Unix Virtualization & Partitioning

Key Qualities of Unix Virtualization & Partitioning Offerings for the Enterprise

Armin M. Warda

Mail <at> **Armin** <minus> **Warda** <dot> **De**

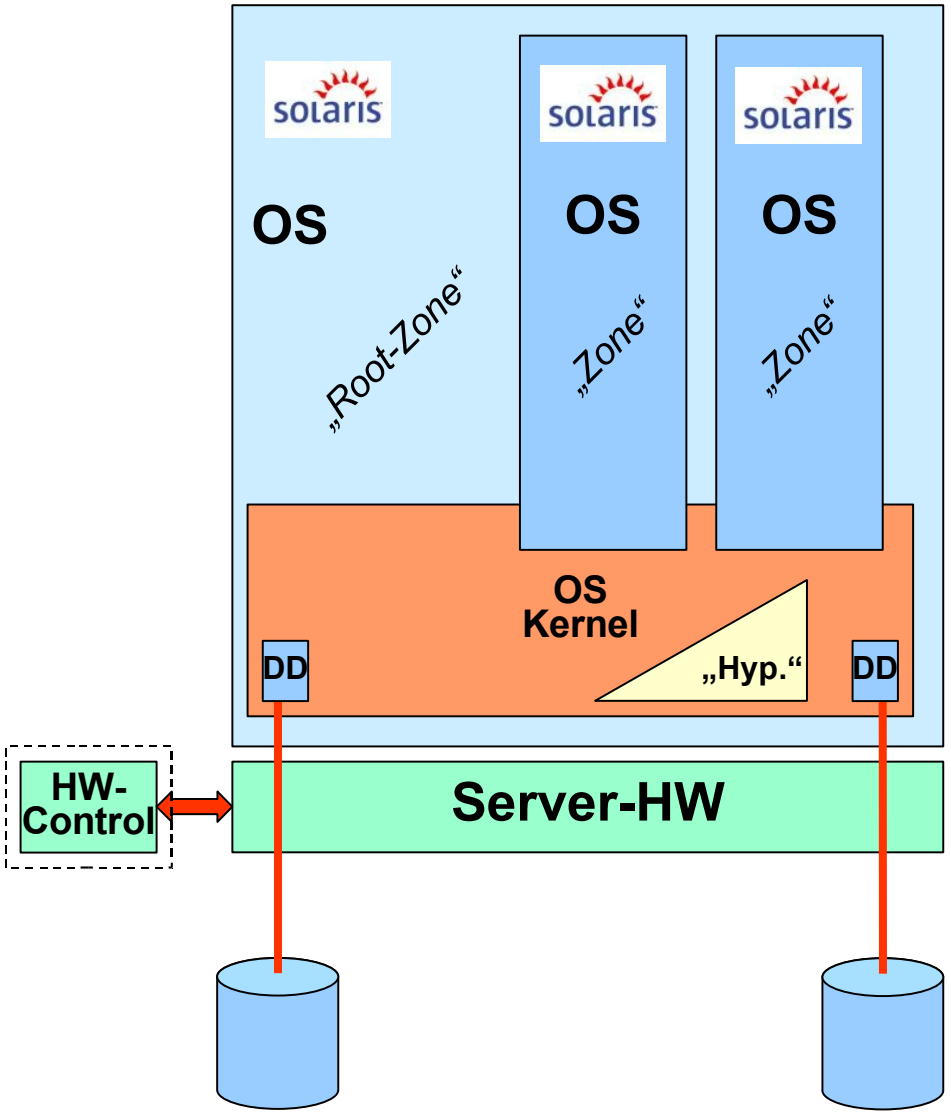
http://armin-warda.de

v0.4 03/2006

Unix Virtualization & Partitioning Concepts

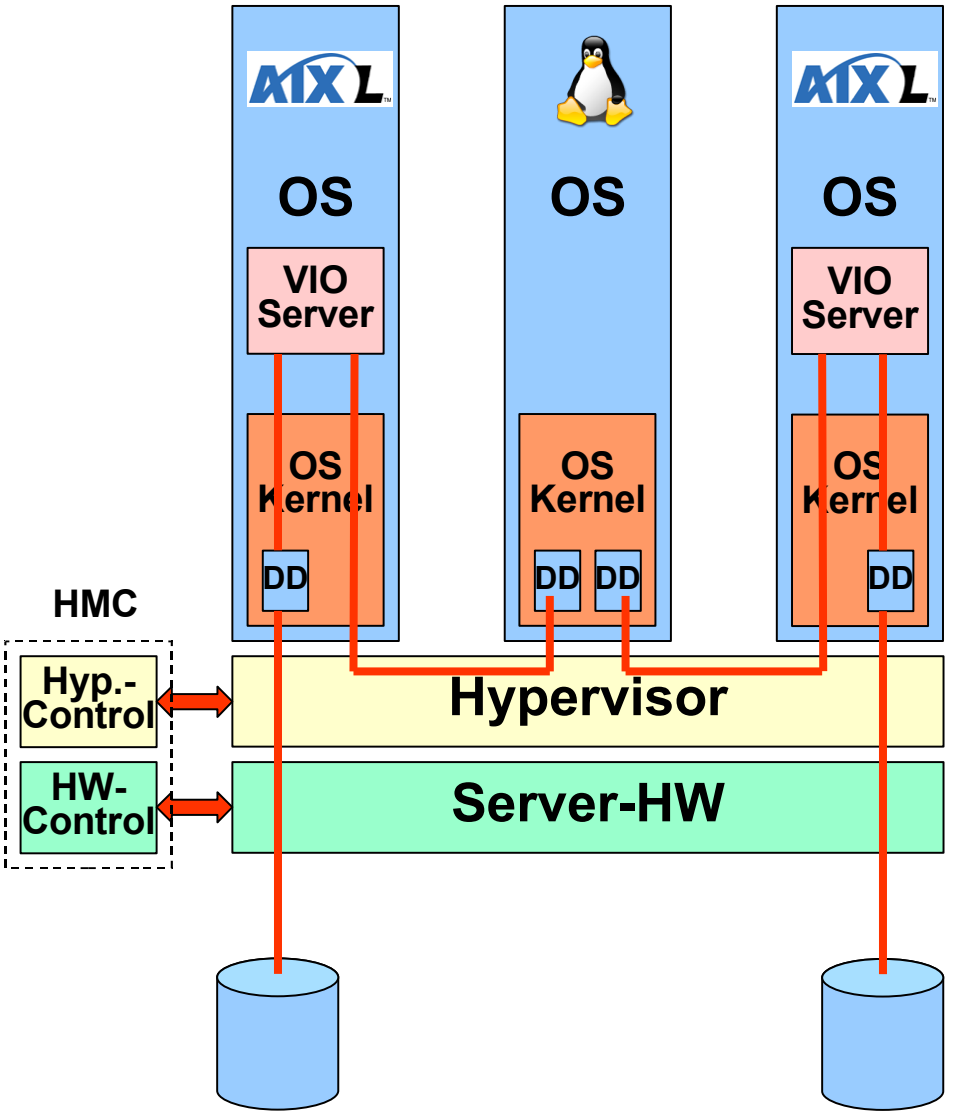
- OS-Virtualization: *Solaris Zones, Virtuozzo*
- Para-Virtualization: *IBM p5 LPAR, Xen*
- Full-Virtualization: *VMware, MS Virtual Server*
- HW-Partitioning: *FSC XPAR, SUN Domains*
- No Virtualization: *Single Unix-System* with many apps
- No Virtualization: *Multiple Unix-Systems* each with a single app only

OS-Virtualization: Solaris Zones



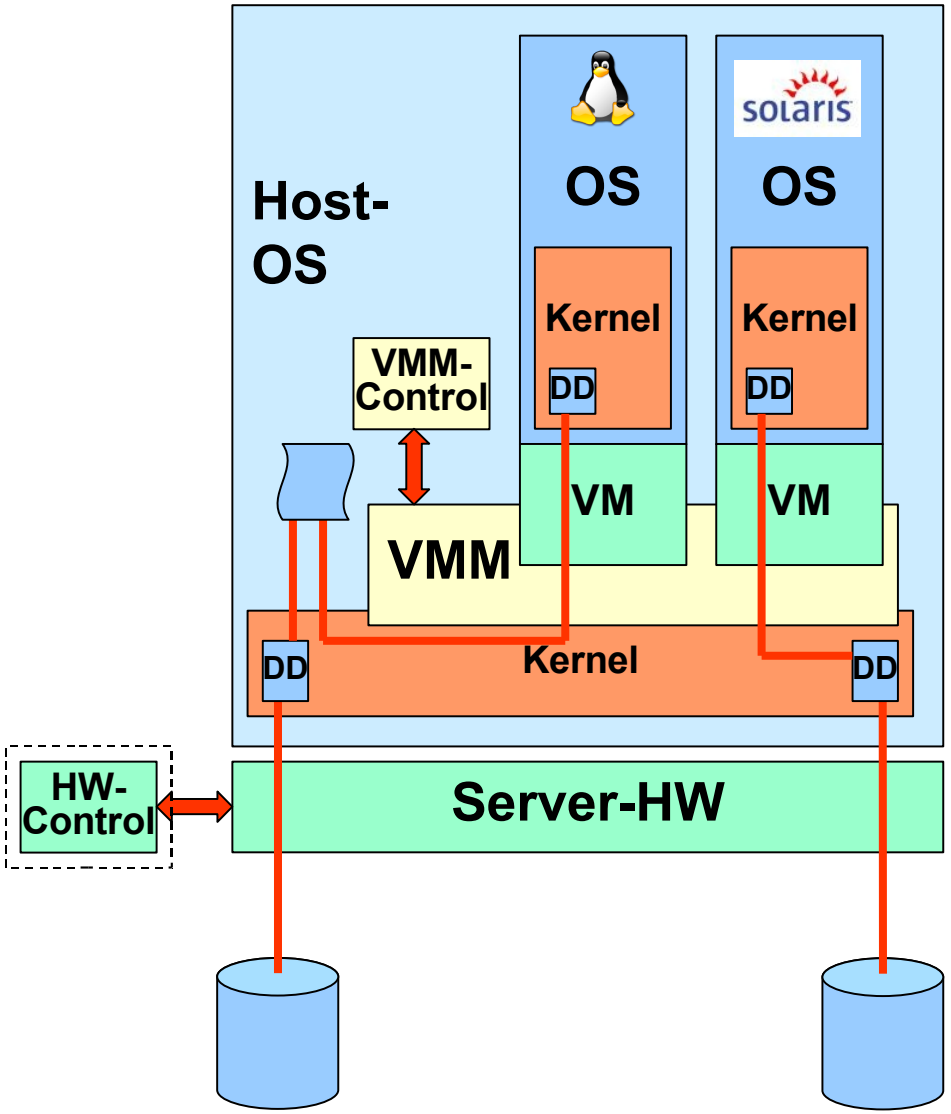
- No explicit VMM or Hypervisor
- Guest-OS share OS-Kernel with Host-OS
- Host OS-Kernel modified to support multiple name-spaces (e.g. chroot, IPC-objects, ps)

Para-Virtualization: IBM p5 LPARs



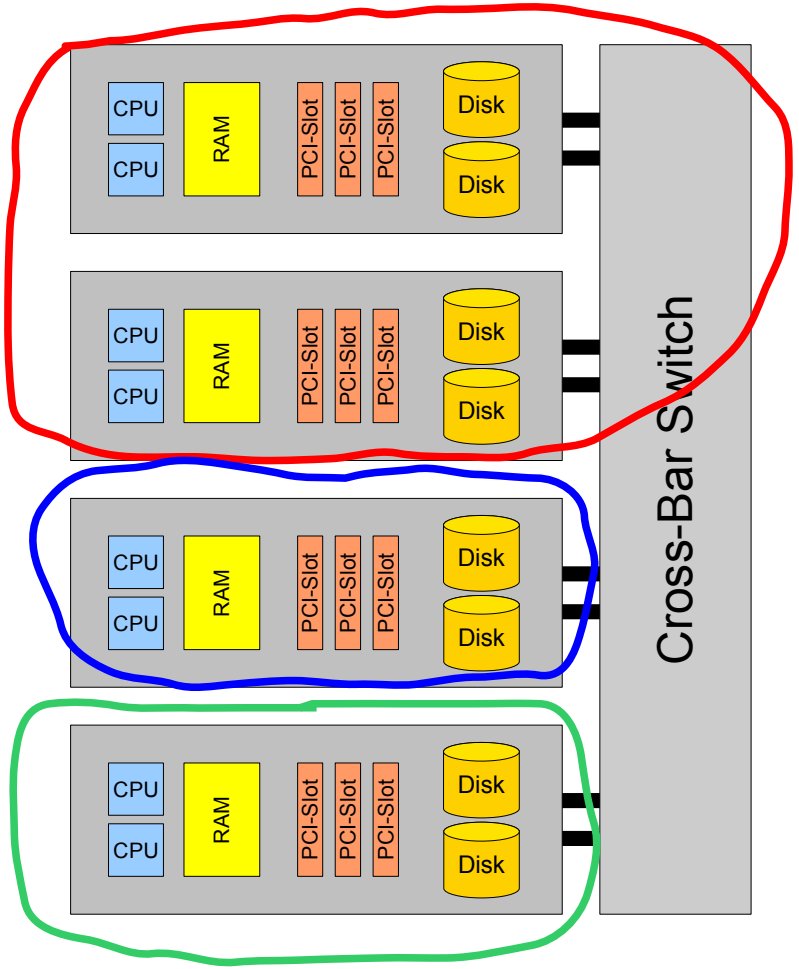
- OS-Kernel source codes are modified to execute hypervisor calls instead of privileged instructions
- No Host-OS, but Hypervisor is kind of „Mini-OS“
- LPAR can use regular Device Drivers to access physical devices (Partitioning), or
- LPAR can use Device Drivers for virtual Devices, which are virtualized by Virtual-I/O-Server

Full-Virtualization: VMware



- Virtual Machine Monitor (VMM) runs on (or besides) Host-OS and provides multiple Virtual Machines (VM)
- Different Guest-OS run in VMs with unmodified OS-Kernel and use regular DDs
- Guest-OS may use special DDs, e.g. for accelerated graphics
- Control of VMM from Host-OS
- Privileged instructions are replaced by dynamic code translation or trapped by CPU-HW (Intel Vanderpool, AMD Pacifica) and emulated by VMM

HW-Partitioning: XPAR, Domains



FSC XPARs:

- System-Boards (XSB) with
  - 2 CPU cores
  - RAM
  - PCI-Slots
  - Disks

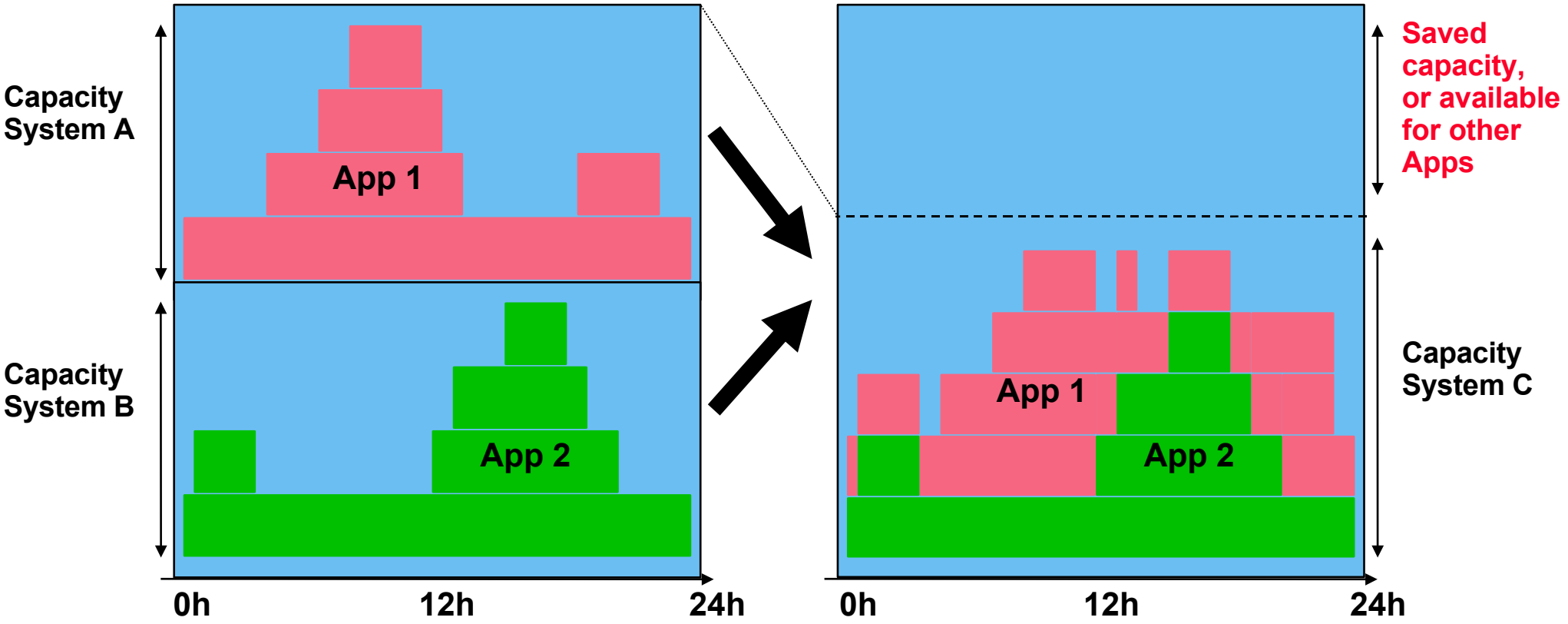
Sun Domains:

- Processing-Boards with
  - 8 CPU cores
  - RAM
- I/O-Boards with
  - PCI-Slots
  - Disks

Key Virtualization & Partitioning Qualities

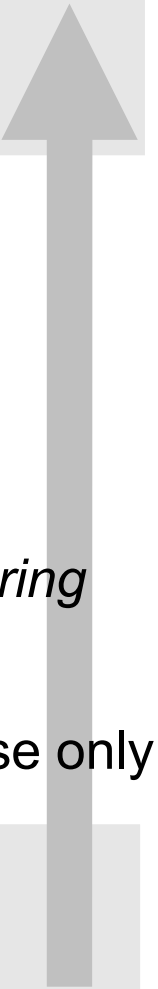
- Resource-Sharing
- Isolation
- Efficiency
- Rapid Deployment

Why Resource-Sharing?



Which offers best Resource-Sharing?

1. Single Unix-System - *everything shared*
2. OS-Virtualization - *RAM-sharing*
3. Para-Virtualization - *cede & confer CPU cycles*
4. Full-Virtualization - *fine-grained, CPU- & I/O-Sharing*
5. HW-Partitioning - coarse-grained, alternating use only
6. Multiple Unix-Systems - *nothing shared*

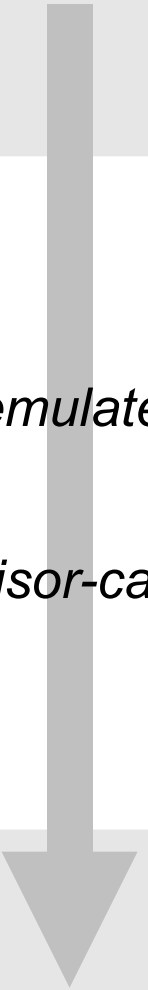


Why is Isolation so important?

- different networks (production, test, dev,..)
- different system administrators
- different Unix derivates
  - different releases of same Unix derivate
    - independent release-cycles
- fault-isolation and -containment
- no resource monopolization

Which offers best Isolation?

1. Multiple Unix-Systems - *nothing shared*
2. HW-Partitioning - *almost nothing shared*
3. Full-Virtualization - *no hypervisor-calls: trap & emulate*
4. Para-Virtualization - *cooperation through hypervisor-calls*
5. OS-Virtualization - *shared kernel, libs*
6. Single Unix-System - *almost everything shared*



Why is Efficiency important?

- everybody wants good performance
- and low overhead

  (= resources you pay for, but that do not work for you...)
- nobody likes waste of resources *(maybe except for HW-vendors ;-)*

What kind of Overhead?

1. overhead of unused resources,
   because system can't be sized optimally
   - fine-grained resource sharing => low overhead

2. overhead of the
   virtualization/partitioning mechanism:
   - hypervisor uses some CPU-cycles
   - more context-switches
   - cache-inefficiencies by CPU-sharing

Which offers lowest Virtualization-Overhead?

1. Single Unix-System - *no virtualization, no overhead*
2. Multiple Unix-Systems - *no virtualization, no overhead*
3. HW-Partitioning - *no virtualization, no overhead*
4. OS-Virtualization - *almost no overhead*
5. Para-Virtualization - *low overhead (HCALL, RDMA)*
6. Full-Virtualization - *high overhead (trap & emulate)*

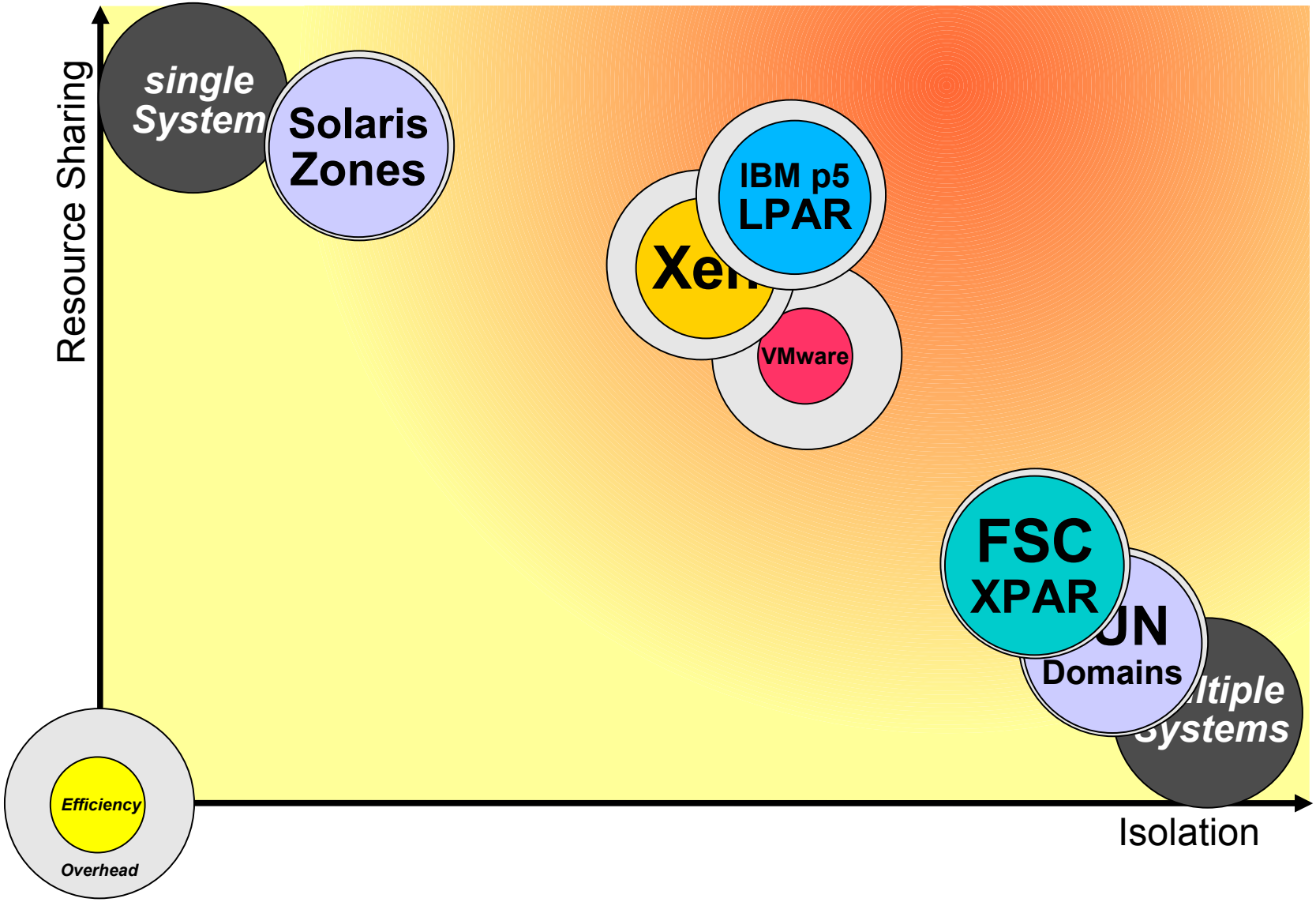
Why is Rapid Deployment important?

- systems on-demand
- time-to-market for new services
- temporarily needed systems
- systems not needed continuously,
  - e.g. only for some hours/day, days/week,..

Which offers fastest Rapid Deployment?

1. Single Unix-System - *need some capacity*
2. OS-Virtualization - *need some capacity & setup*
3. Full-Virtualization - *need some capacity & installation*
4. Para-Virtualization - *need some capacity & installation*
5. HW-Partitioning - *need system-board & installation*
6. Multiple Unix-Systems - *need system & installation*

Resource-Sharing, Isolation & Efficiency



Other Qualities relevant to Virtualization

- Scalability
- UMA'ness (cf. NUMA)
- System-Management Tools
- Security

Other Virtualization & Partitioning Concepts

- Clustered Para-Virtualization: *Virtual Iron (VFe)*
- Full-IA-Virtualization: *Bochs, QEMU, Hercules*

But these are not enterprise-ready,

or just intended for cross-plattform development.