Deflating the hype:
Embedded Virtualization in 3 steps

Klaas van Gend

MontaVista Software LLC

For Embedded Linux Conference Europe 2010, Cambridge

Agenda

- Why “multicore” made the topic more relevant
- Partitioning
- Virtualization
- Containers
- The Three Steps
- Summary

Asymmetric Multiprocessing

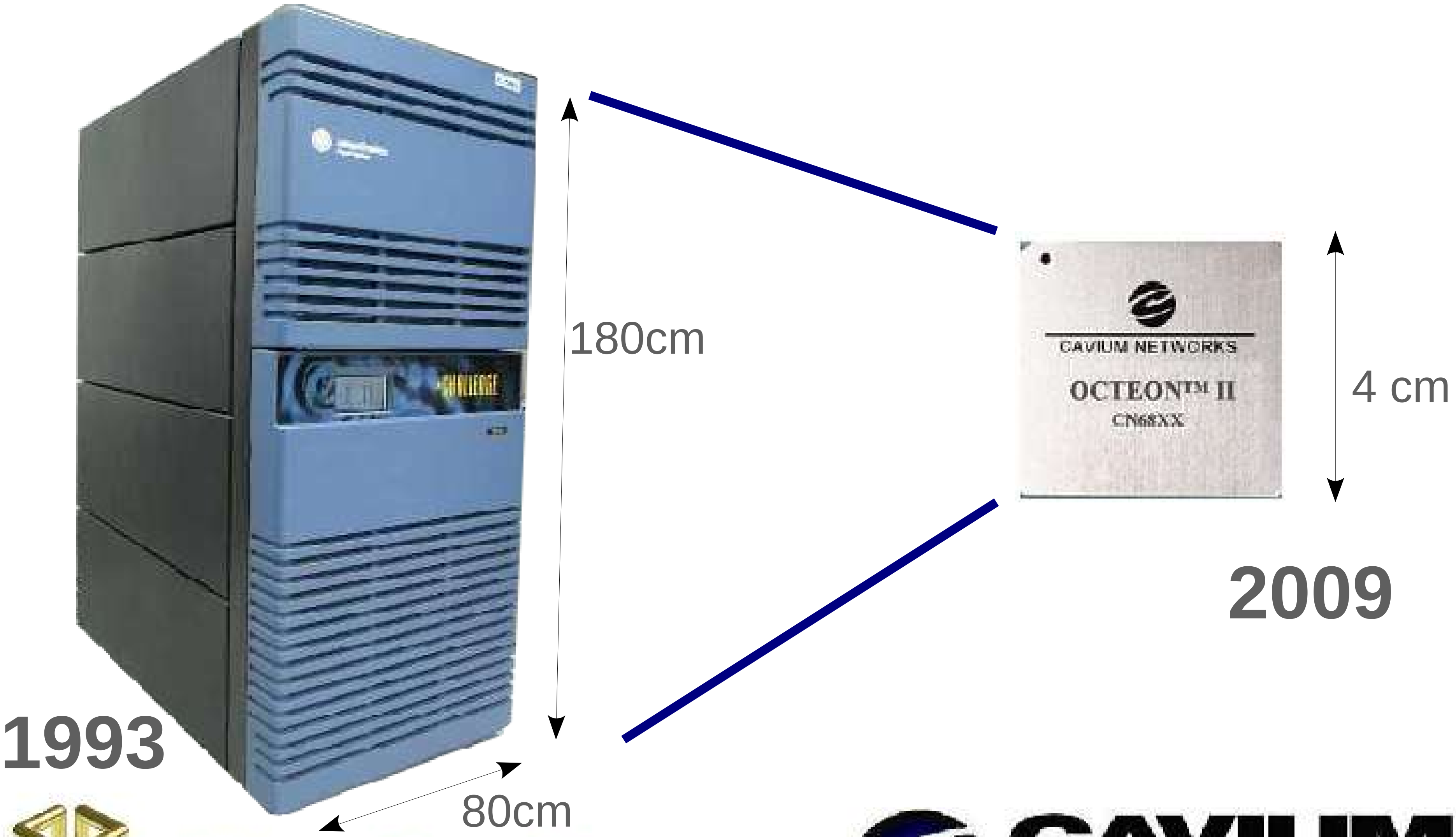
Commodore AMIGA

1985

TEXAS INSTRUMENTS

200x

Symmetric Multiprocessing



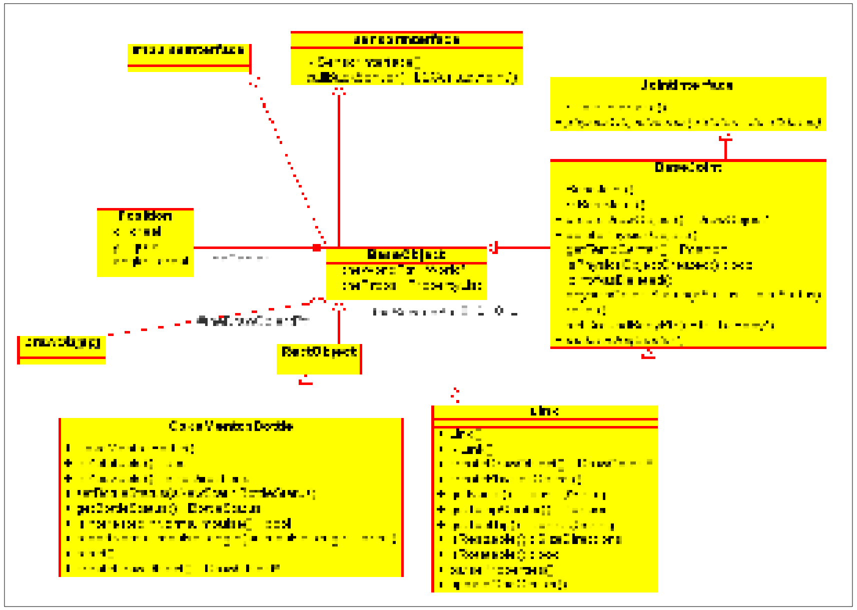
1993

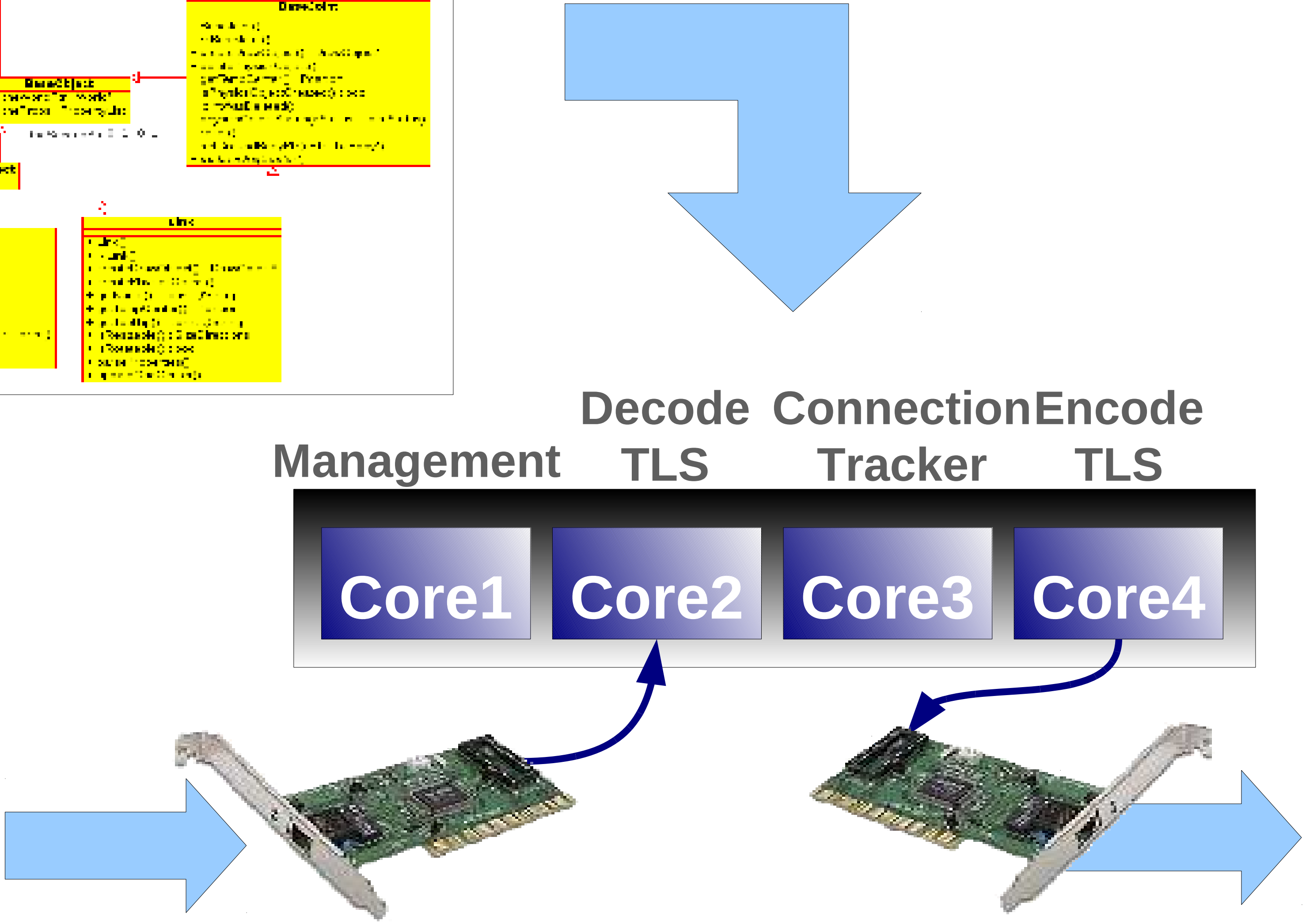
2009

SiliconGraphics

CAVIUM NETWORKS

System Architecture & Partitioning





Management

Decode TLS

Connection Tracker

Encode TLS

Core1

Core2

Core3

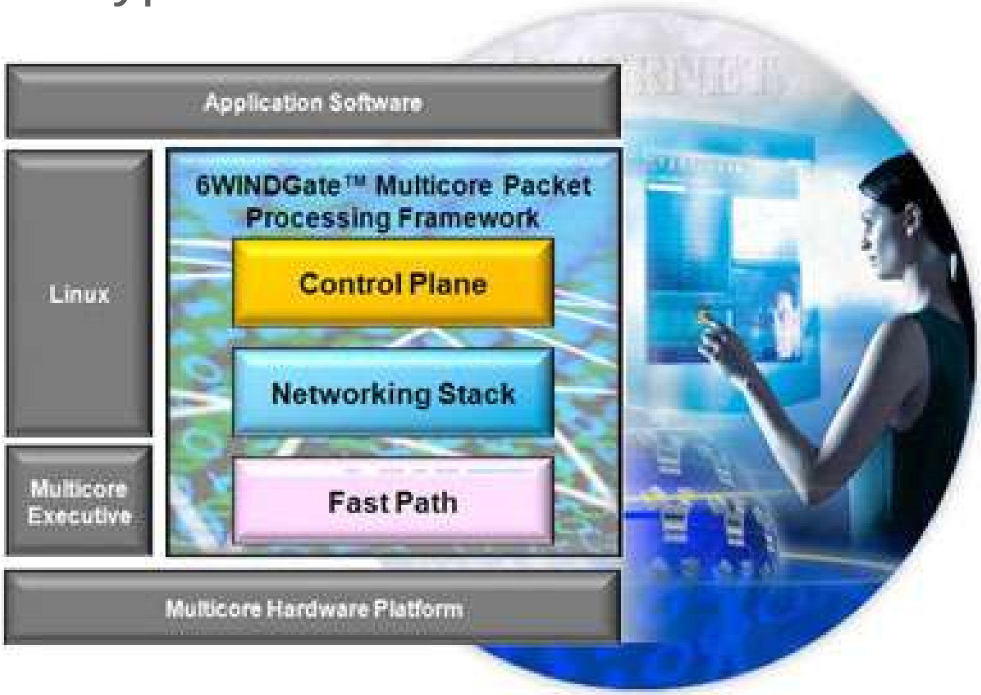
Core4

RT and Throughput considerations

- **PREEMPT_RT makes Linux real time**
  - Depending on HW and requirements
- **RT Microkernels + Linux**
  - hybrid approach, 2 different environments
  - Merging the bad properties of both
- **But Linux has its drawbacks**
  - Complete IP stack
    - not optimized for specific use cases
  - Process Migration amongst CPUs
  - BKL and scheduling overhead

Partitioning & AMP

- **Dedicated code is faster than general purpose code**
  - “Let's bypass Linux”
- **Assign cores to run without any blocking**
  - “Let's bypass Linux”



- **But... you bypassed Linux**
  - You have to write new environment

The 100% Linux alternative

- RT Patch – determinism, low latency
- UIO - direct hardware access from user space
- CPU affinity
- Scheduler improvements
- IRQ handling improvements

  You now have a process running in Linux which:

  - has a core 100% to itself,
  - can use most of Linux's regular APIs *,
  - uses regular IPC* to talk to other processes

- Some “secret sauce” ingredients to the new Bare Metal Engine™ in MontaVista Carrier Grade Linux 6

*: don't use blocking APIs, duh!

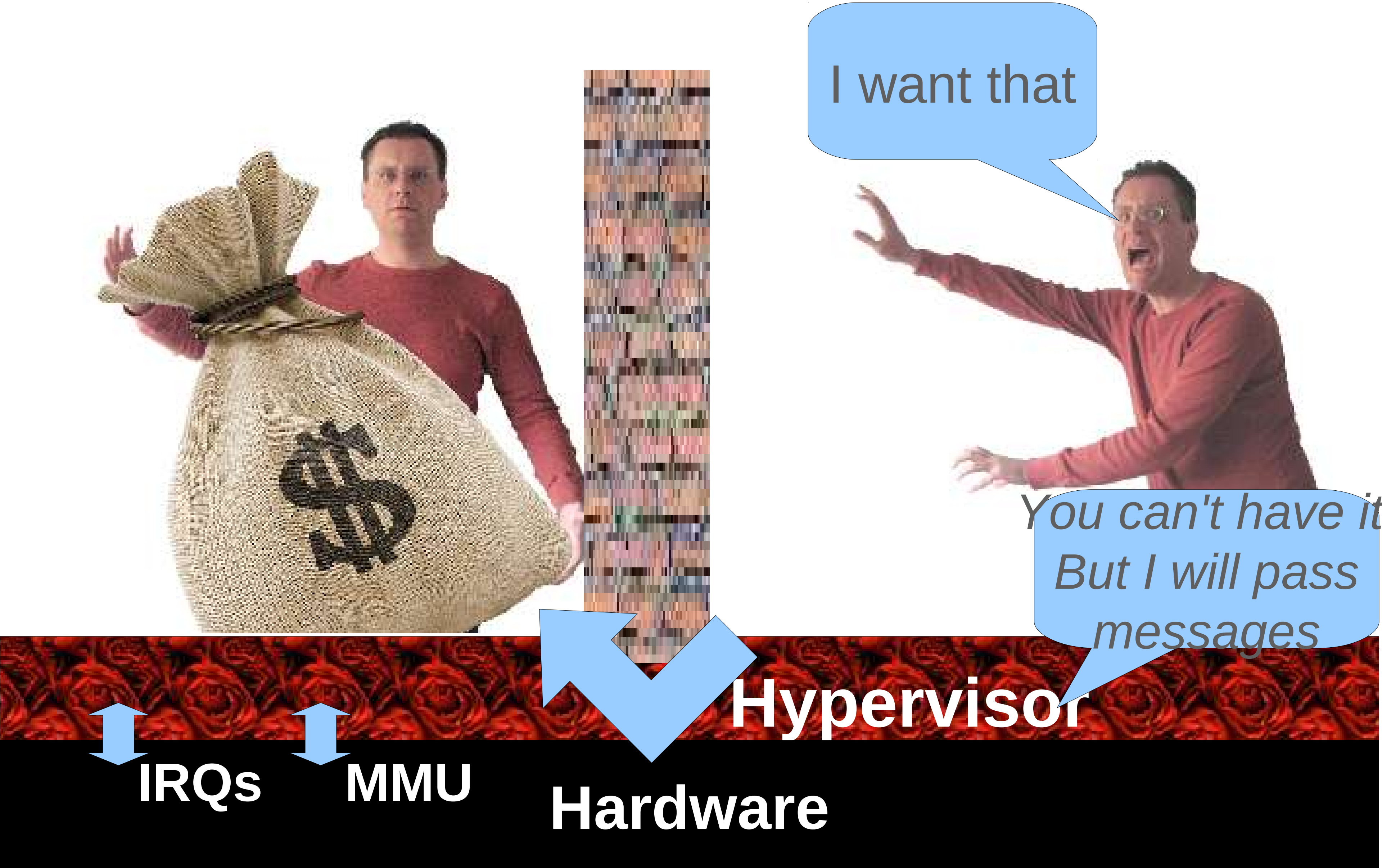
Virtualization

Virtualization somewhat right: dual OS on one CPU (if you must)

- **Reduce BOM**
- **Two OS'es:**
  - Linux for Applications
  - RTOS for GSM base
- **Single ARM9 CPU**

virtualLogix Real-Time Virtualization™

OS/License/Security Isolation



I want that

You can't have it
But I will pass messages

Hypervisor

IRQs

MMU

Hardware

Virtualization the right way:

Server Hardware Consolidation

In data centers:

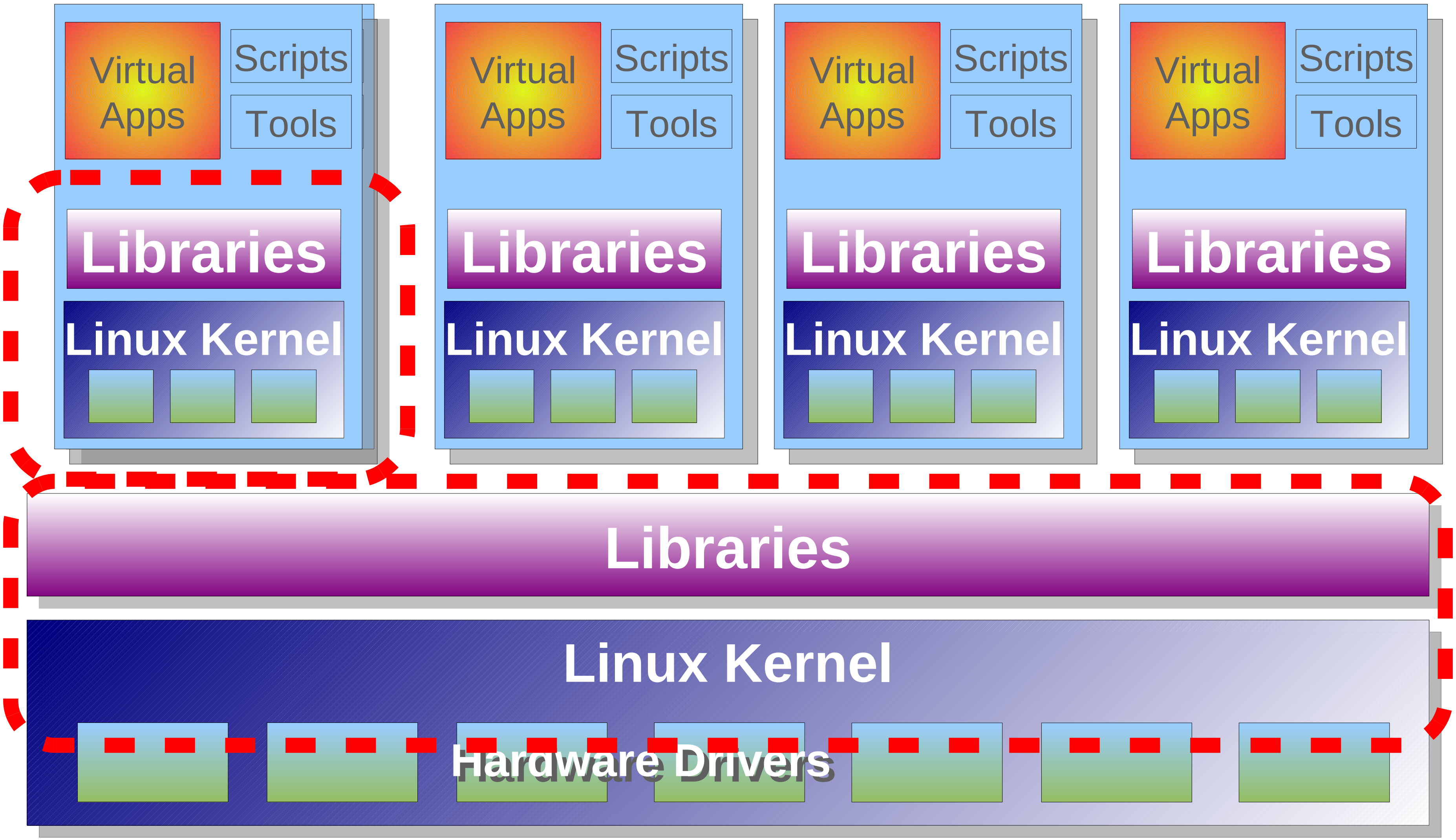
- Mail server(s)
- Web server(s)
- Intranet server
- File server(s)
- Firewall

All overdimensioned HW

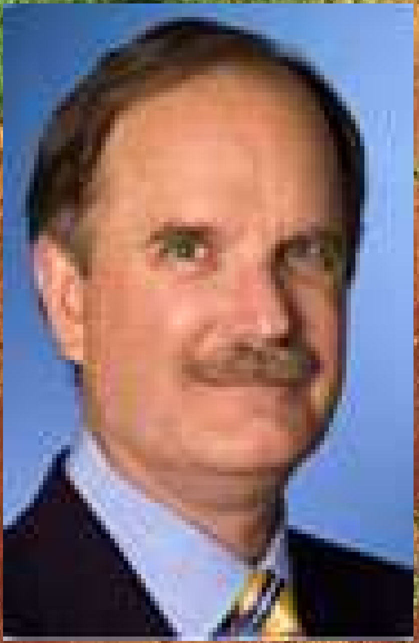
All usually below 10% load

Combine into 1 box using virtualization

Virtualization & Performance



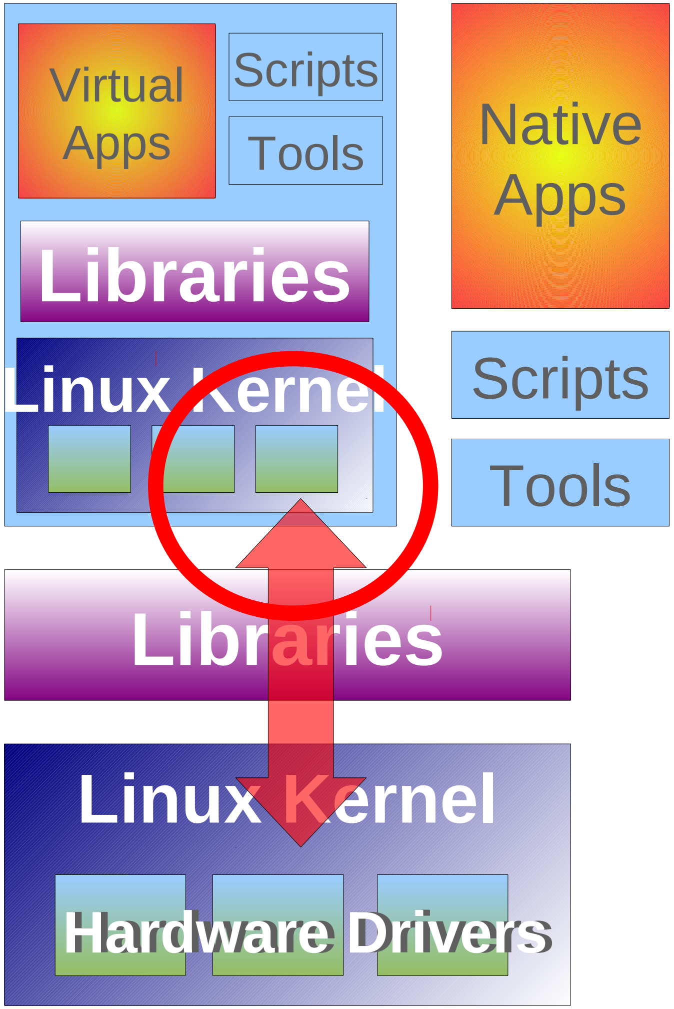
The Second Rule of Ready:



“No extra layer of software can make your system faster.”

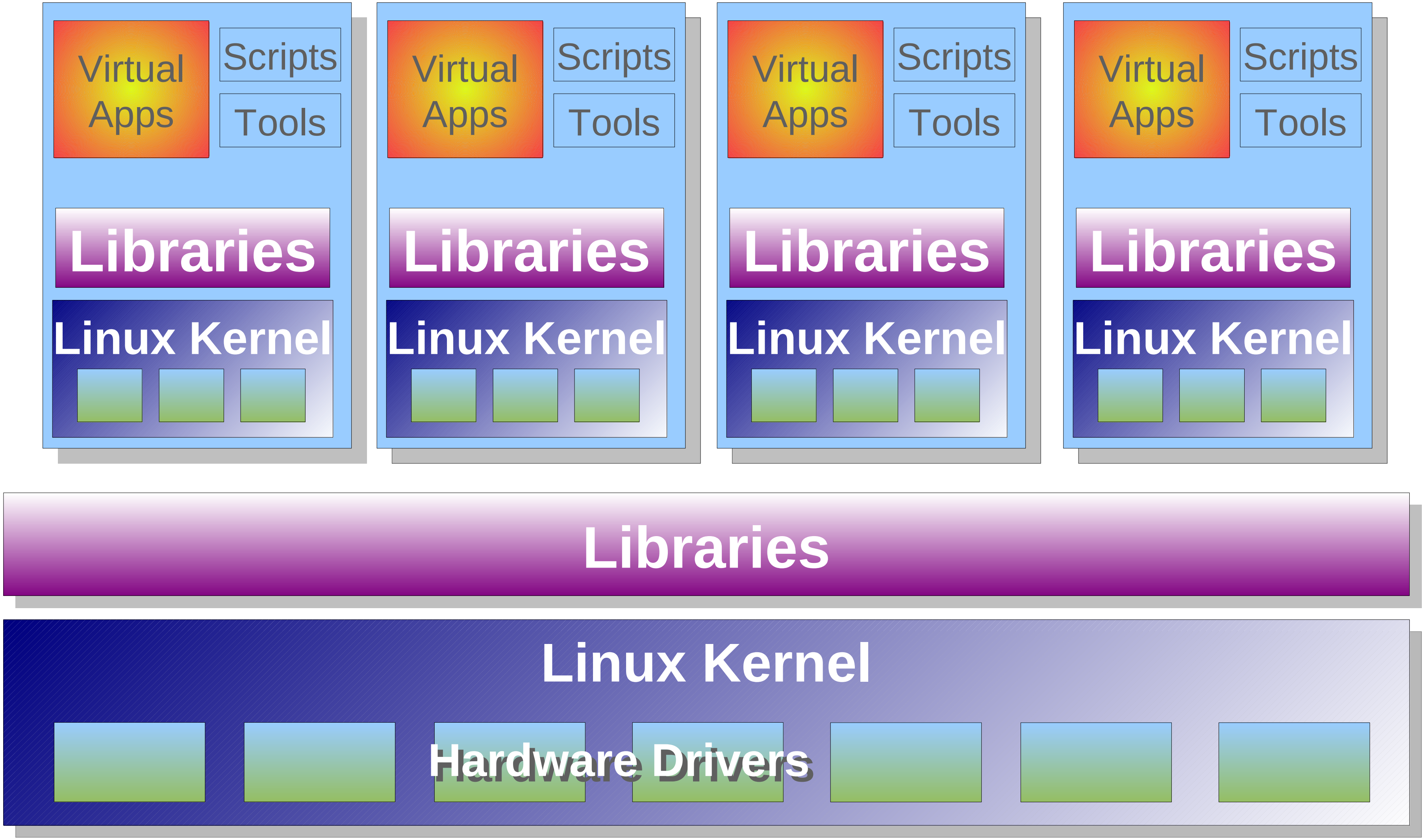
Jim Ready,
pioneer of VRTX and embedded Linux

Virtualization & Para-virtualization



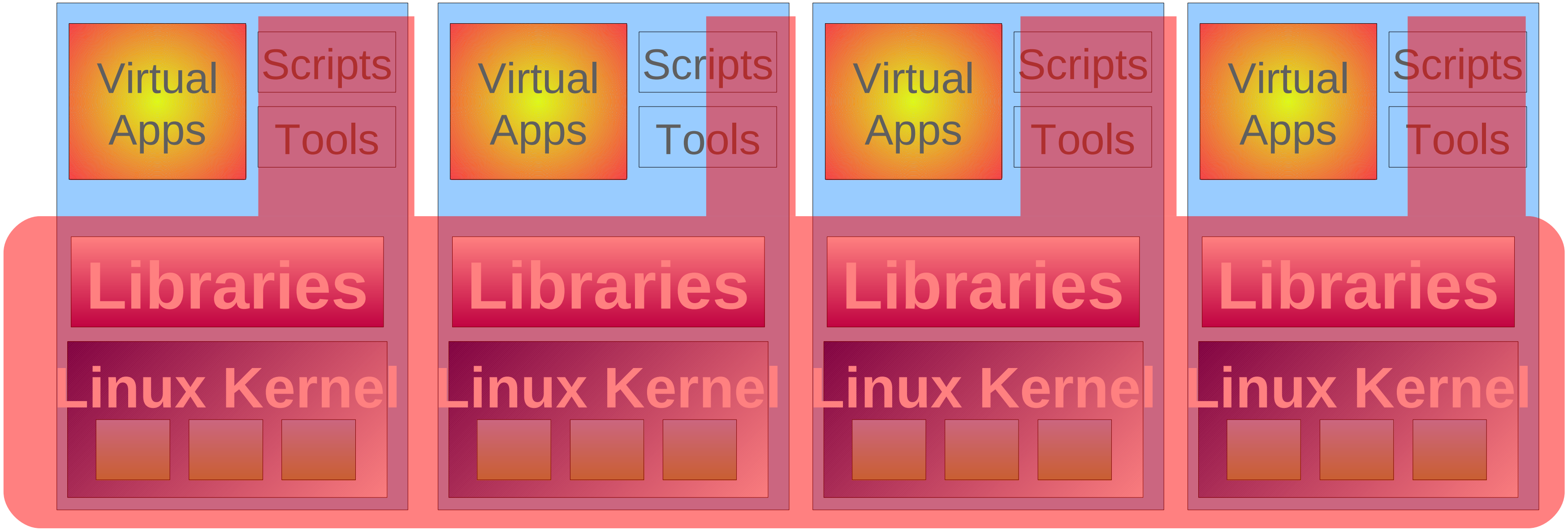
Para virtualization: guest OS has knowledge of virtualization environment. Usually to prevent expensive operations

Virtualization, Performance & Space

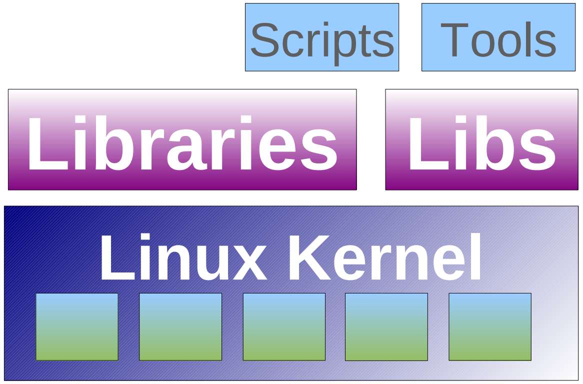


10/28/10

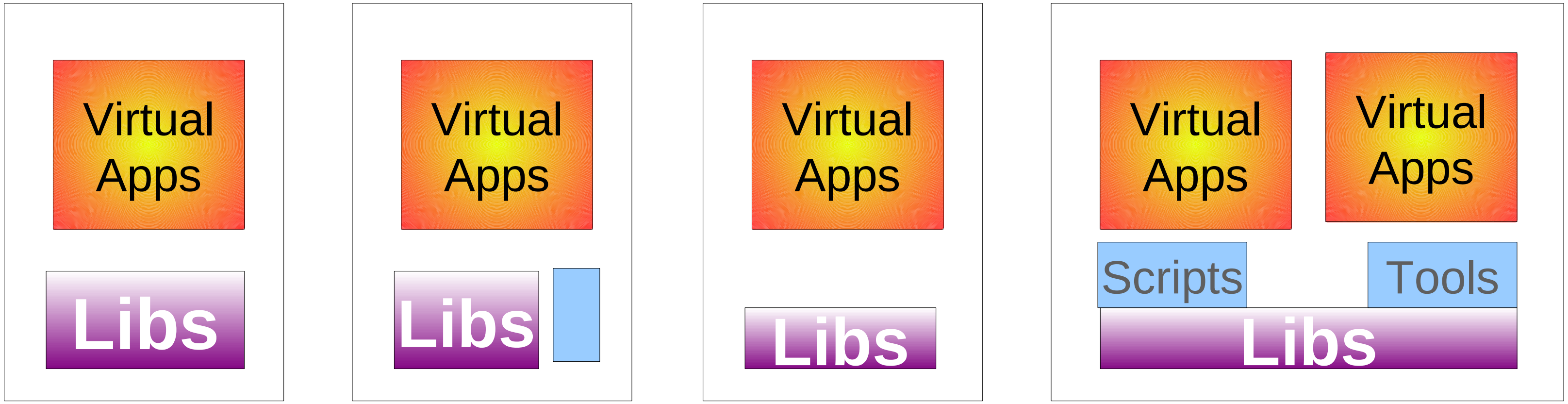
Virtualization, Performance & Space

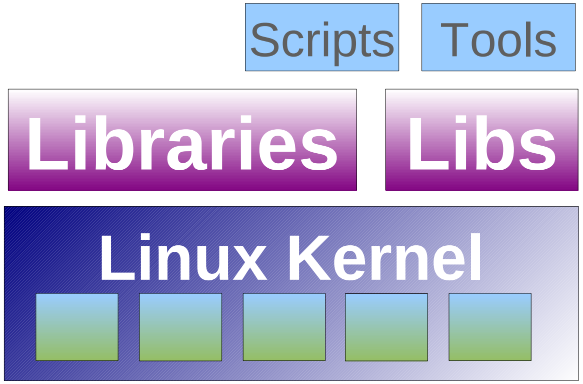


What a waste of space and performance!



Containers





- **More than chroot**
- **Allow custom libraries (glibc)**
- **Same across all:**
  - Udev, D-Bus, kernel

Containers, part II

- **Interesting features:**
  - Use network bridging for multiple Ethernet interface
    - Custom handling per container possible
  - Integrates nicely with Cisco VRF
- **Best feature: Resource usage limitations**
  - Partition CPU, network, disk bandwidth
- **Also interesting: integration with SELinux**
  - Make a container really secure

The Three Steps

$$\sum_{i=1}^{17} slide(i) \quad = \quad 3 \ steps$$

Q #1 Multiple OSes or Bare Metal?

- **Or: do you really need something besides Linux?**
- **Answer YES if:**
  - You need an RTOS, Bare Metal, Windows
- **Answer NO if:**
  - You need multiple Linuxes or one Linux “with tricks”

Q#2a Reason for multiple OS'es?

Do you have Legacy code

- And you don't want to port it?

- **Do you require features that are not in Linux?**
  - Windows?
  - RTOS?
  - Bare Metal?

You are now aware of the alternatives???

Q#2b Reasons for multiple Linuxes?

No reason for virtualization if:

- you think you need multiple versions of tools or libraries
- you need multiple network ports
- You want to partition the resource usage

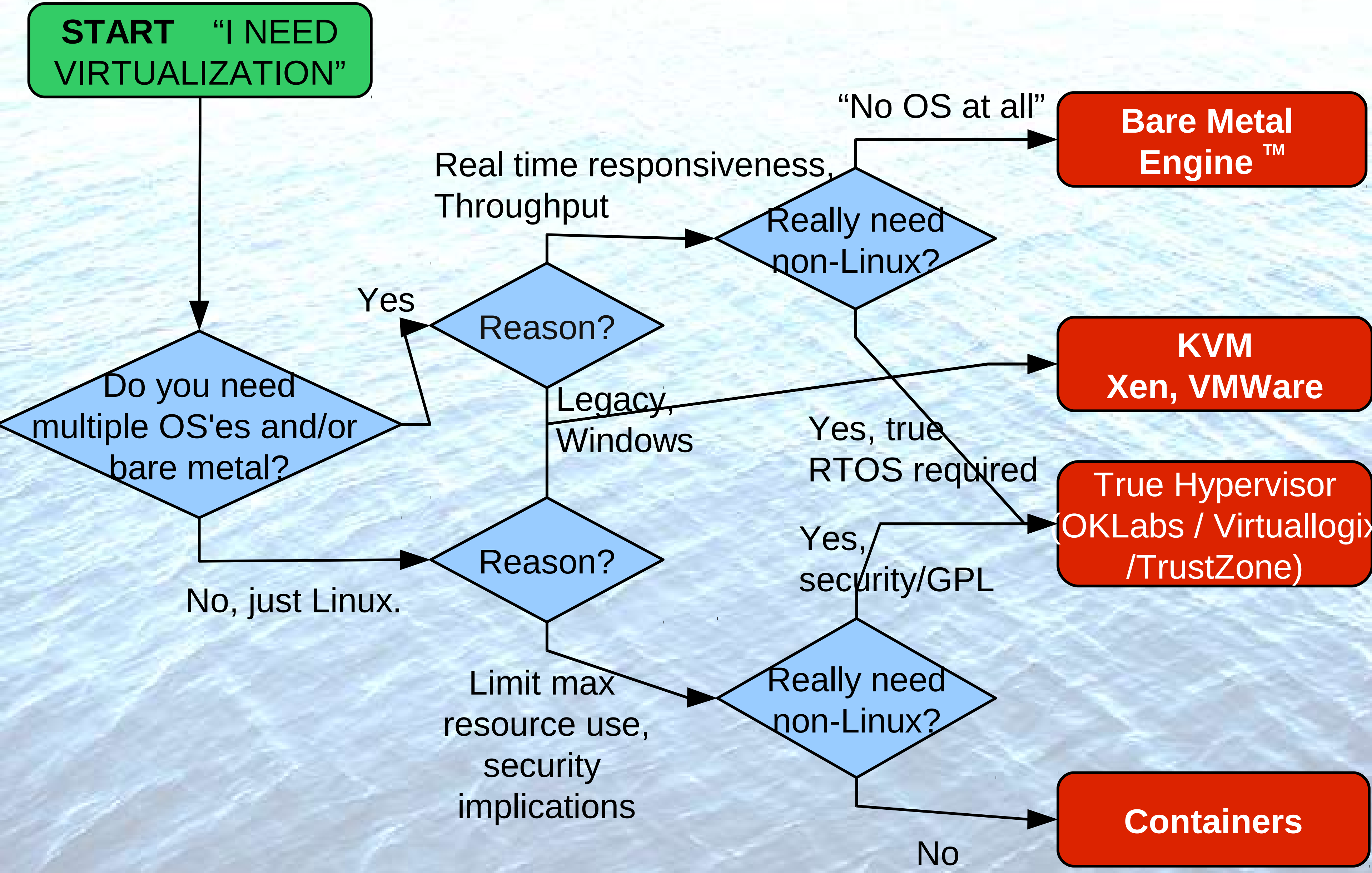
- **Do you need kernel drivers that no longer work with current kernels?**
  - Hmmm... What are you getting yourself into?

Q#3 Really need non-Linux?

- **Legacy**
- **RTOS-type real time response**
  - Classic argument for microkernel + Linux
  - PREEMPT_RT !
- **GPL/Security shielding**
- **Much higher network performance than Linux kernel**
  - Use Asymmetric MultiProcessing?
    - FreeScale QorIQ
    - Cavium Octeon
    - Netlogic XL* family

} Bare Metal Engine™

The three steps in one diagram



START “I NEED VIRTUALIZATION”

Do you need multiple OS'es and/or bare metal?

Yes

No, just Linux.

Reason?

Real time responsiveness, Throughput

Legacy, Windows

Reason?

Limit max resource use, security implications

Really need non-Linux?

“No OS at all”

Yes, true RTOS required

Really need non-Linux?

Yes, security/GPL

No

Bare Metal Engine ™

KVM
Xen, VMWare

True Hypervisor (OKLabs / Virtuallogix /TrustZone)

Containers

Summary

- “There's nothing new here”
- There are obviously good examples of virtualization
- Linux features are underestimated by most
  - Containers – partitioning & resource limiting
  - KVM – true virtualization
  - Simulate “bare metal” high performance without going to another environment using UIO & friends