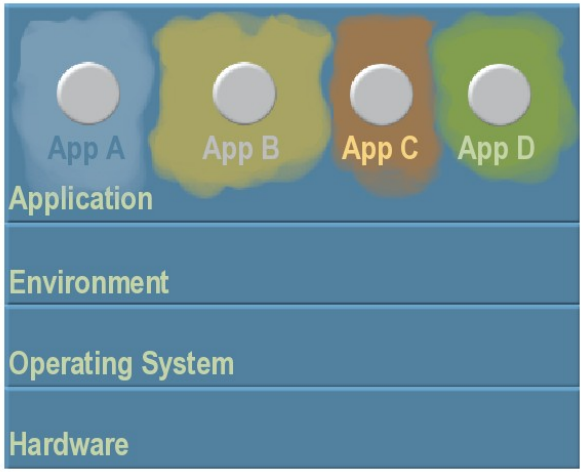
Virtualisation
Part 2
Virtualisation serveurs

Bernard PIERRÉ
Architecte
Sun Microsystems Strasbourg

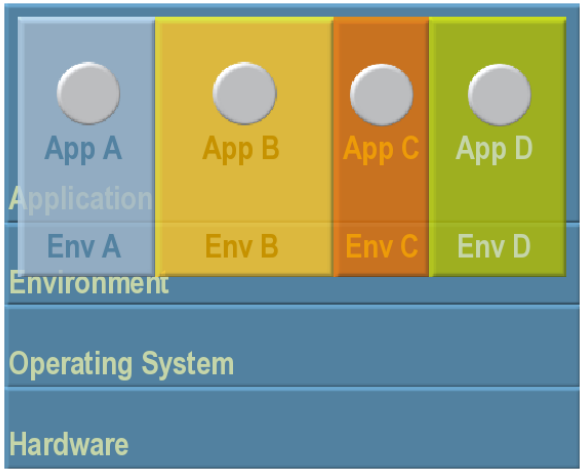
Agenda : Partie 2

- Virtualisation Serveurs
  - > Offre virtualisation Sun
  - > Partitionnement matériel
  - > Solaris Containers
  - > Virtualisation système
  - > Sun xVM
  - > Logical Domains

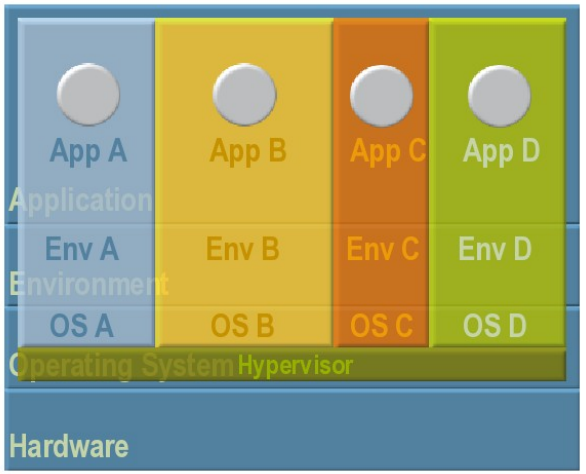
Types de virtualisation « serveurs »



Gestion ressources



Virtualisation OS



Machines Virtuelles

Partitions matérielles

Des moyens : caractéristiques …

Tendance : flexibilité

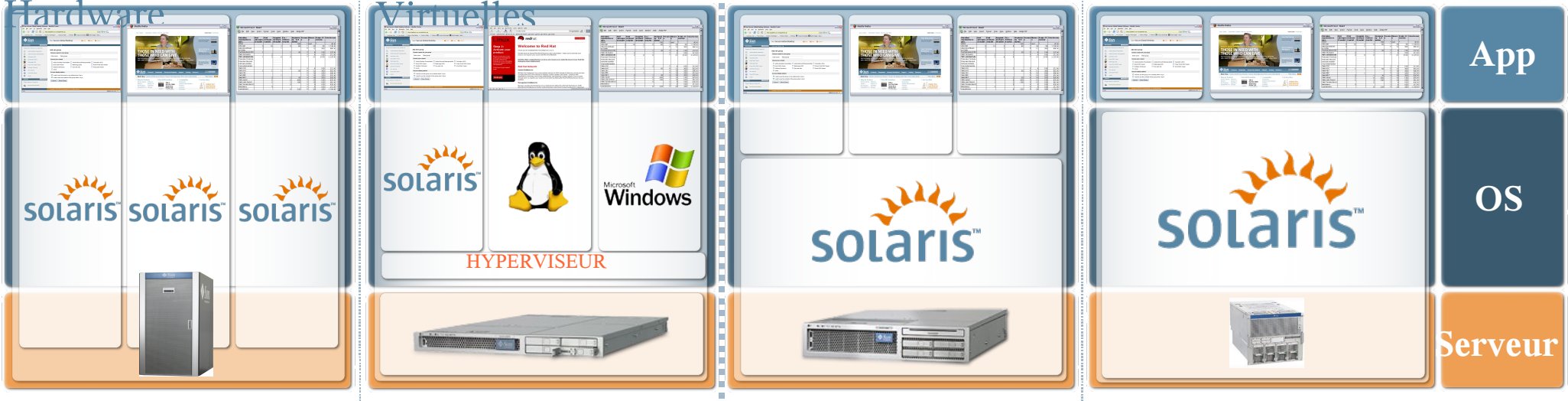
Tendance : isolation

Partitions Hardware

Machines Virtuelles

Virtualisation OS

Gestion des ressources



Multiples OS à gérer

OS unique

Très haute disponibilité, maintenabilité, fiabilité (RAS)

Très Scalable

Technologie Mature

Capacité à exécuter différentes versions d'OS

Capacité à déplacer un OS « à chaud »

Capacité à gérer différents types et versions d'OS

Découplage du hardware et des versions OS

Très « scalable » et faible overhead

Un unique OS à gérer

Division propre de l'administration du système et des applications

Gestion des ressources à faible granularité

Très « scalable » et faible overhead

Un unique OS à gérer

Gestion des ressources à faible granularité

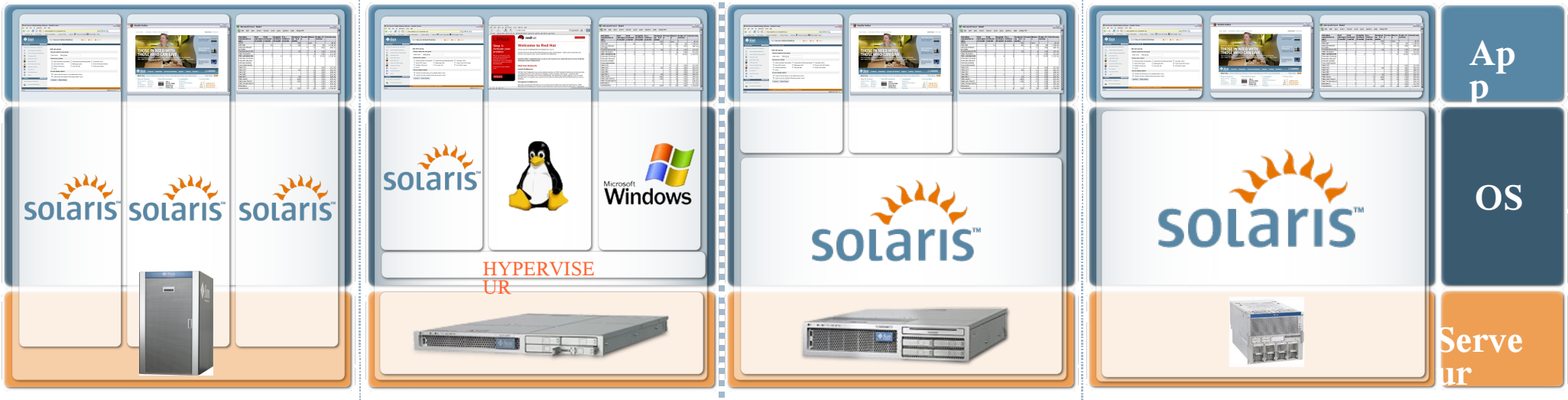
Serveurs Sun : Offres de virtualisation

Partitions physiques

Machines Virtuelles

Virtualisation OS

Gestion de ressources



Multiples OS à gérer

OS unique

Dynamic System Domains (série M)

Sparc Ldoms (série T)

Solaris 8 & 9 Containers

Containers Solaris (Zones + SRM)

Solaris ressource Manager (SRM)

xVM infrastructure Vmware, Xen

Microsoft Hyper-V

xVM VirtualBox

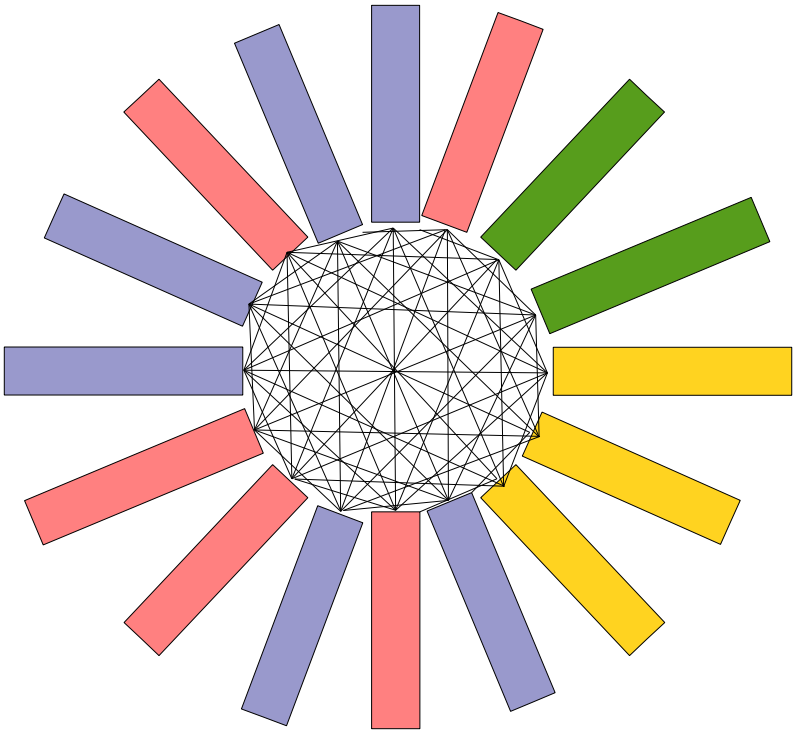
Sparc

x64

Partitionnement physique

Dynamic System Domains

- Naissance chez Sun en 1997 : serveur E10000



Architecture « crossbar »
Composée de 16 cartes de 4 processeurs
Cartes CPU/MEM reliées 2 à 2 par un crossbar
Liens activables par Service Processor
Chaque groupe de cartes forme un domaine
Chaque domaine a sa propre image Solaris

Dynamic System Domains
Les cartes peuvent êtres ajoutées, retirées,
Déplacées d'un domaine à l'autre,
dynamiquement
À chaud, sans interruption de l'instance Solaris

Histoire des domaines dynamiques

- Naissance chez Sun en Janvier 1997
  - > serveur E10000 : 64 procs UltraSparc II
- Repris sur gamme SunFire Enterprise
  - > Désormais sur toute la gamme UltraSparc III et IV
  - > midrange E2900, E4900, E6900 et high end SunFire 12k, 15k, 20k, 25k
- Repris sur gamme Série M aujourd'hui
  - > Aujourd'hui sur toute la gamme Sparc64 VI
  - > M4000, M5000, M8000, M9000
  - > Amélioration : granularité “à la CPU près”

Hard Partitions sur Sun M-Series

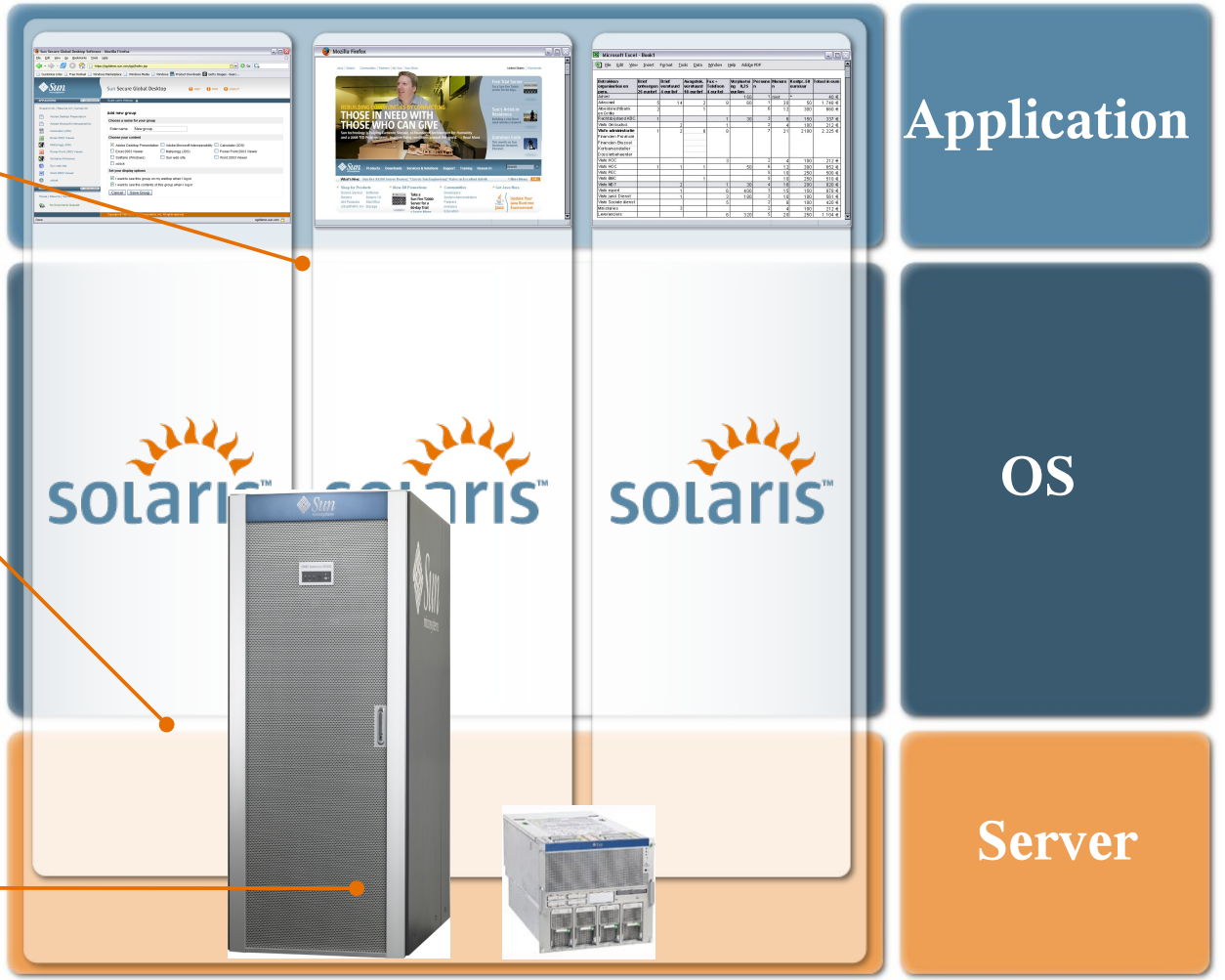
Isolation matérielle totale

(haute disponibilité)

Granularité au processeur (1-CPU)

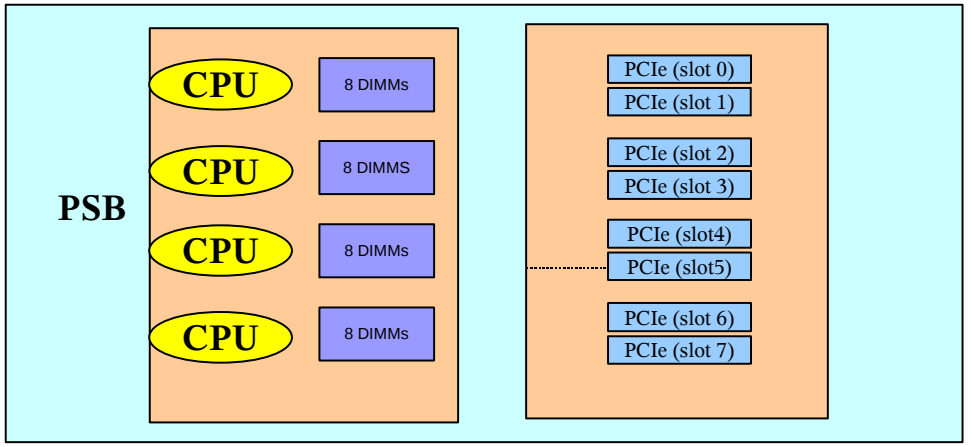
M4000/2 domaines
M5000/4
M8000/16
M9000-32/24
M9000-64/24

SPARC milieu et haut de gamme (M-4000 à M-9000*)

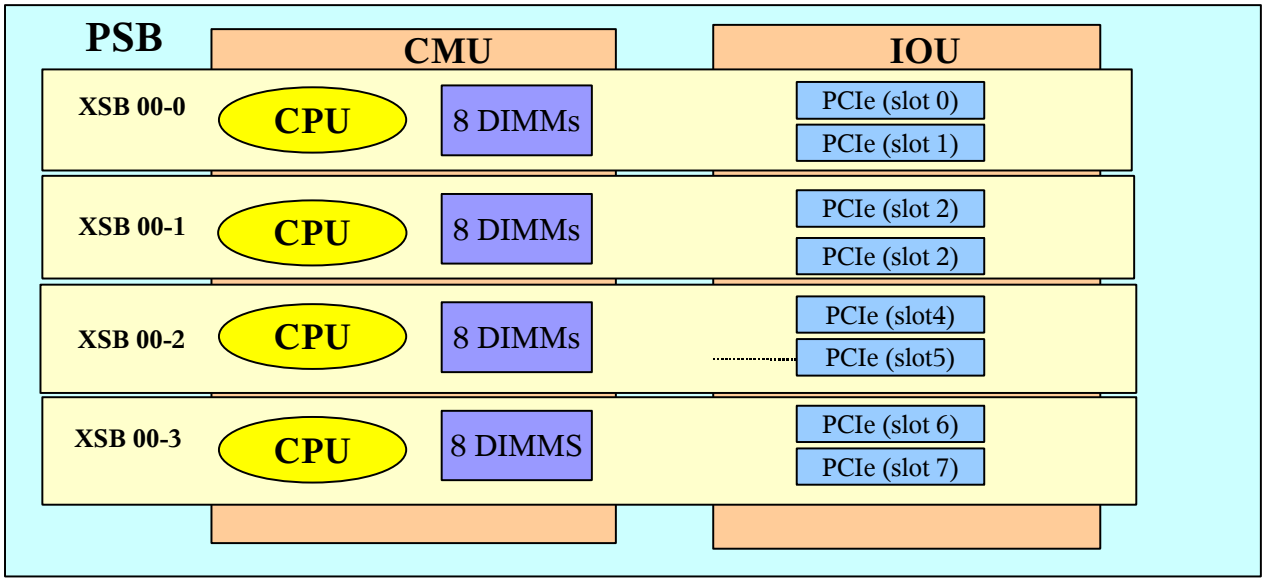


XSB's sur M8000/M9000

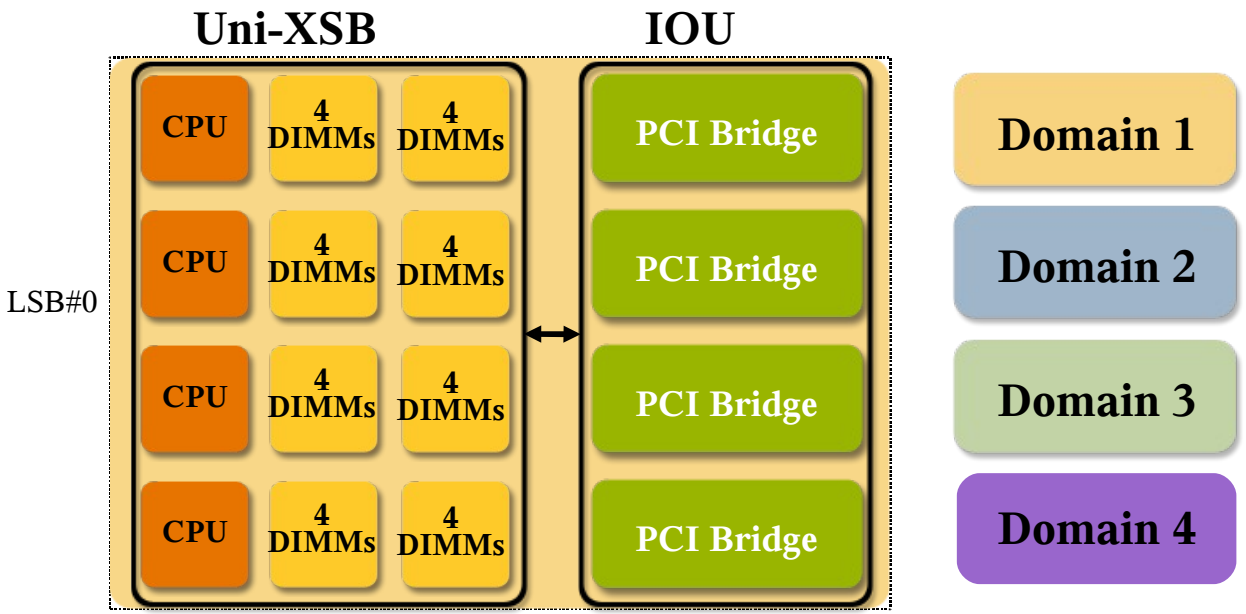
PSB en Uni-Mode = Uni-XSB



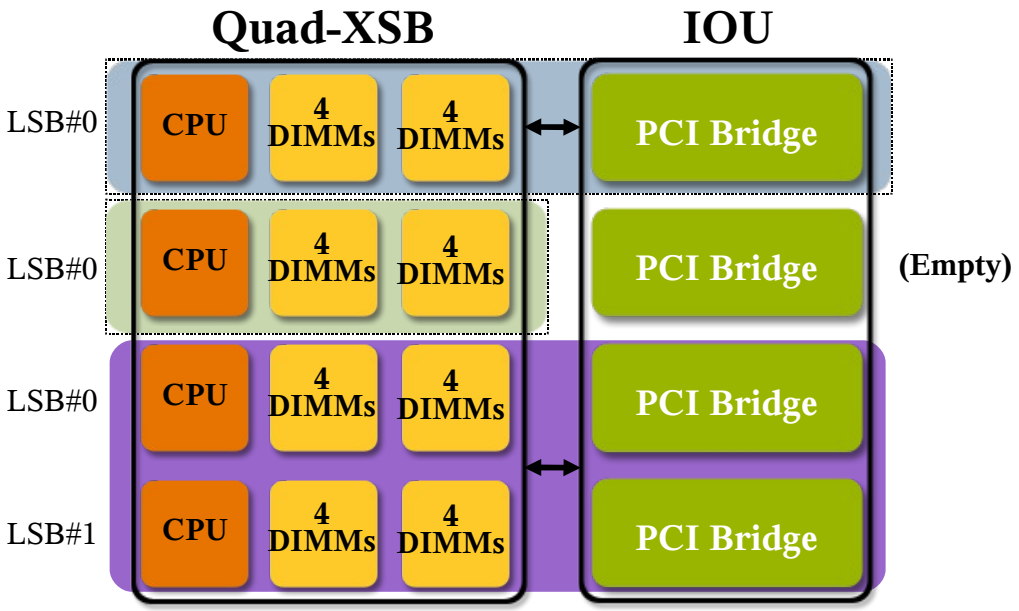
PSB en Quad-Mode = Quad-XSB

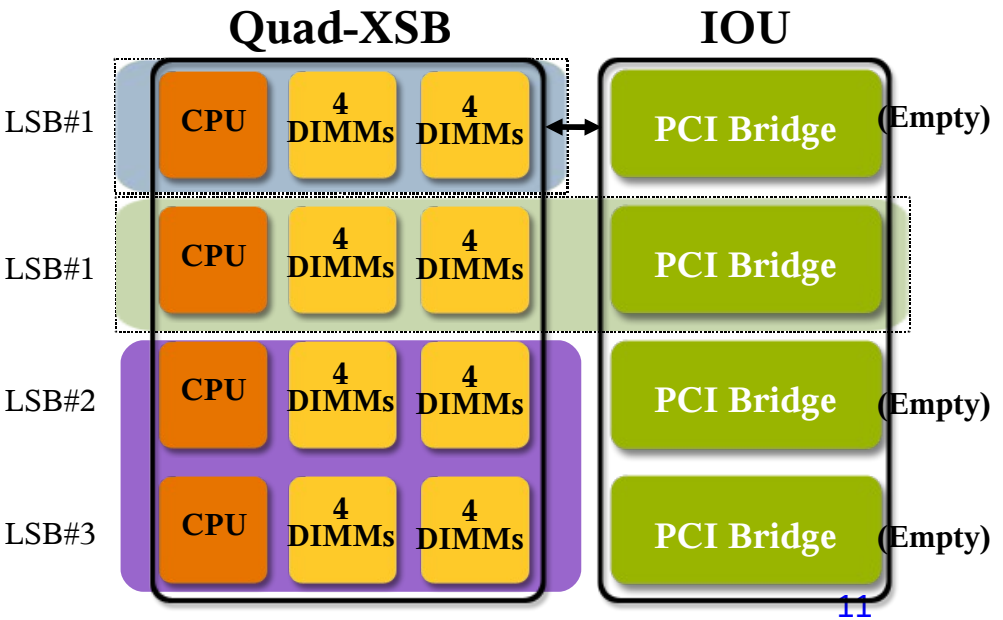


Exemple de partitionnement M8000-M9000



- ***Règle de Base/Contrainte***
- ***L'utilisation d'une CMU ne requiert pas d'IOU installée***
- ***L'utilisation d'une IOU requiert une CMU installée***

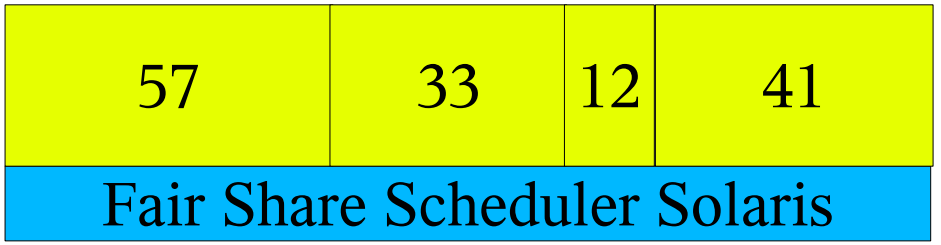




Sun microsystems

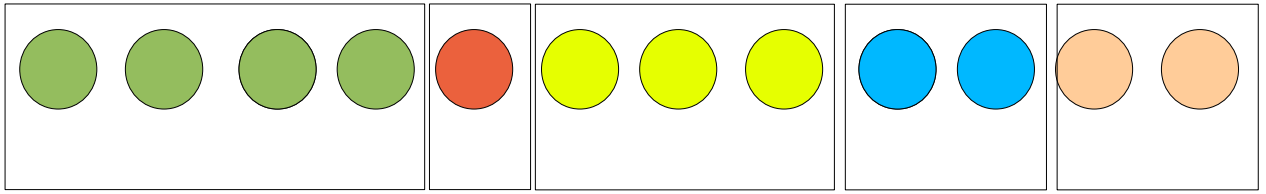
Solaris Containers

Solaris Ressource Manager



Principe de gestion par parts (granularité quelconque)
Allocation en 'time sharing' par Solaris Scheduler
 A des projets (groupes de processus)
 A des zones (solaris 10)
Chaque entité peut consommer sa part
Si elle ne la consomme pas, les autres entités y ont accès
Principe de minimum garanti sur demande

Ressource Management : Processor sets

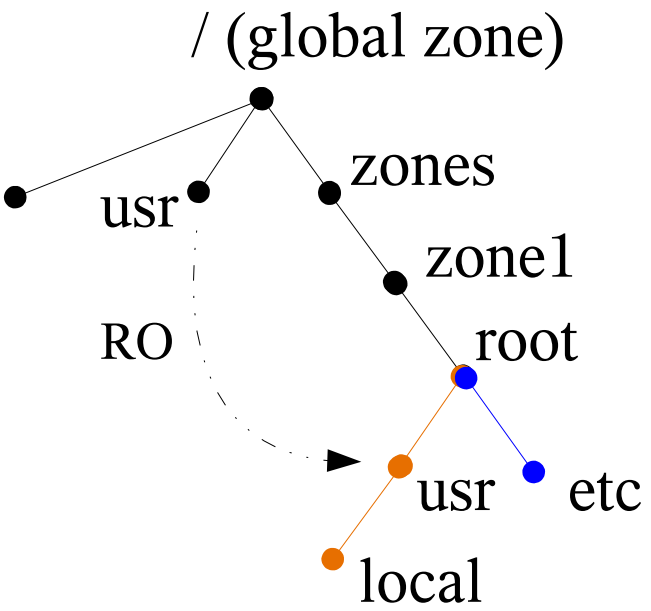


Principe d'allocation avec granularité « à la CPU » :
 CPUs réelles (processeurs traditionnels)
 CPUs virtuelles (1 thread d'un core d'un chip CMT)
Allocation par :
 Binding de processus (une seule image OS : S8, S9)
 Création de pools (solaris 10)

Principe de partitionnement étanche
Les processus hors du container ne peuvent accéder aux CPUs associées

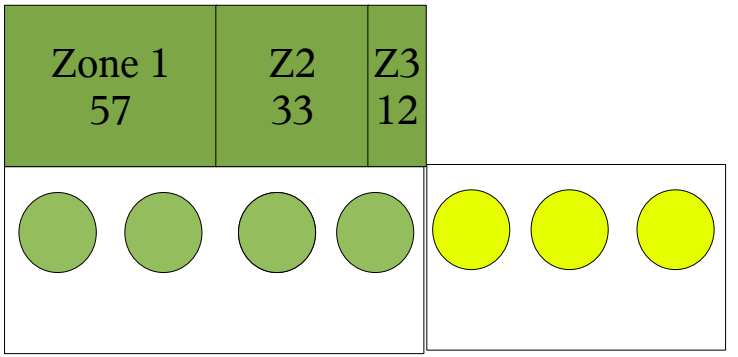
Solaris 10 Zones

- Une seule installation Solaris : global zone
- Création d'entités administratives isolées : zones
- Chaque zone
  - > Possède ses propres fichiers système
  - > Est vue comme une machine virtuelle
  - > Arborescences privées ou partagées
  - > Ressources E/S partagées ou privées
- La zone globale administre les zones

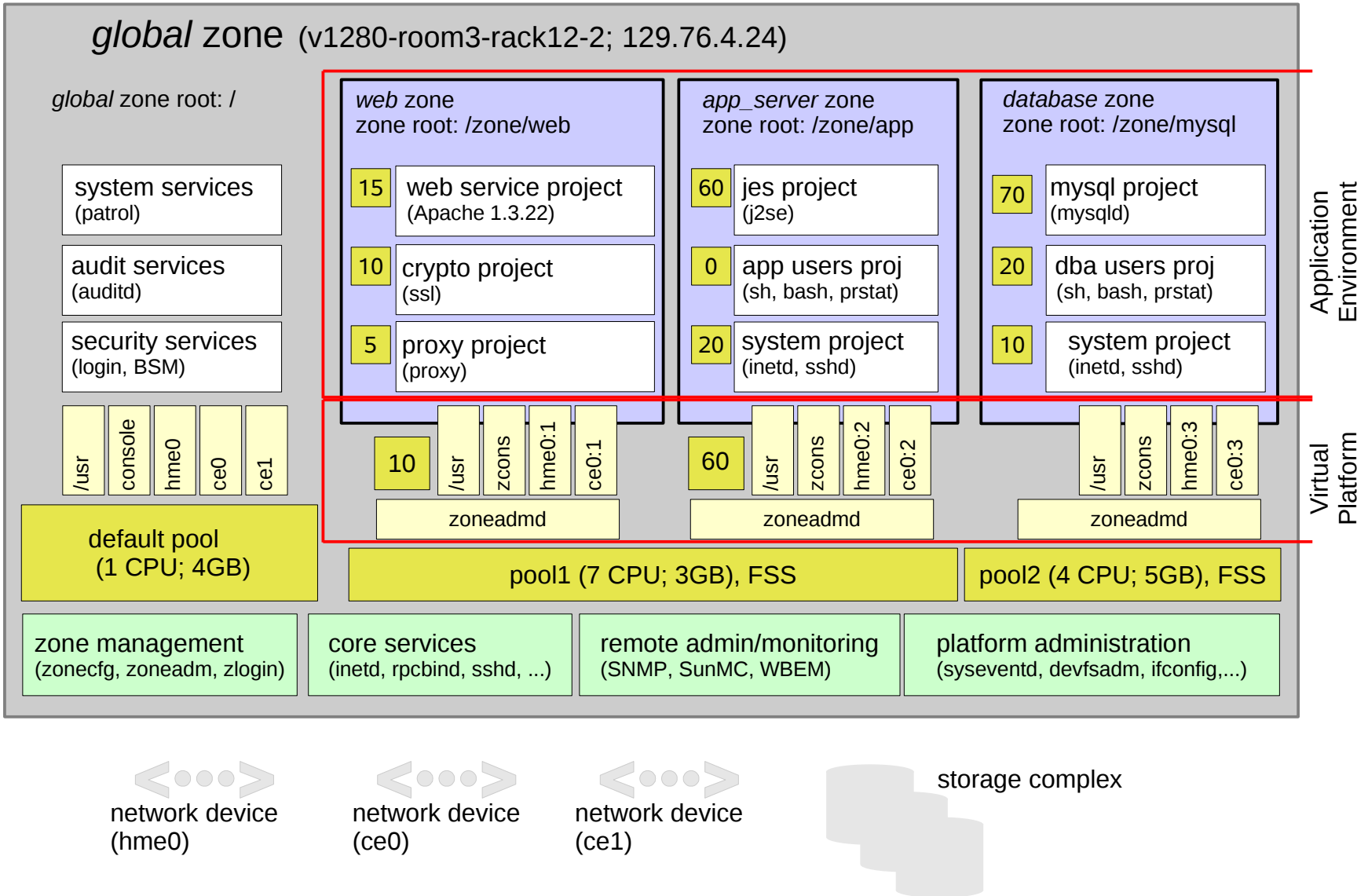


Solaris 10 Containers

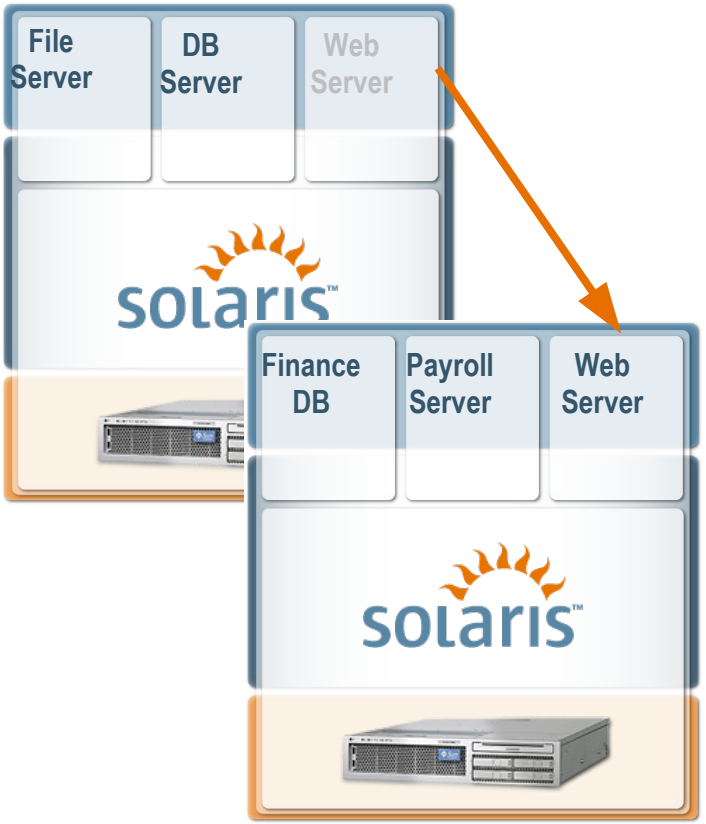
- Containers
  - > Vision machine virtuelle : zone
  - > Gestion de ressources
- Gestion ressources
  - > Création ressource pools (facultatif)
  - > Création zones dans des pools
  - > FSS entre zones dans un pool



Solaris Containers

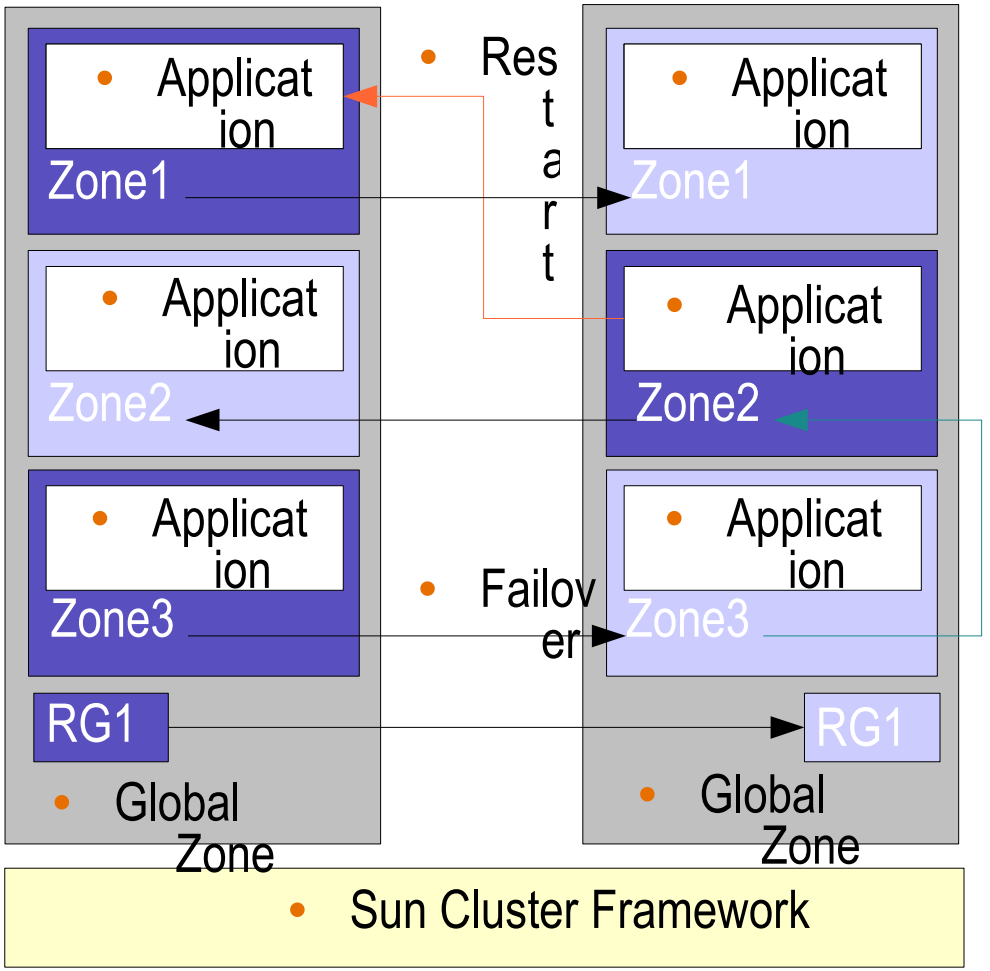


Solaris Containers



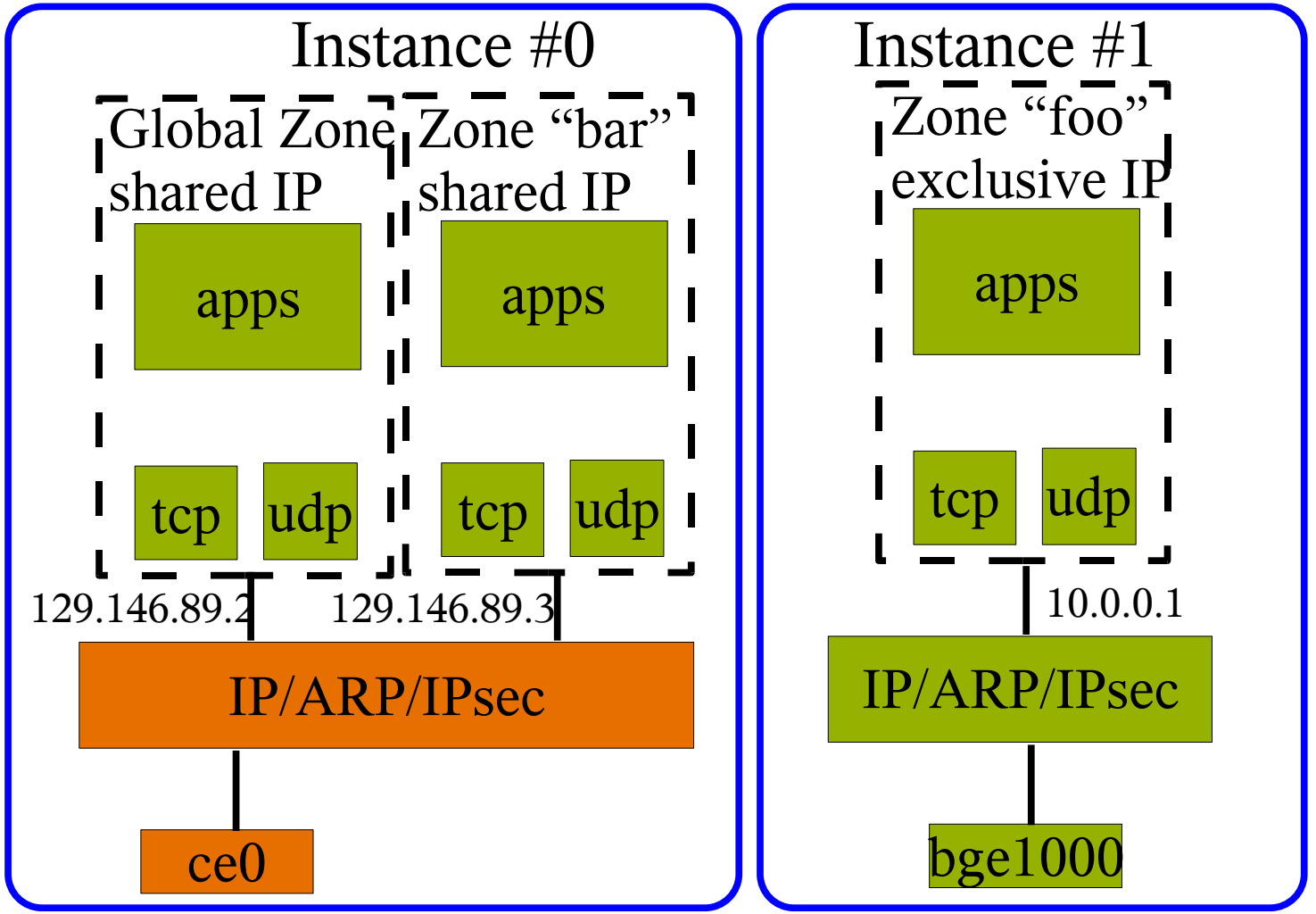
- Réplication rapide par clonage
- Migration de système à système
- Renommage
- Privilèges de sécurité
- Allocation CPU et memoire
- Piles réseau privées ou partagées
- Gestions des zones par zfs (snap, ..)
- Mécanisme identique sur Sparc et x64
-

Sun Cluster and Solaris Containers

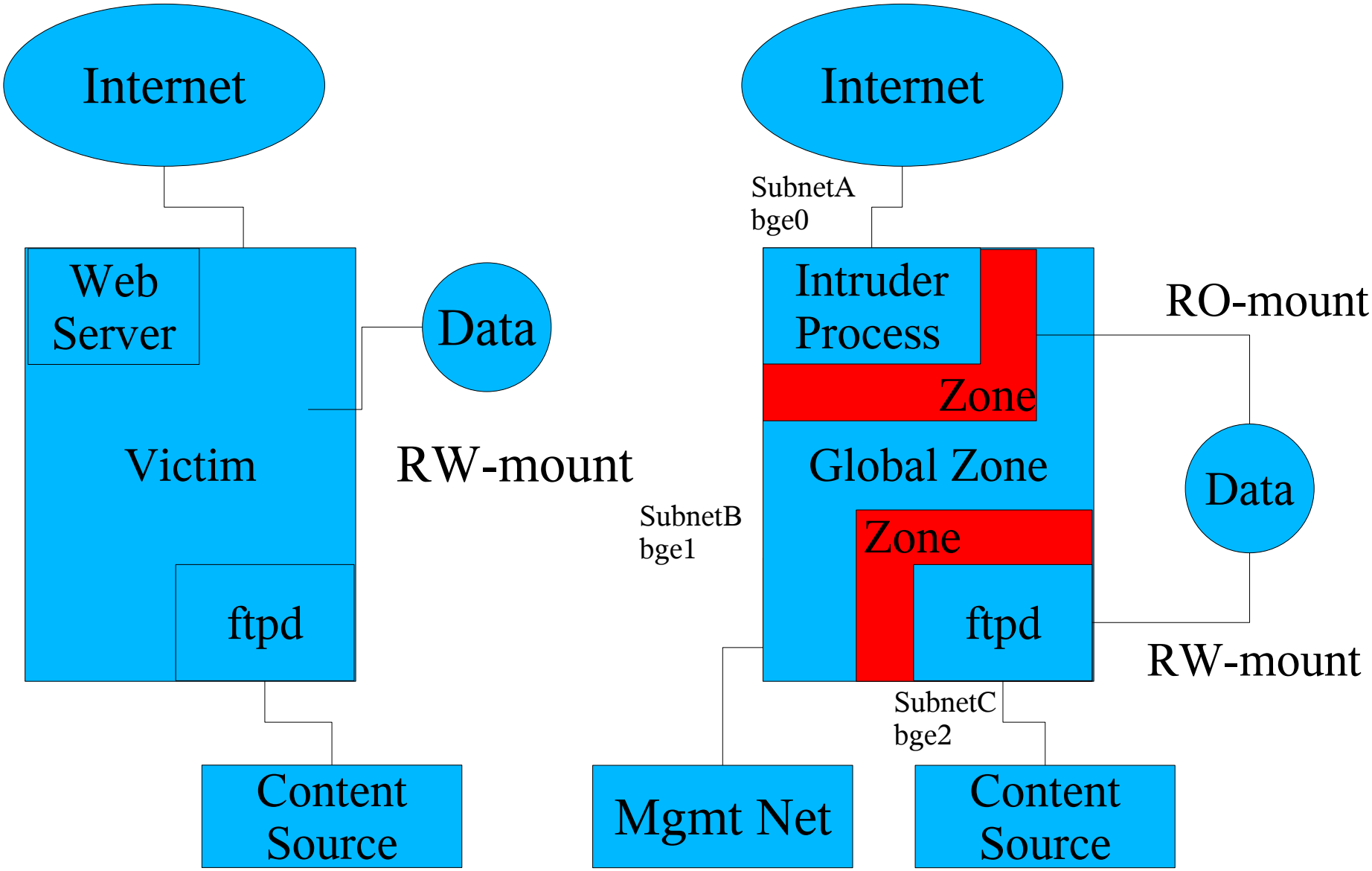


- Failover zones
  - > Mécanisme de cluster peut gérer le failover de zones
- Zone nodes
  - > Une zone peut être vue comme comme noeud d'un cluster

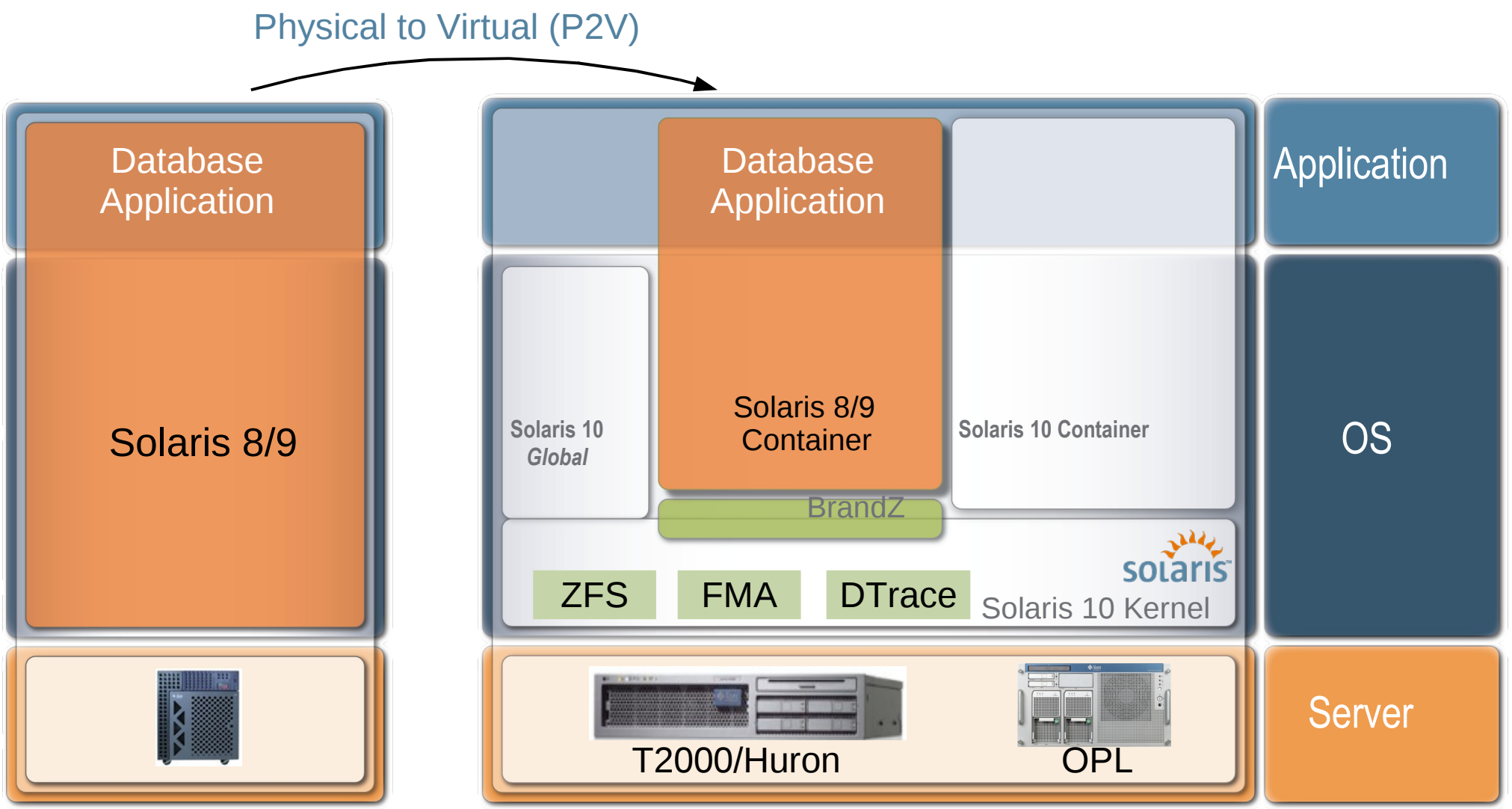
IP Isolation: Multiple IP Instances



Exemple : Secure Web Content

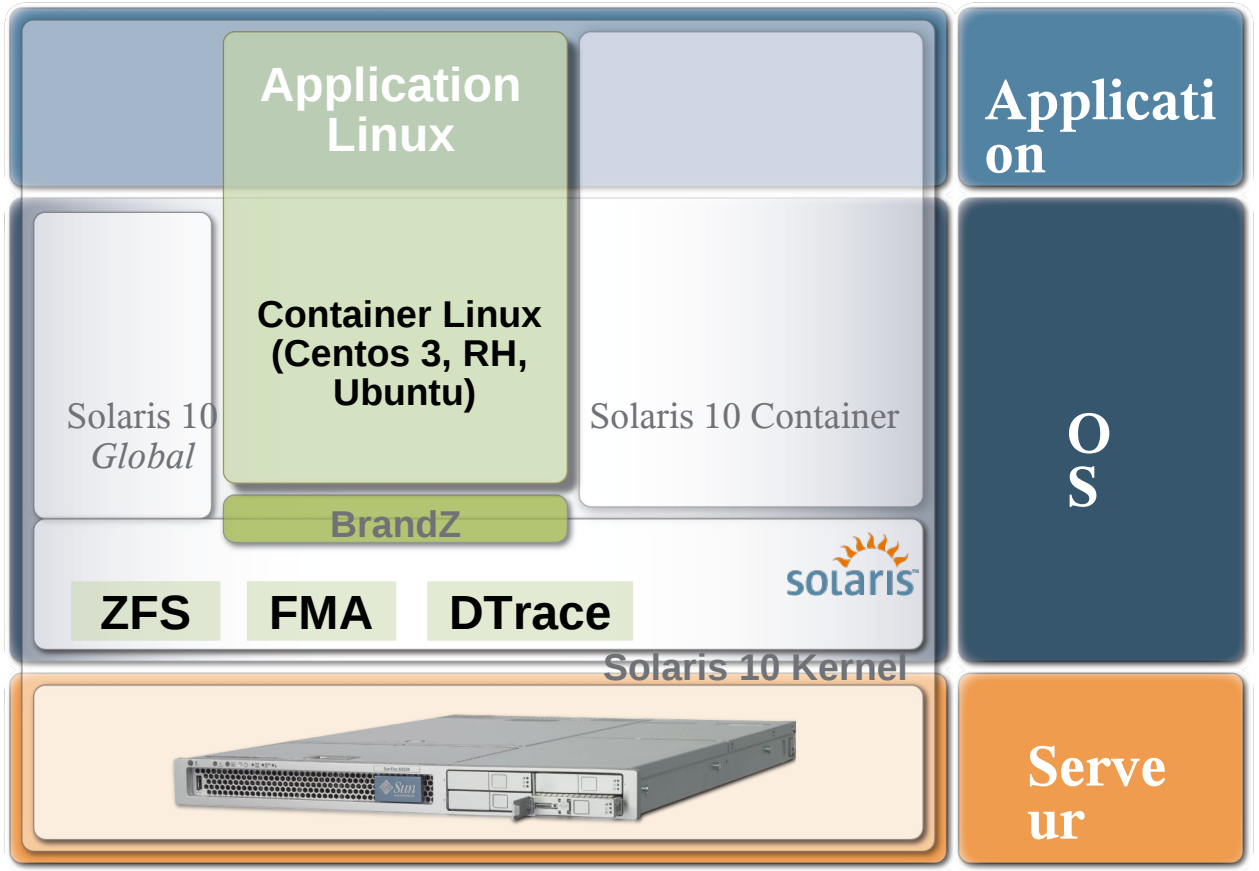


Solaris 8/9 Containers (Sparc)



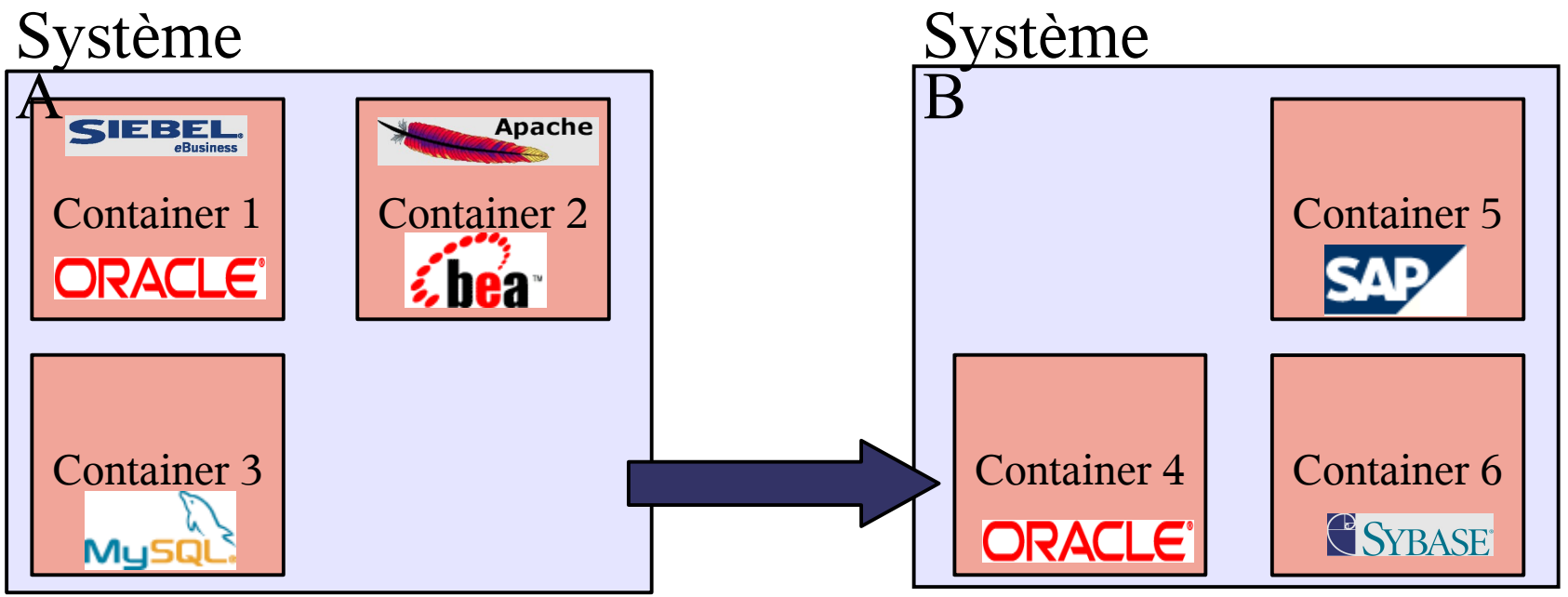
Using Containers to help customers migrate to Solaris 10

Solaris Containers pour les applications Linux (BrandZ)



Containers volants

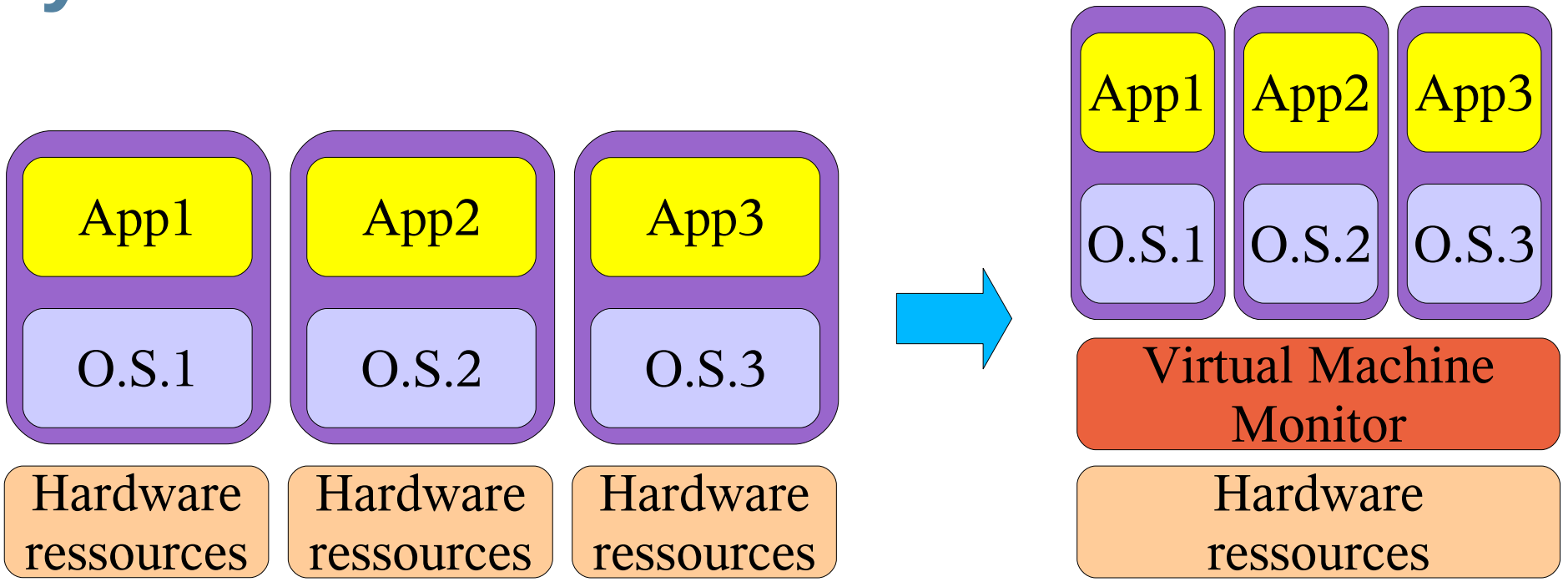
- Plan de secours
- Gestion des ressources et des pics de charge
- Duplication d'environnements



Sun microsystems

Machines Virtuelles

System Virtualization



- Caractéristiques d'un environnement nativement virtualisable
- (Popek & Goldberg / 1974)
  - > Equivalence : App1/OS1 tourne de façon identique
  - > Contrôle complet des ressouces virtualisés par Virtual Machine Monitor (garantir sécurité, isolation, ...)
  - > Efficacité : Plus grand nombre d'instruction possible executées sans intervention de VMM

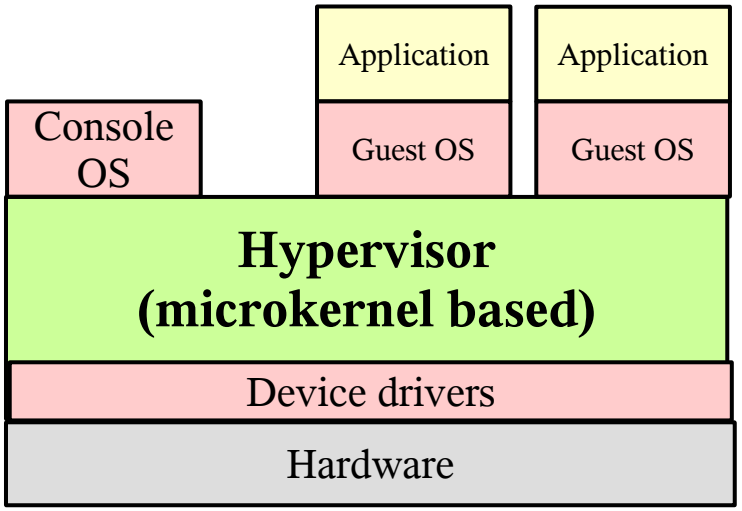
Conditions de la virtualisation

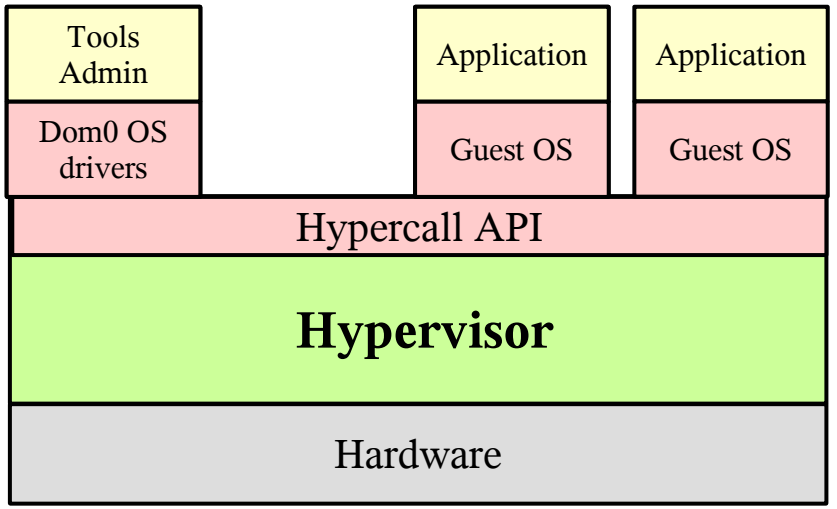
- Machines nativement virtualisables si
  - > Les caractéristiques “Popek & Goldberg” sont réunies
  - > Combinaison des caractéristiques hardware et logiciel VMM
  - > Essentiellement lié au jeu d'instructions du processeur
- Parmi les instructions processeur
  - > Mode privileged : ~system mode vs user mode, I/Os,..
  - > Control sensitive : Changent la configuration des ressources
  - > Behavior sensitive : Comportement dépend de la cfg des ressources.
- Condition essentielle sur le jeu d'instructions :
  - > Toutes les opérations sensibles doivent être protégées (privileged)
  - > **Sur architecture x86 IA-32 : 17 instructions “sensibles” non protégées !**

Types de virtualisation

- Full-virtualisation native
  - > Première machine IBM CP/CMS virtualisable démontrée en 1967
  - > UltraSparc CMT/sun4v via “hyperprivileged mode”
- Pour “contourner” les limites du jeu d'instructions (essentiellement développées sur architectures x86)
  - > Full virtualization par binary translation
    - > Vmware, Sun xVM VirtualBox, QEMU, Microsoft Virtual PC
  - > Hardware assisted full virtualization: instructions complémentaires AMD-V, Intel VT
    - > Vmware, Sun xVM Virtualbox v2 x64, Xen 3.x, Microsoft Hyper-V
  - > Para-virtualisation (famille xen)
  - > Hybride : Full + drivers para-virtualisés

Full vs Para virtualisation





Full virtualisation (binary translation)

Emulation transparente du HW par l'hyperviseur qui intercepte les appels système, détecte et traite les instructions sensibles non protégées.

Guest OS inchangé

Para virtualisation

Hyperviseur et Guest OSes coopèrent via Hypercall API

Guest OS doit être adapté

Fonctionalités inter “guests-OS” possibles

Full vs. Para virtualisation

Virtualisation « complète »

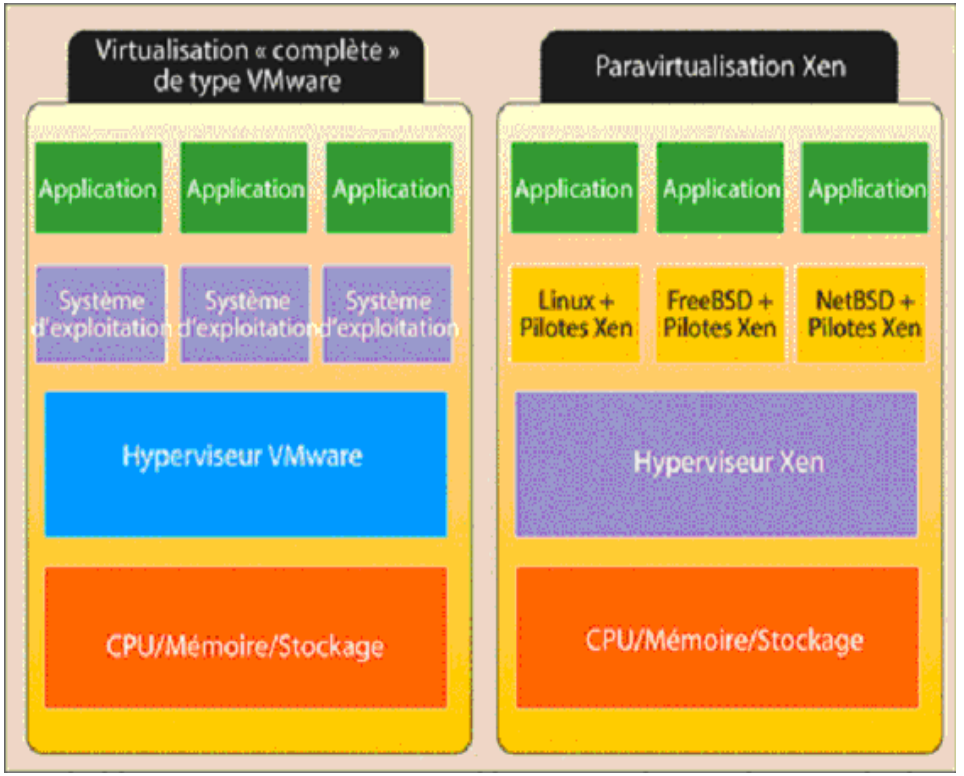
OS «invité» non modifié

Overhead non négligeable du aux mécanismes d'émulation

Para-virtualisation

OS «invité» modifié avec adaptateur, via des pilotes

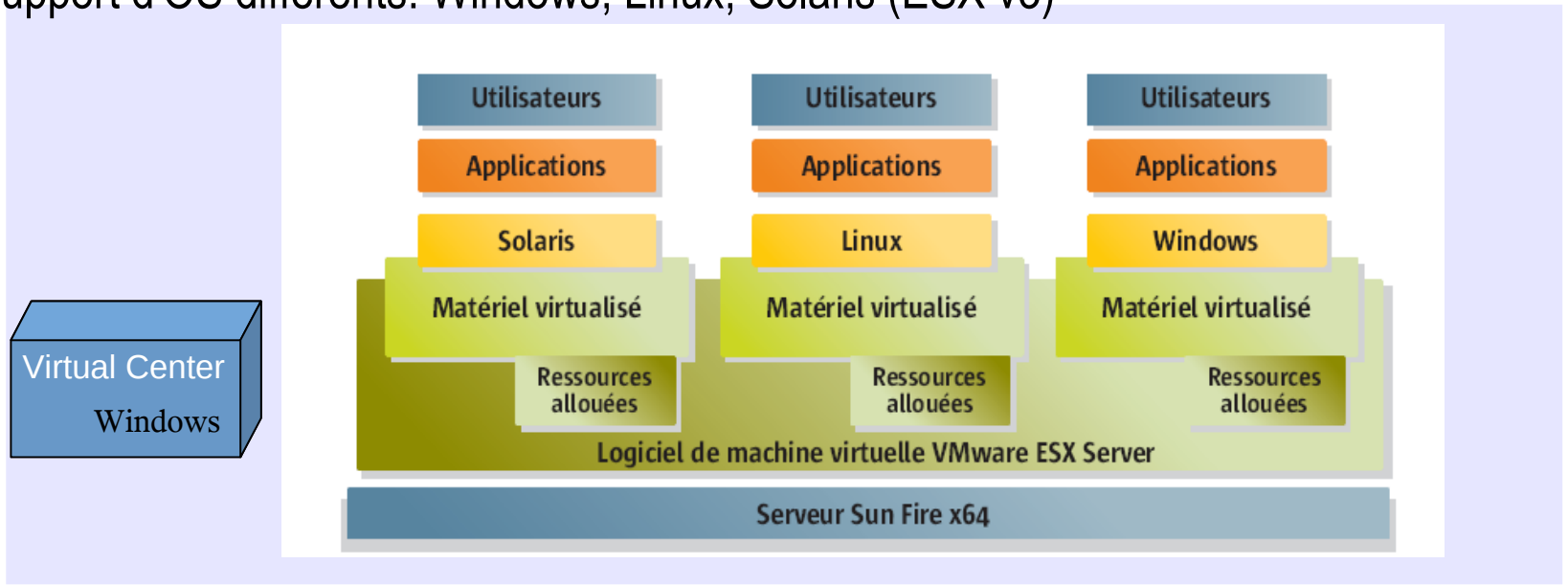
Overhead faible, dialogue VMs avec hyperviseur



VMware

•Certifié sur la gamme x64 de Sun

–Support d'OS différents: Windows, Linux, Solaris (ESX v3)



✓1 Hyperviseur Software: Vmware
✓1 Gestion centralisée avec Virtual Center

Sun xVM

Sun xVM | Portfolio

Open Virtualization for Desktop to Datacenter

Open developer virtualization platform

Sun xVM | VirtualBox

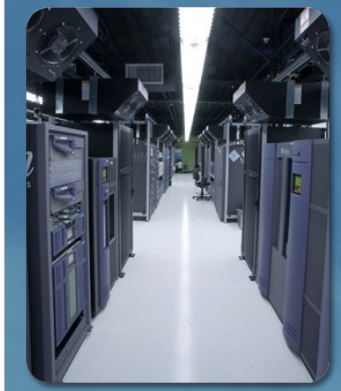
Only VDI with choice: Windows, Open Solaris and Linux delivered securely

Sun xVM | VDI

Sun xVM | Server

Enterprise-class hypervisor

Sun xVM | Ops Center



Manage heterogeneous datacenters

Sun xVM | Server

Enterprise-Class Hypervisor

File Server

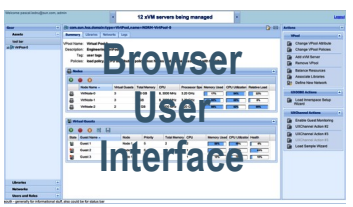
Web Server

Web Server

Mail Server

Mail Server

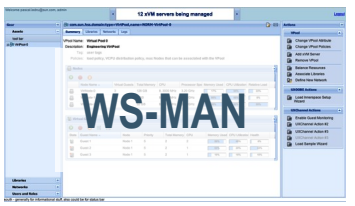
File Server



PV Drivers

Fast Update

Sun xVM | Server



Network

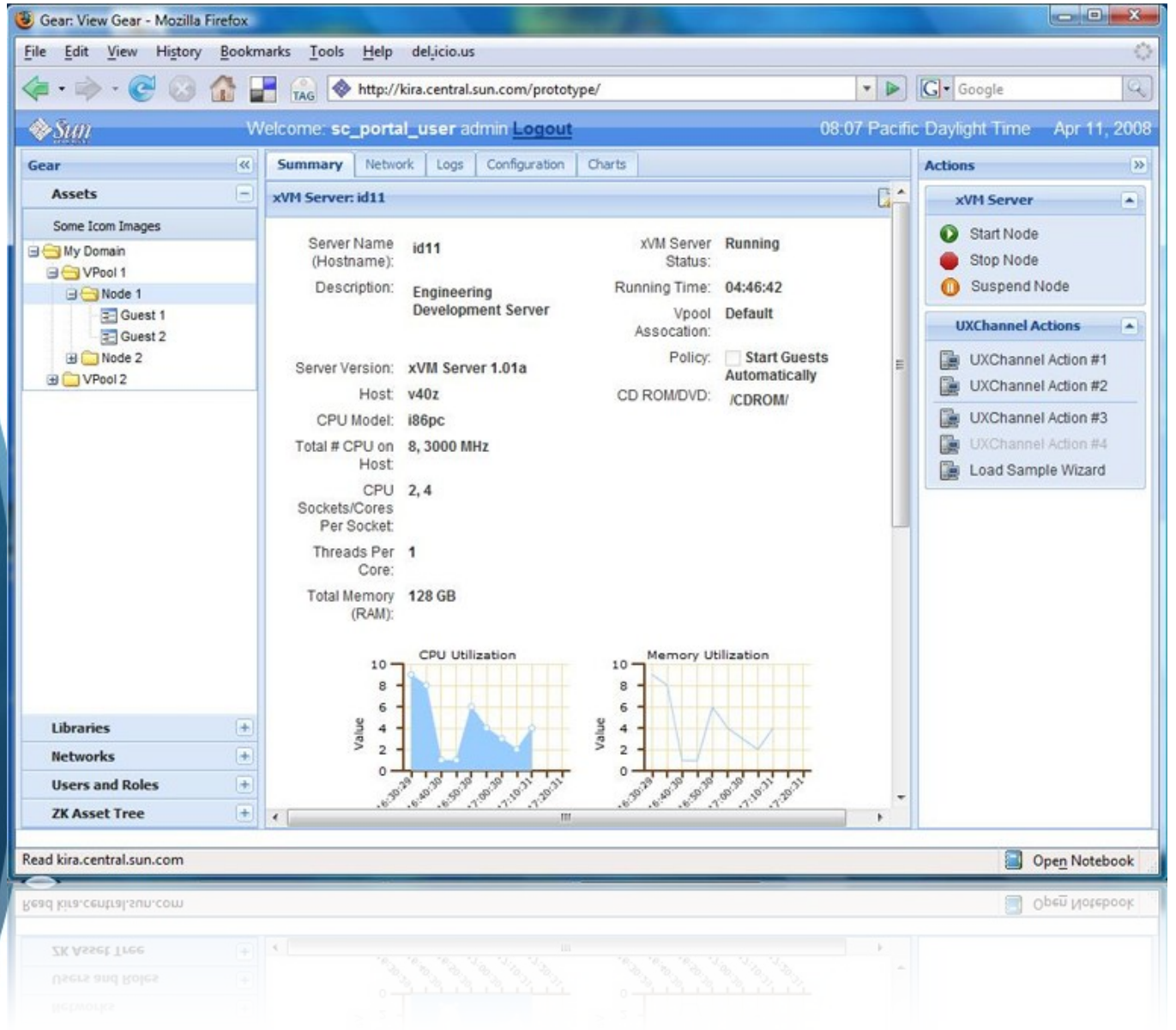
ZFS

VMDK

Self Healing

Sun xVM | Ops Center

Gestion d'environnements hétérogènes



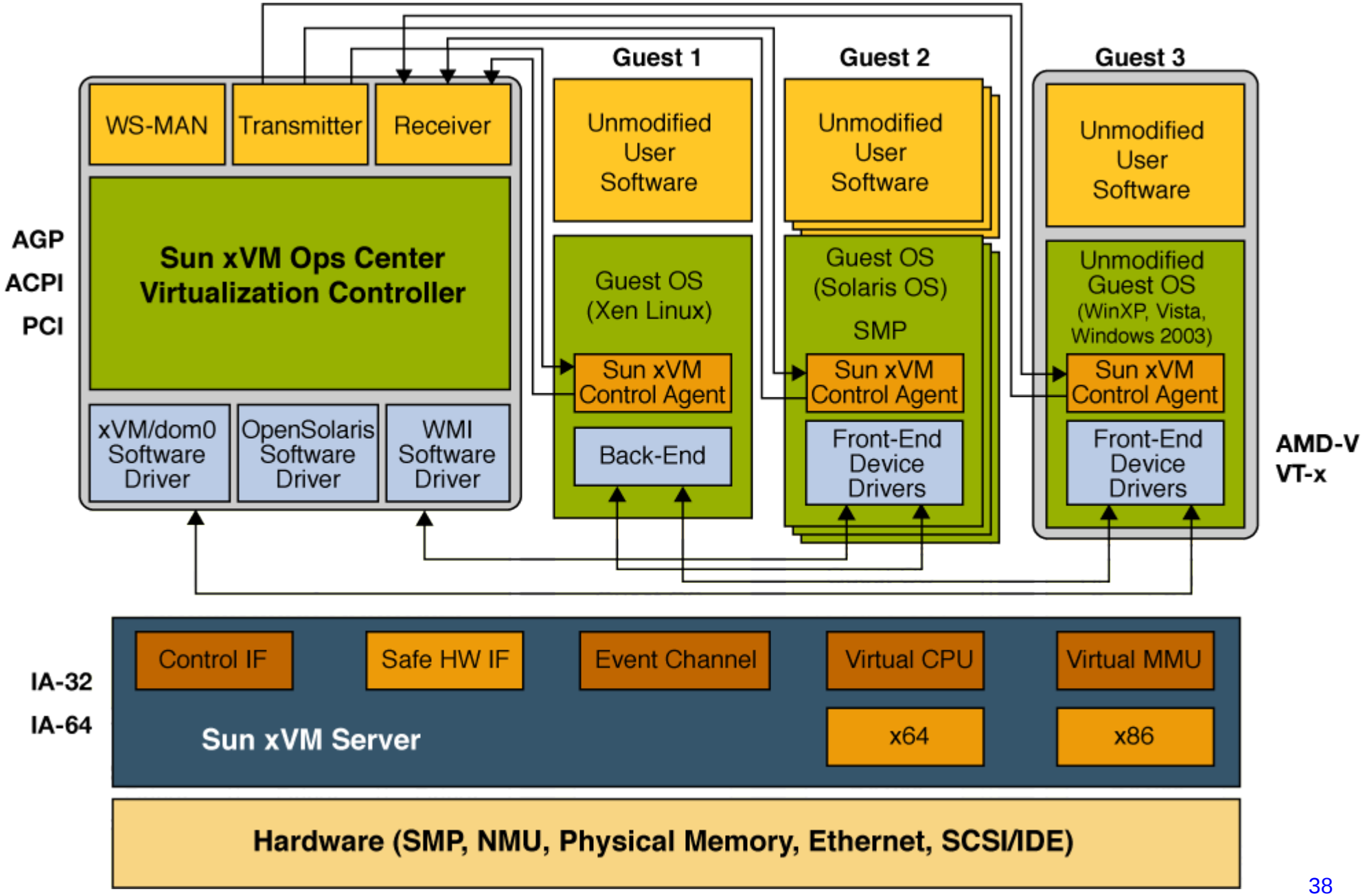
Origine Sun xVM Server

- Xen
  - > Hyperviseur Open source
  - > XenSource : compagnie commerciale distribuant une implémentation Xen (Xen Enterprise) et son support
  - > XenSource racheté par Citrix
- Distributions Xen
  - > Les Linux et Unix majeurs proposent des implémentations dérivées de Xen dans leurs distributions
  - > D'autres societés : Virtual Iron, ...
- Sun Microsystems
  - > XVM Server = 1 des produits de l'offre Sun xVM
  - > xVM Server = implémentation basée sur xen de Sun Microsystems
  - > Solaris comme dom 0 (control domain) et domU (guest OS)

Solaris et xVM
Les avantages de Solaris

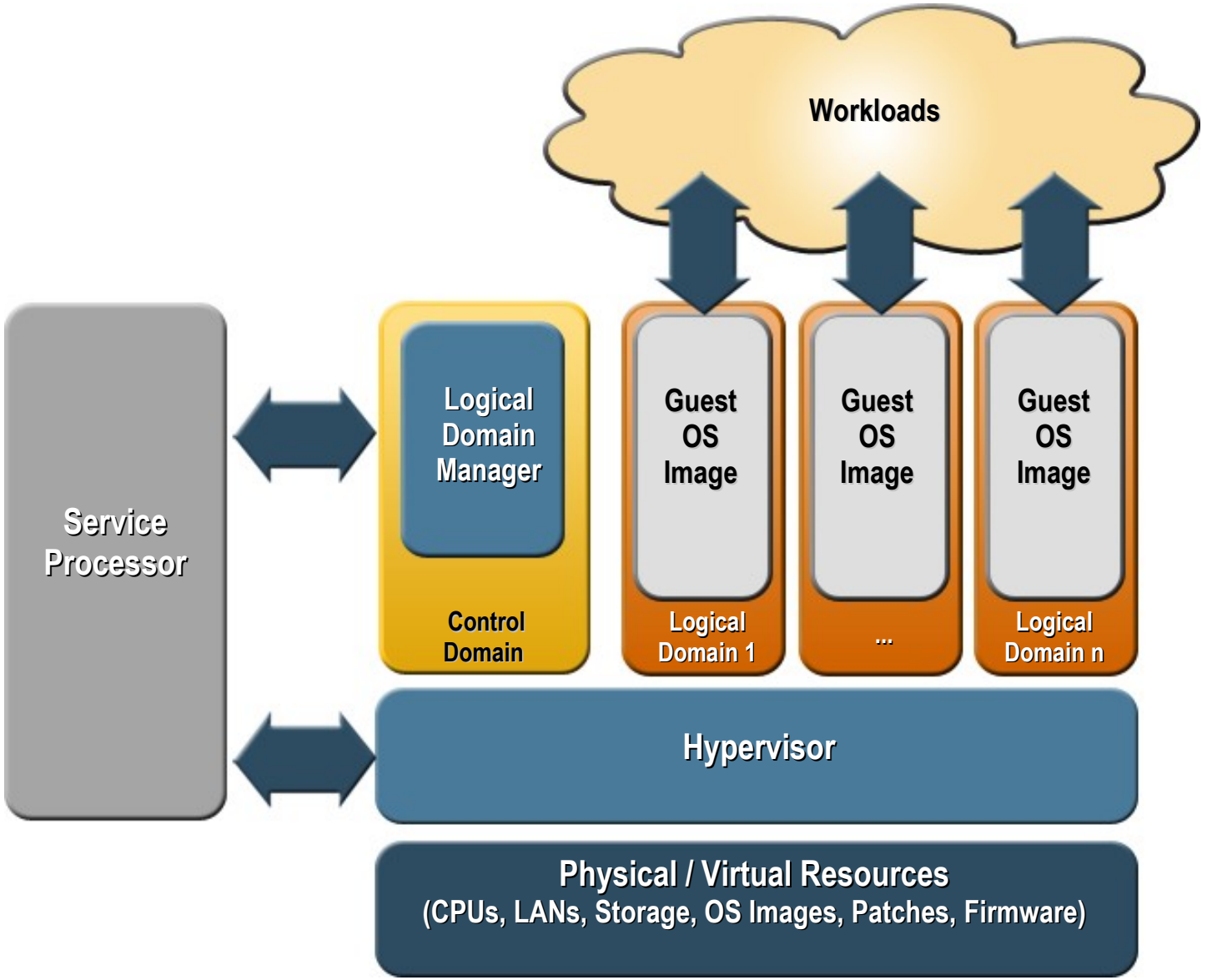
- Predictive Self Healing
  - >Diagnostic en fonctionnement, mises à jour sans reboot
  - >Messages d'erreurs standardisés incluant virtualisation
  - >Mise hors service automatique CPU, mémoire défaillantes
  - >Isolation des fautes drivers et E/S
- Outil dtrace (Dynamic Tracing)
  - >Noyau et hyperviseur observables en fonctionnement
  - >Grand potentiel d'optimisation des performances
- Autres avantages
  - >Virtualisation FS avec ZFS, réseau avec Crossbow
  - >Force de la communauté Opensolaris

Sun xVM Server and xVM Ops Center



Sun Logical Domains

Logical Domains Architecture

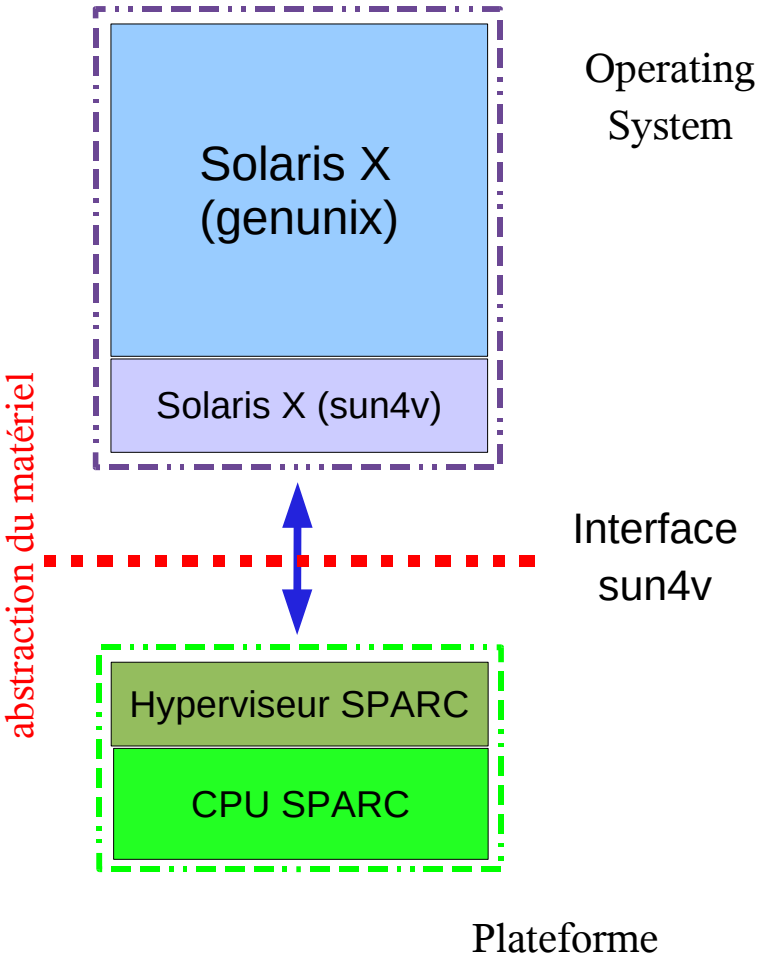


- Protection et isolation en combinant hardware Sparc et firmware de l'hyperviseur
- Hyperviseur light, en PROM, sécurisé :pas de chargement de drivers
- Utilise les capacités de l'OS du control domain et les drivers des services domains
- Modèle générique Domain Channel pourvcréer tous les canaux inter-composants
- Exploite les propriétés des processeurs CMT pour offrir une granularité de partitionnement à la vCPU près (=1 thread

La nouvelle architecture SPARC

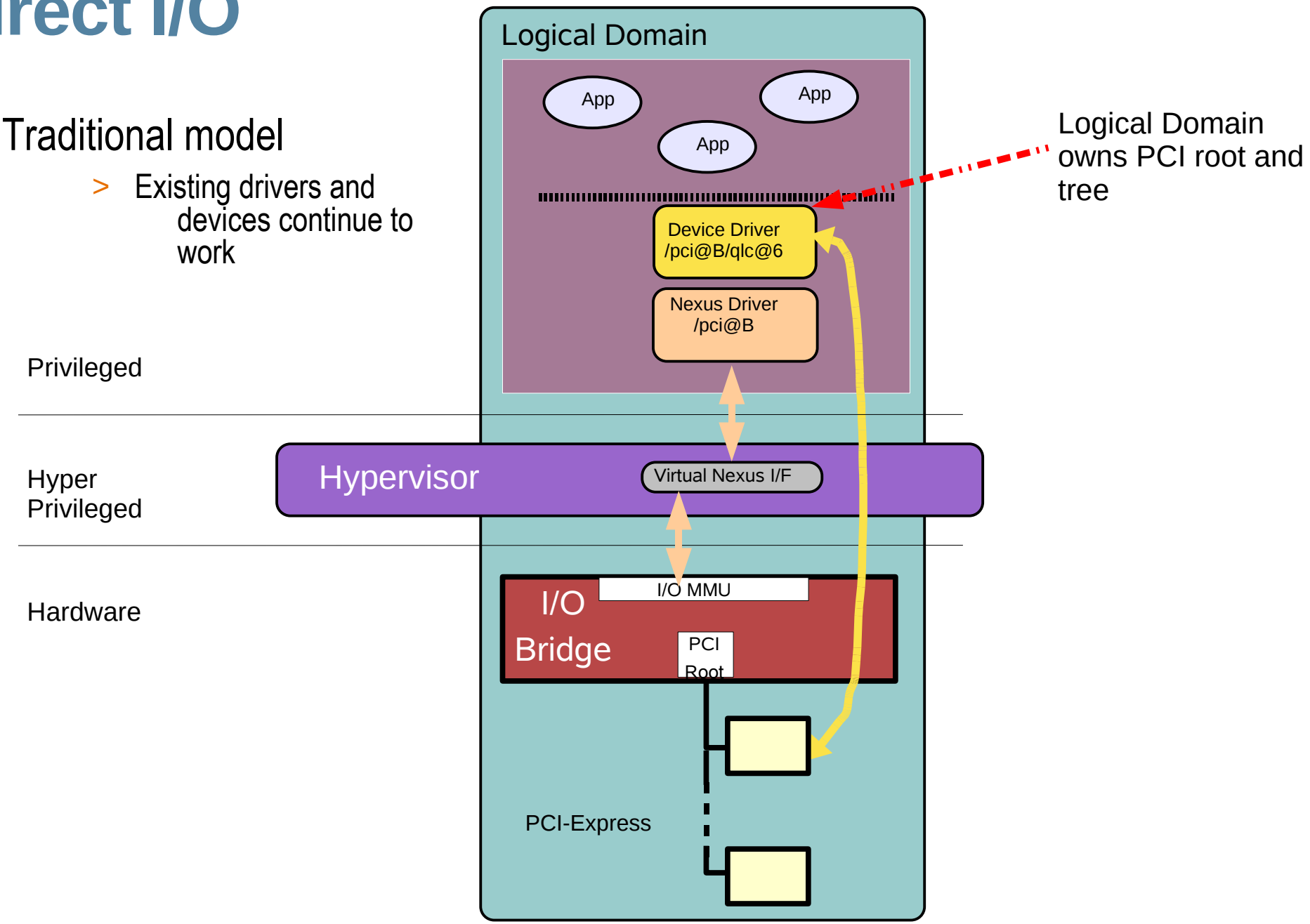
sun4v et Hyperviseur

- Hyperviseur : couche firmware qui implémente l'architecture de la plate-forme
- Solaris a seulement besoin de s'interfacer avec l'hyperviseur
- Introduit avec UltraSPARC T1
- Poursuivi avec UltraSPARC T2(+)
- Existera sur tous les futurs systèmes architecturés sun4v

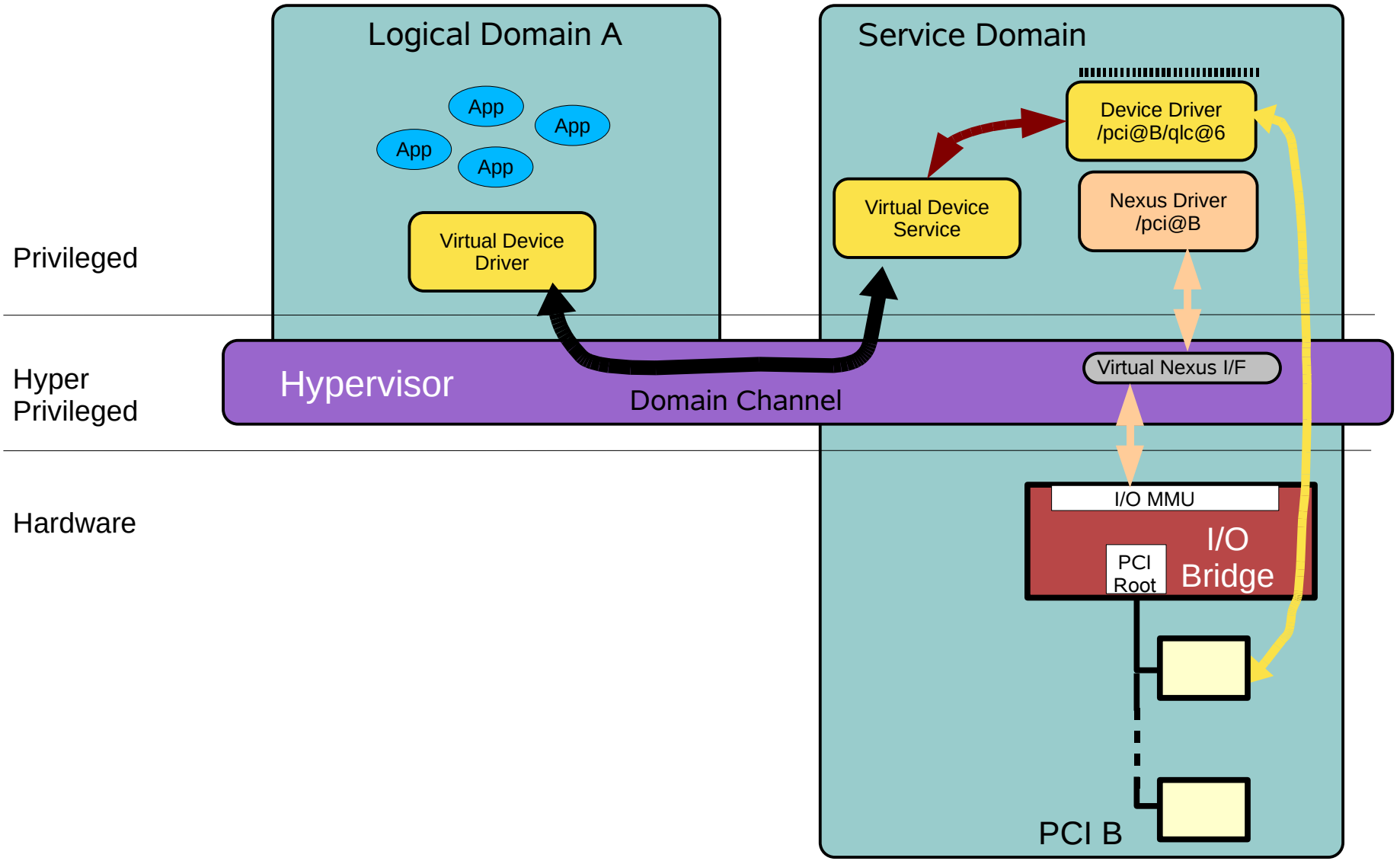


Direct I/O

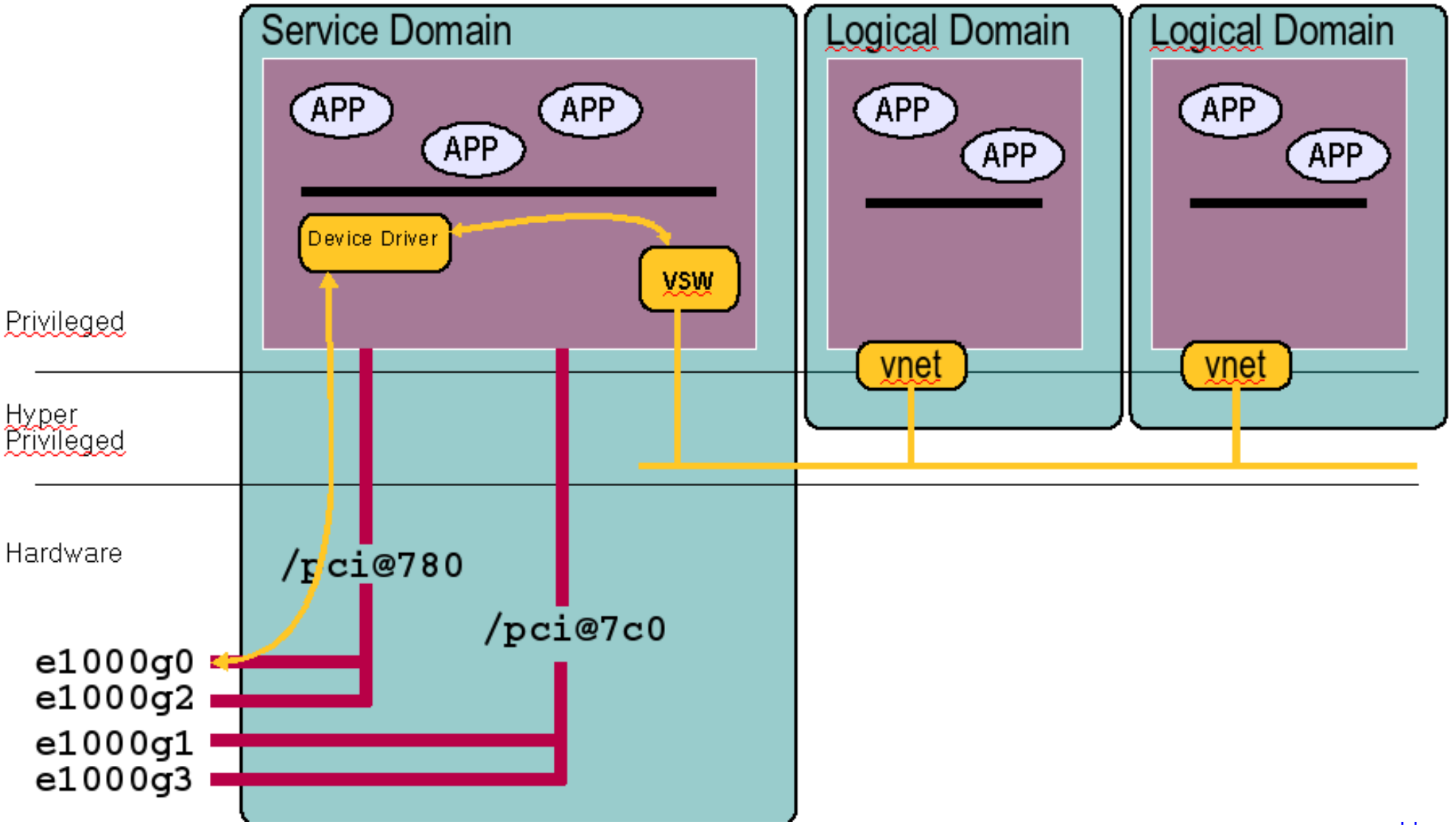
- Traditional model
  - > Existing drivers and devices continue to work



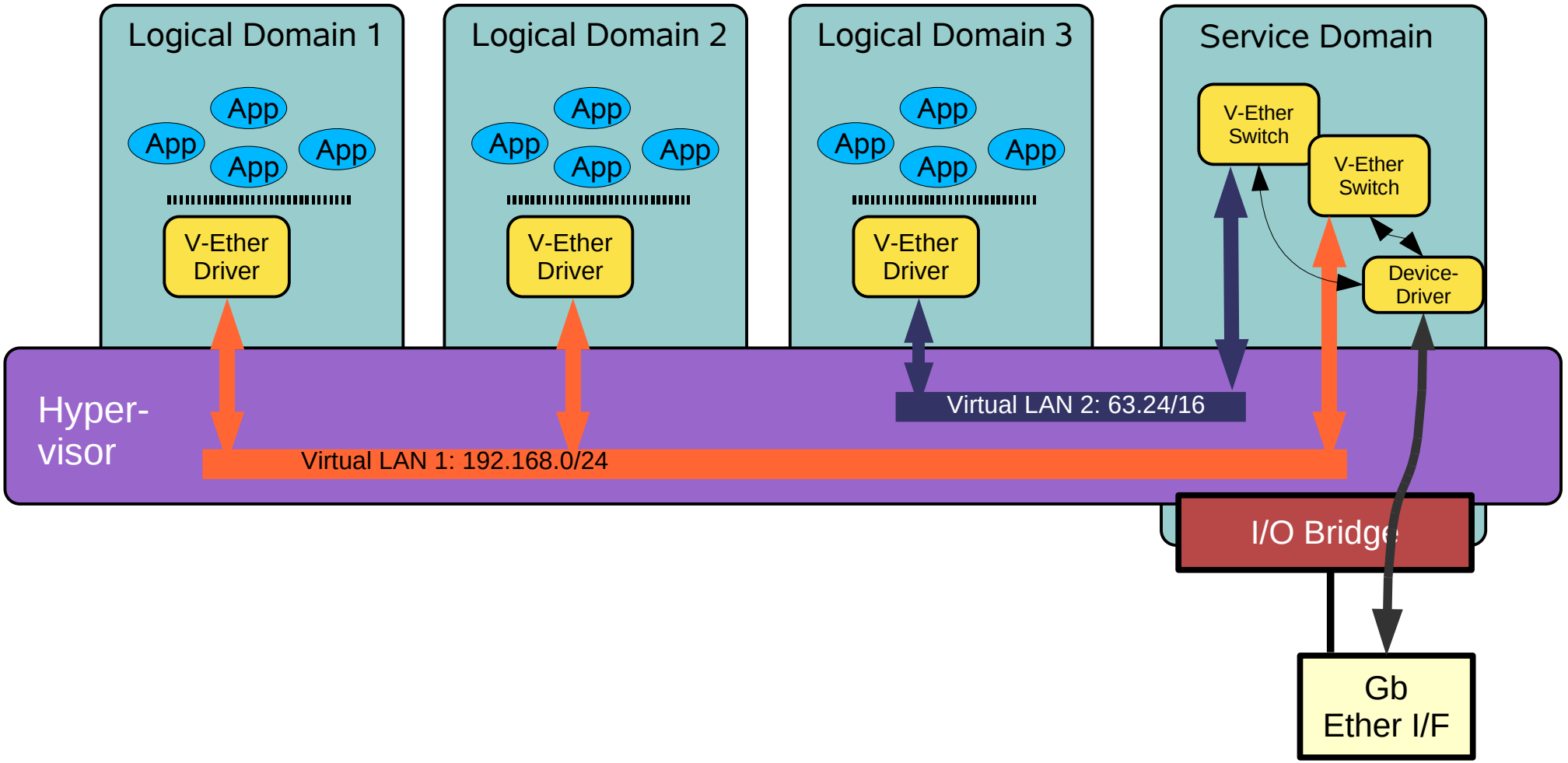
Virtualized I/O



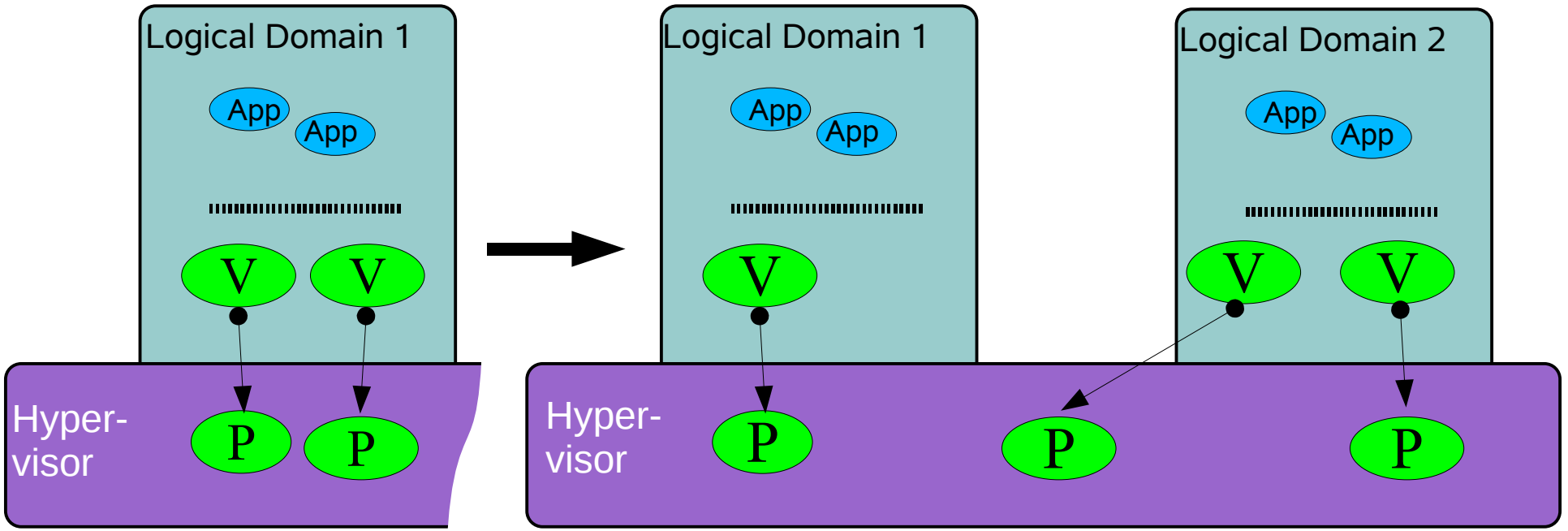
Virtualized I/O



Virtual Ethernet device



vCPU reconfiguration dynamique

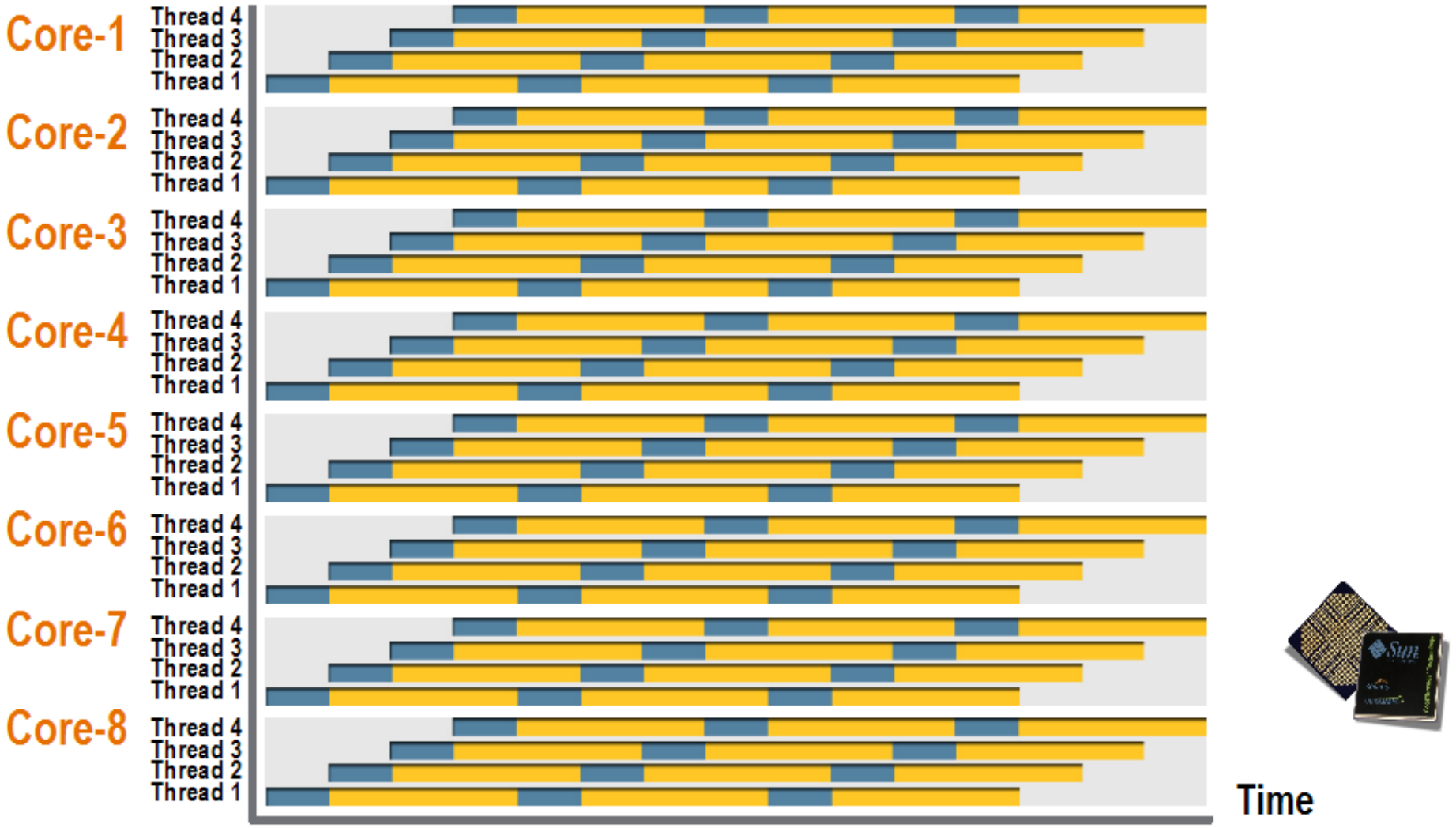


Example command line operations:

```
# ldm remove-vcpu 1 domain1
# ldm add-vcpu 1 domain2
```

Chip Multi-Threading (CMT)

Significantly Higher Throughput from a Team of Multi-threaded Processor Cores



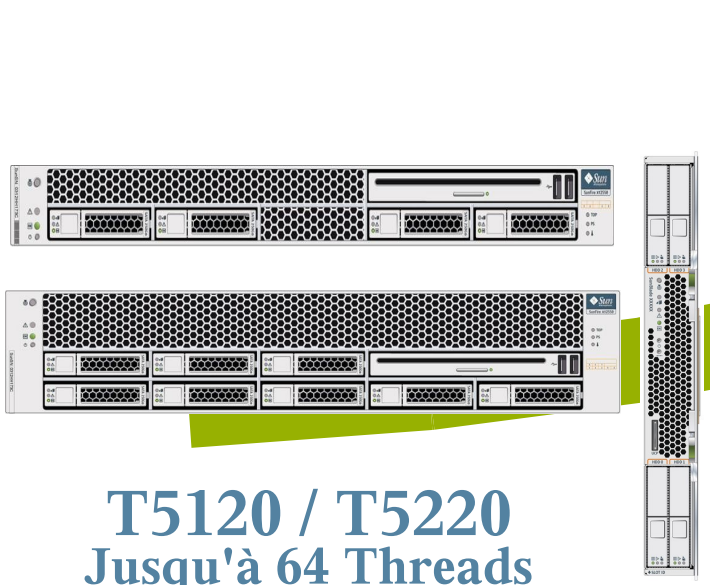
Serveurs CMT à hyperviseur integré

Le nombre de threads est vu comme autant de CPUs virtuelles à affecter aux LDOMs

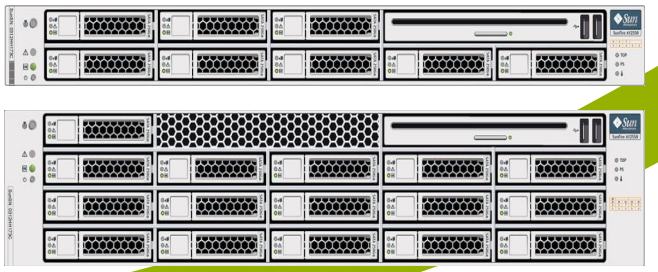
Octobre 2008

Avril 2008

Octobre 2007



T5120 / T5220
Jusqu'à 64 Threads



Maramba T5140 1U / T5240 2U Jusqu'à 128 Threads

Batoka 4RU T5440
Jusqu'à 256 Threads

Blade T6300 (US T1)
Blade T6320 (US T2)

Merci

bernard.pierre@sun.com