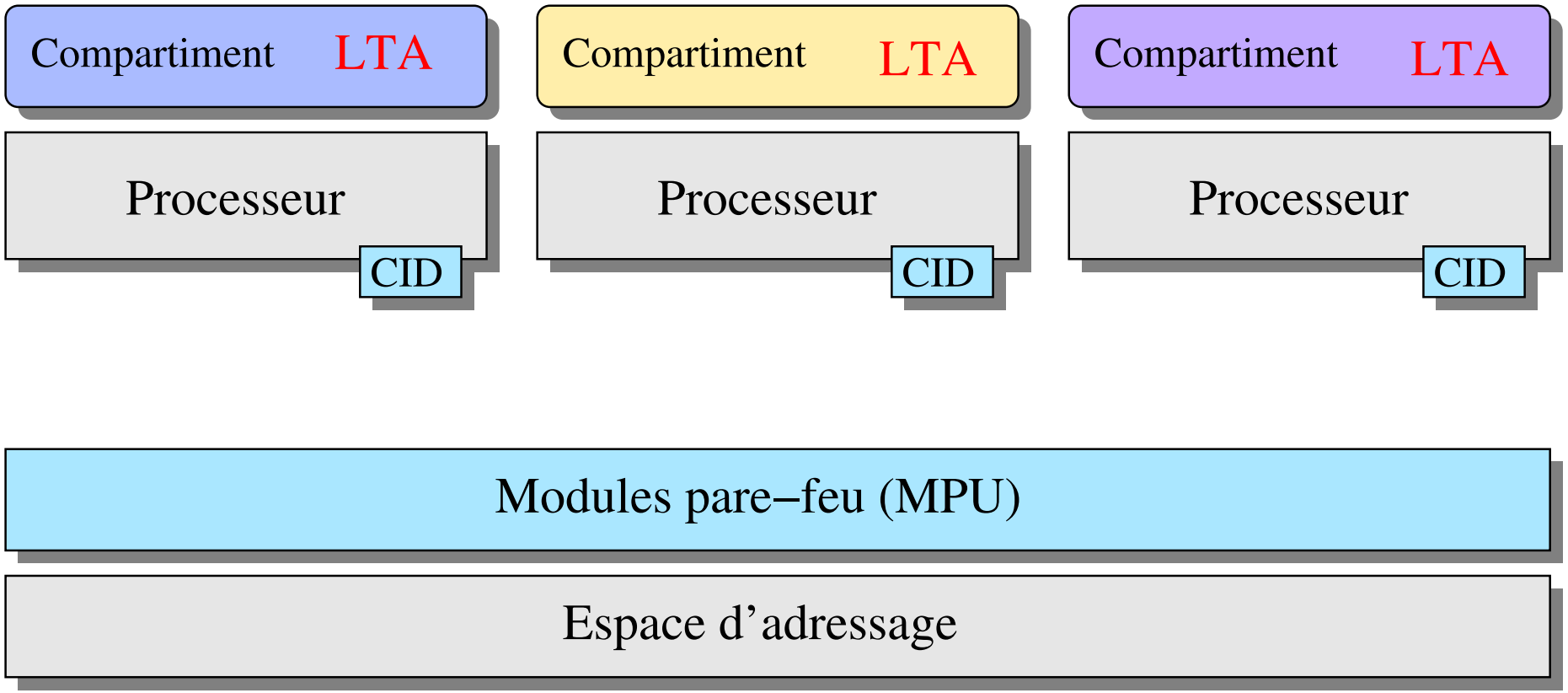


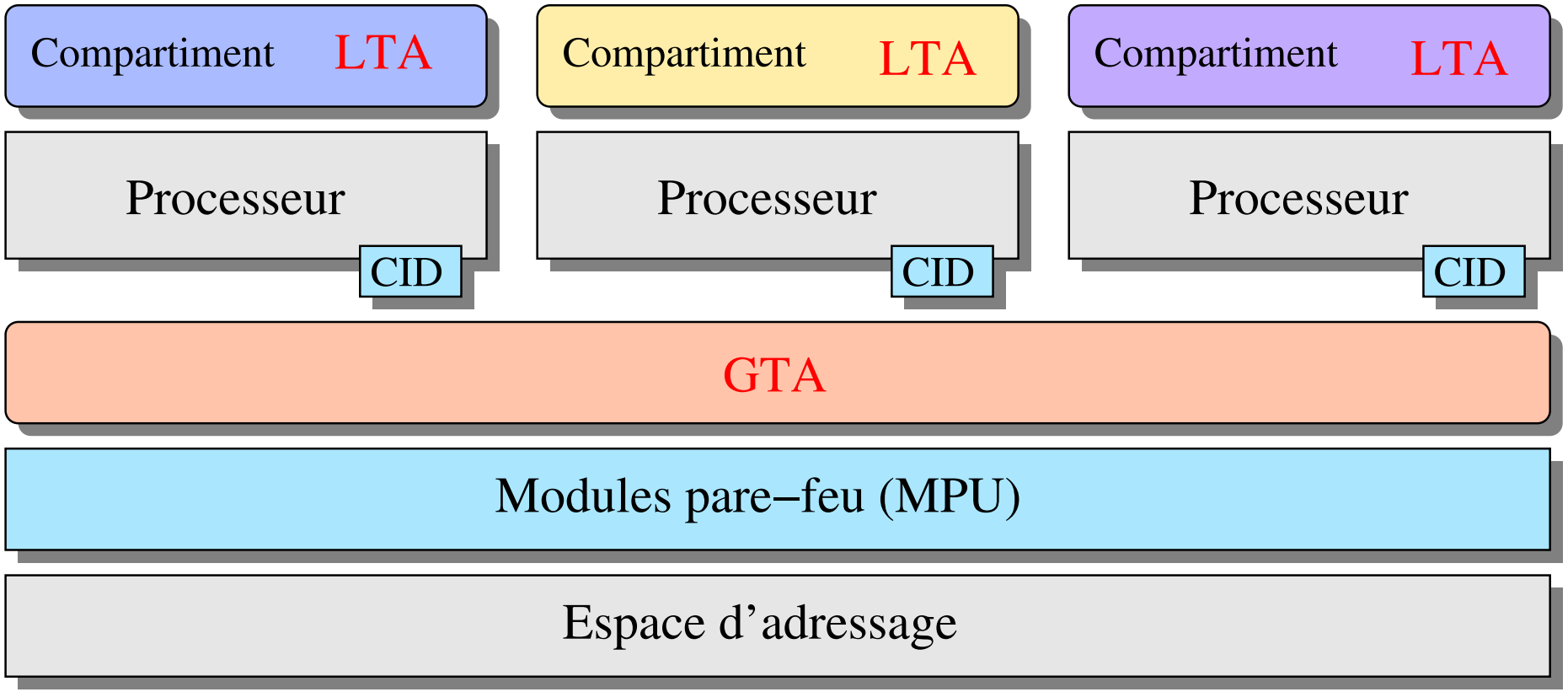
Table en mémoire et approche paginée

- Une table des pages par compartiment
- MPU : cache et automate de traitement des défauts de cache

Protection - Gestion globale



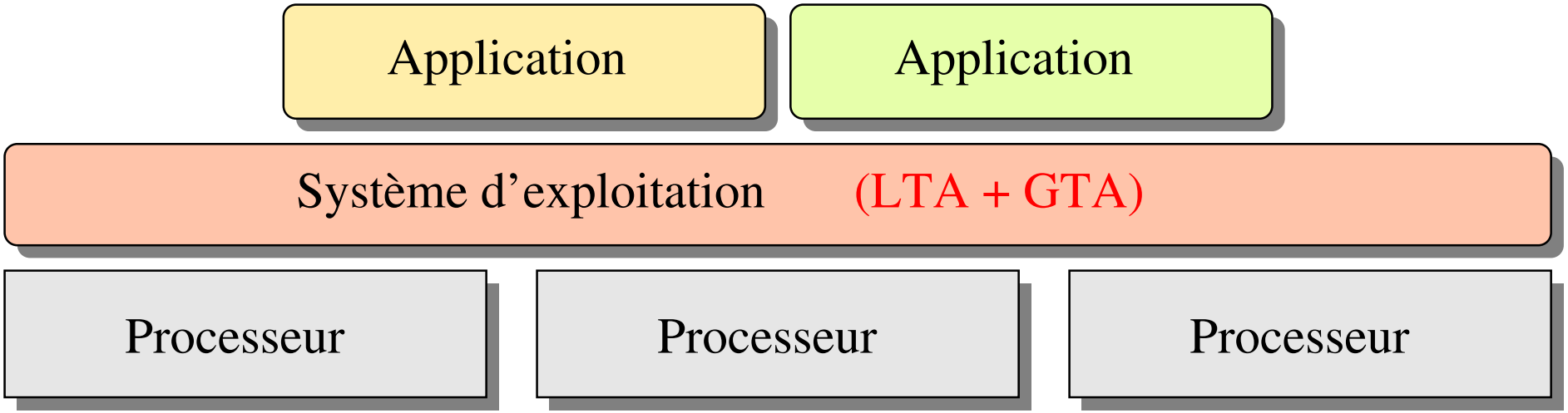
Protection - Gestion globale



Rôle de l'agent de confiance global (GTA, *Global Trusted Agent*)

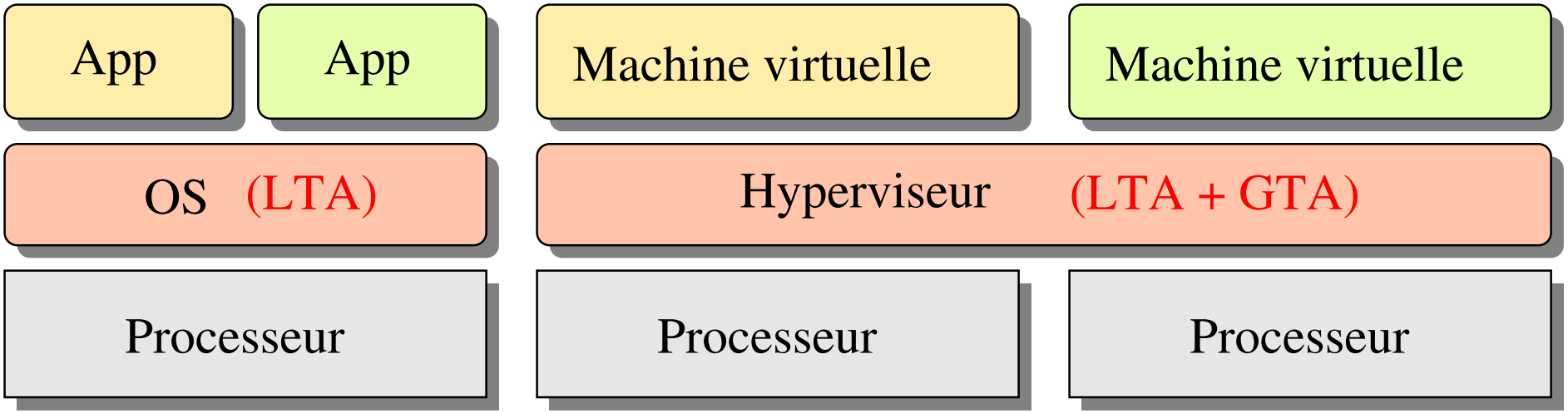
- Attribution des identifiants de compartiment (CID)
- Application de la sécurité : définition des tables des pages

Applications de l'approche



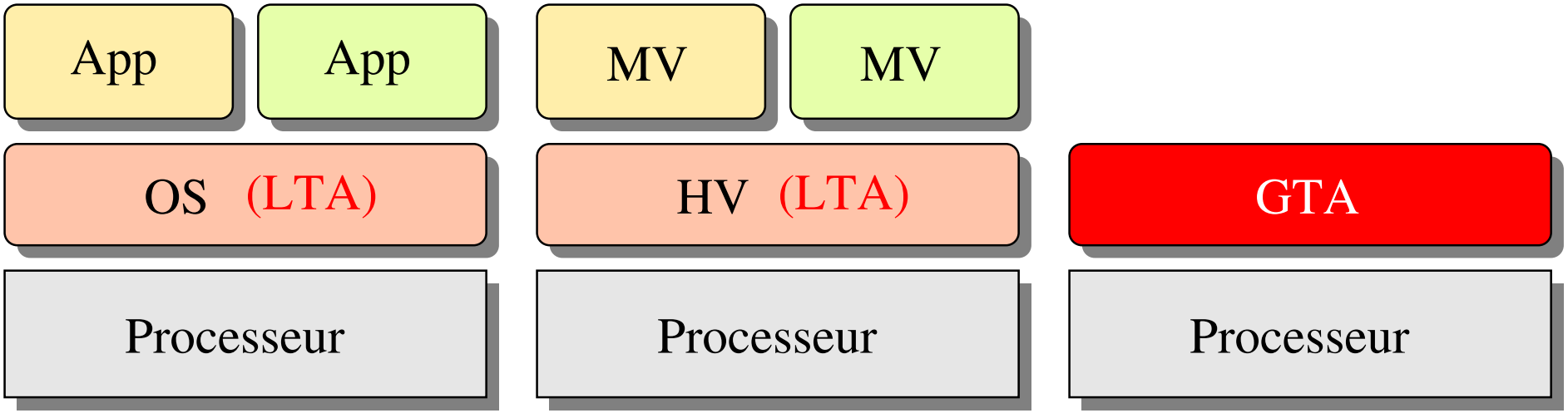
Unique pile logicielle autonome

Applications de l'approche



Multiples piles logicielles autonomes

Applications de l'approche



Paranoïaque : GTA isolé

Plan

Objectifs

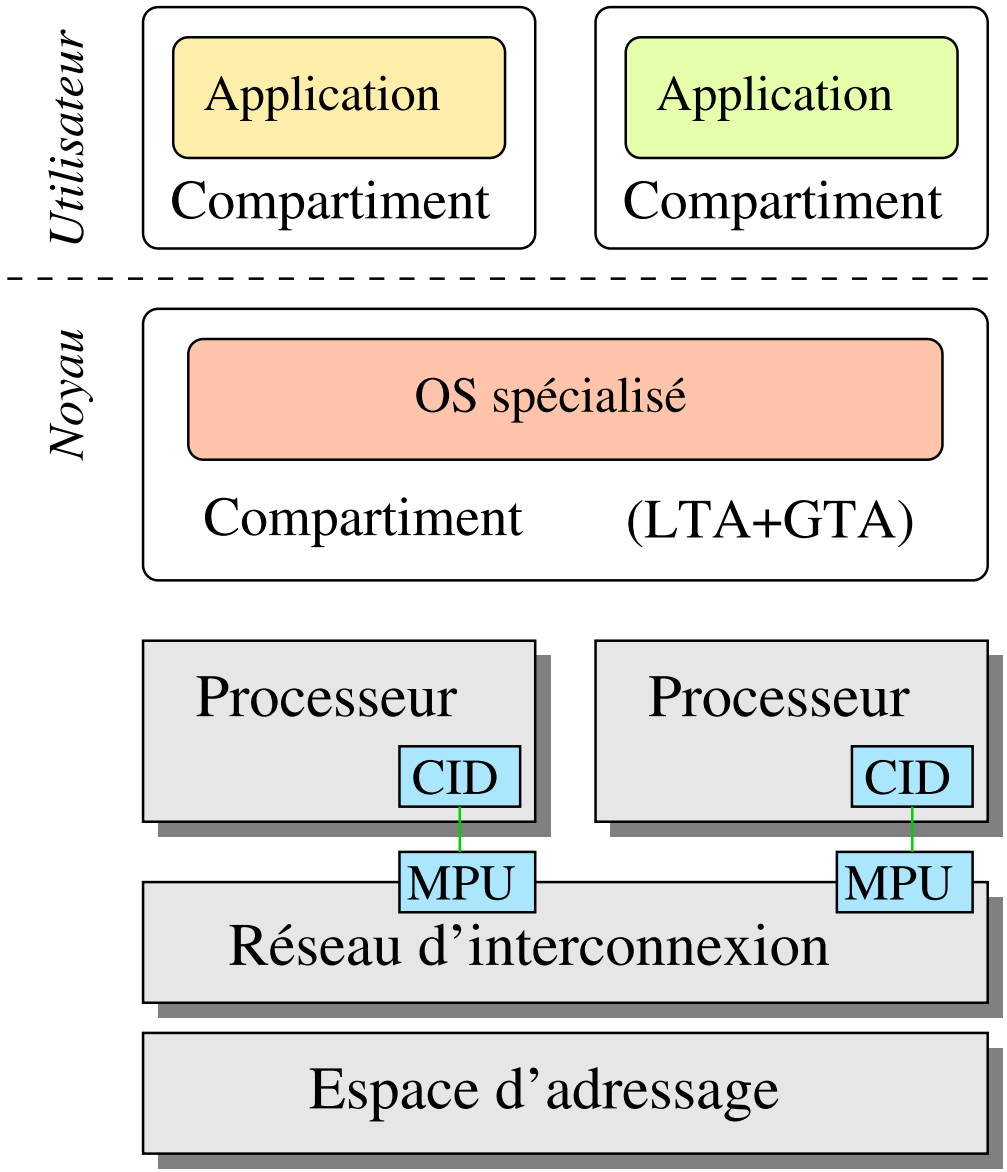
Preuve de concept

- Accent sur la flexibilité et la dynamicité

Évaluation des coûts

- Matériels et logiciels
- Performance et surface matérielle

Scénario


Utilisateur
Application
Compartiment
Application
Compartiment
Noyau
OS spécialisé
Compartiment
(LTA+GTA)
Processeur
CID
Processeur
CID
MPU
MPU
Réseau d'interconnexion
Espace d'adressage

Mise en œuvre matérielle

SoCLib

- Environnement de prototypage virtuel en `SystemC`

Propagation du CID

- Le protocole `VCI` fournit le champ `TRDID`

Processeur multi-compartiment

- `MIPS32` : modes d'exécution *utilisateur/noyau*, sans MMU
- Registre de CID dans un coprocesseur interne
- Étiquetage des caches processeurs avec le CID

MPU

- Intégration au réseau sur puce `DSPIN`
- Localisés dans les interfaces d'accès au réseau, en parallèle avec la traduction de protocole

Mise en œuvre matérielle

SoCLib

- Environnement de prototypage virtuel en `SystemC`

Propagation du CID

- Le protocole `VCI` fournit le champ `TRDID`

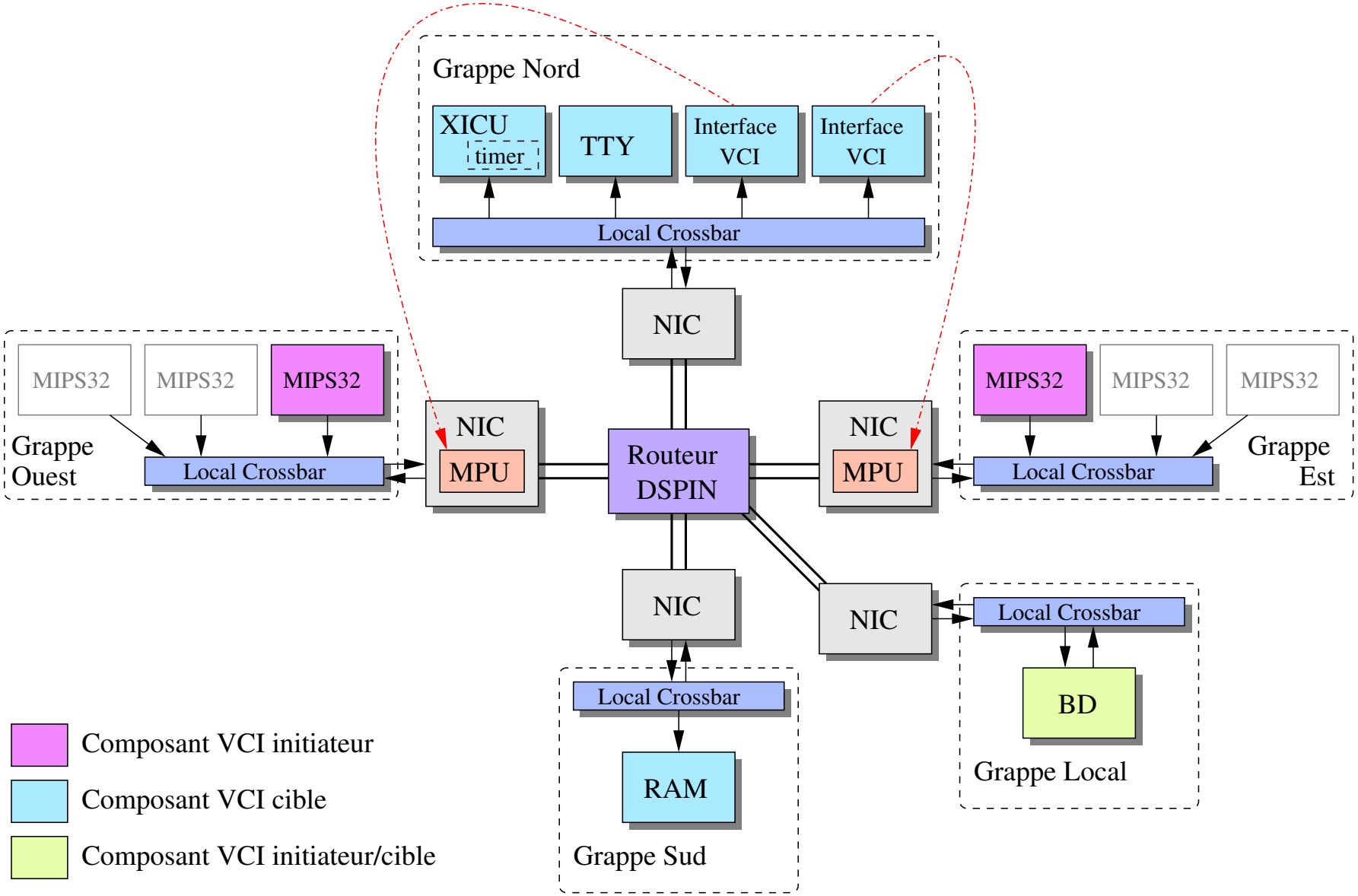
Processeur multi-compartiment

- `MIPS32` : modes d'exécution *utilisateur/noyau*, sans MMU
- Registre de CID dans un coprocesseur interne
- Étiquetage des caches processeurs avec le CID

MPU

- Intégration au réseau sur puce `DSPIN`
- Localisés dans les interfaces d'accès au réseau, en parallèle avec la traduction de protocole

Plateforme matérielle d'expérimentation



Mise en œuvre logicielle

MutekH

- Exo-noyau pour systèmes embarqués

Applications *utilisateur*

- Interface noyau/utilisateur : `libsyscall`, `libuser`
- Chargement dynamique : `libelf`

Fonctions de LTA

- Gestion du registre de CID
- Ordonnancement des compartiments

Fonctions de GTA

- Attribution des identifiants,
- Création et mise à jour des tables de pages de protection : `libgta`

Mise en œuvre logicielle

MutekH

- Exo-noyau pour systèmes embarqués

Applications *utilisateur*

- Interface noyau/utilisateur : `libsyscall`, `libuser`
- Chargement dynamique : `libelf`

Fonctions de LTA

- Gestion du registre de CID
- Ordonnancement des compartiments

Fonctions de GTA

- Attribution des identifiants,
- Création et mise à jour des tables de pages de protection : `libgta`

Tests de performance

Applications de test orientées multimédia

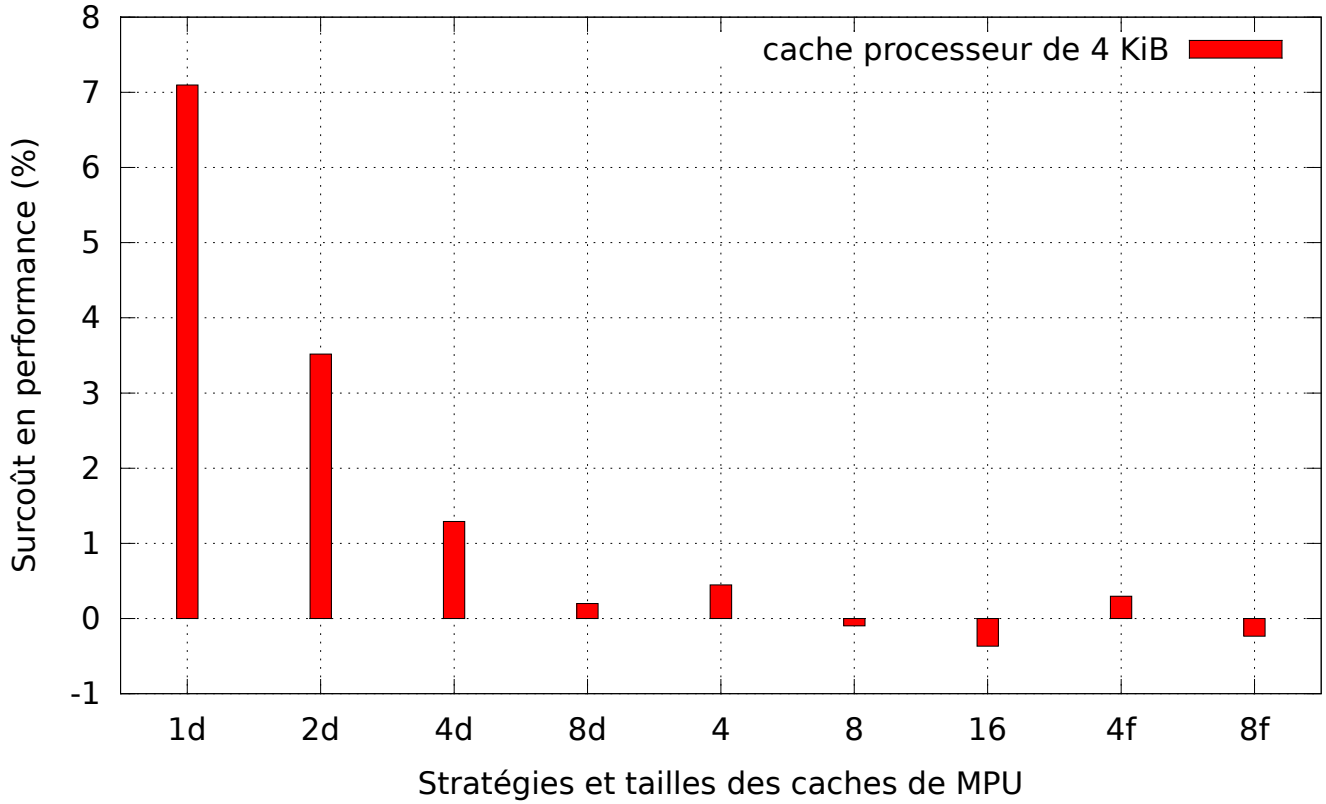
- Six compartiments :
  - Codeur et décodeur JPEG
  - Codeur et décodeur XviD
  - Décodeur MP3
  - Chiffrement AES
- Occupation mémoire :
  - De 50 kilooctets à 2 mégaoctets

Type d'exécution

Exécution sans mécanisme de protection, puis avec, et mesure de la dégradation en performance

- Plateforme à deux processeurs : commutation tous les 2 millions de cycle (10ms à 200 mégaherzt) et migration possible des compartiments
- Plateforme à six processeurs : assignation statique des compartiments

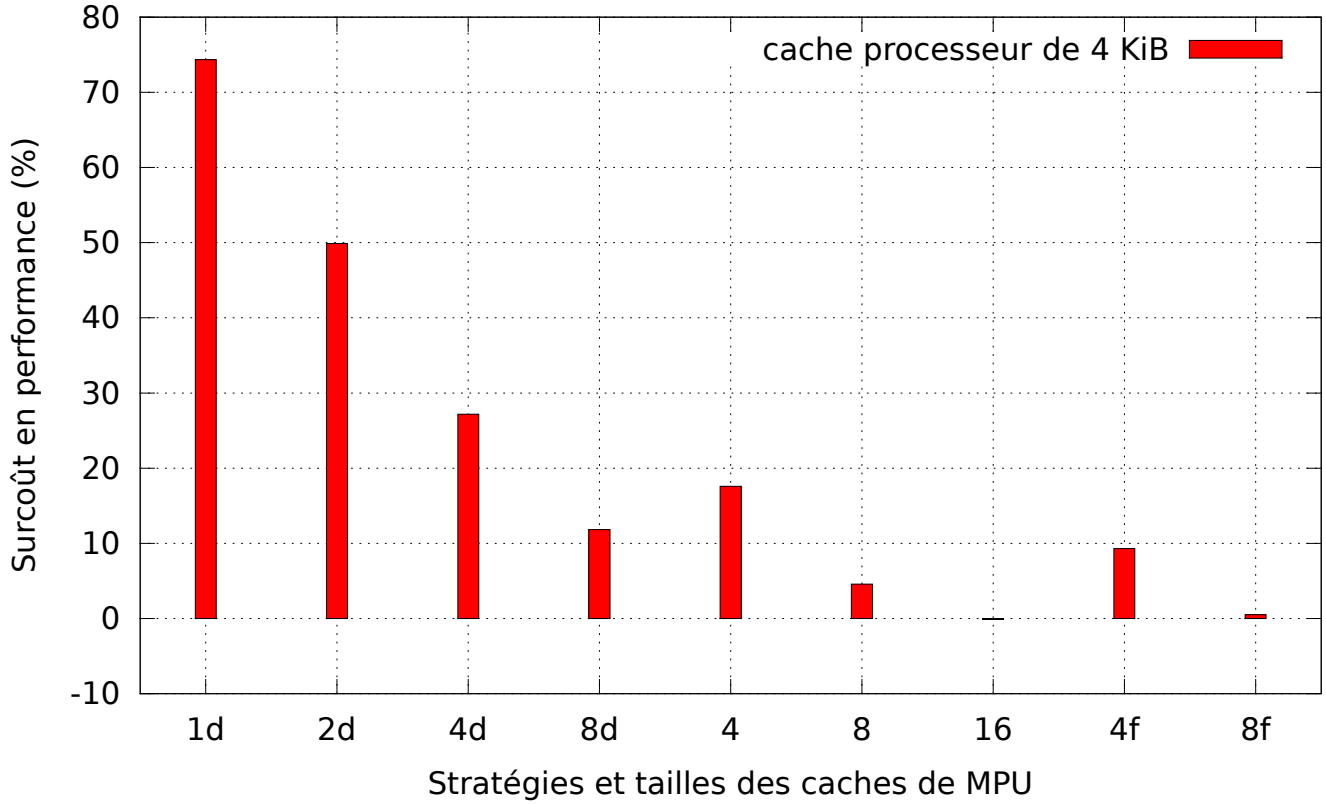
Coûts en performance - deux processeurs



MPU : différentes stratégies de caches à n-entrées

- Correspondance direct (`1d`, `2d`, `4d`, `8d`) ; Associatif par ensemble à 2 voies (`4`, `8`, `16`) ; Totalement associatif (`4f`, `8f`)

Coûts en performance - six processeurs



- Montée en charge : trois processeurs attaquent simultanément une MPU
- Test des mêmes stratégies de caches que précédemment

Coûts en surface matérielle

Processeur

- Ajout d'un registre de contrôle
- Extension de l'interface de sortie

Caches processeurs

- Extension de l'étiquette
- 2% d'augmentation pour des lignes de caches de 16 mots, avec un CID sur 8 bits.

MPU

- Estimation par comptage des bits mémorisants
- 3% d'augmentation par rapport à l'infrastructure du réseau sur puce, sur notre plateforme d'expérimentation

Plan

Conclusion sur l'approche multi-compartiment

Concept

- Représentation d'entités logiques et protection, au niveau matériel

Propriétés

- Co-hébergement multi-niveaux
- Cohérent avec les aspects MP-SoC
- Sécurisé, flexible et dynamique
- À faible coût (mise en œuvre, performance et surface)
- Permet l'accès direct aux périphériques de type DMA
- Supporte nativement la technologie ARM TrustZone
- S'intègre de façon transparente dans un réseau sur puce

Perspectives

Poursuite de la validation

- Influence de la latence entre les MPU et les tables de pages
- Scénario plus complexe : Linux, etc.

Perspectives

Poursuite de la validation

- Influence de la latence entre les MPU et les tables de pages
- Scénario plus complexe : Linux, etc.

Adaptation à d'autres types de plateforme

- Intégration dans une plateforme matérielle *manycore*, en grappes (par exemple, *TSAR*)

Perspectives

Poursuite de la validation

- Influence de la latence entre les MPU et les tables de pages
- Scénario plus complexe : Linux, etc.

Adaptation à d'autres types de plateforme

- Intégration dans une plateforme matérielle *manycore*, en grappes (par exemple, *TSAR*)

Virtualisation de l'espace d'adressage

- Intégration de MMU dans le réseau sur puce (*NoC-MMU*)
- Coûts en performance et en surface ?

Publications

Conférences internationales

- Joël Porquet, Christian Schwarz and Alain Greiner. Multi-compartment : a new architecture for secure co-hosting on SoC. *In SoC'09 : Proceedings of the 11th International Symposium on System-on-Chip*, pages 124–127, 2009.
- Joël Porquet, Christian Schwarz and Alain Greiner. NoC-MPU : a secure architecture for flexible co-hosting on shared memory MPSoCs. *Accepted at DATE'11 : the Conference on Design, Automation and Test in Europe*, 2011.

Brevets (Europe et États-Unis)

- Joël Porquet and Christian Schwarz. Method of routing an interrupt signal directly to a virtual processing unit in a system with one or more physical processing units, 2010.
- Joël Porquet and Christian Schwarz. Method for enabling several virtual processing units to directly and concurrently access a peripheral unit, 2010.

Merci de votre attention