Security Challenges in Virtualized Environments

Joanna Rutkowska,
Invisible Things Lab

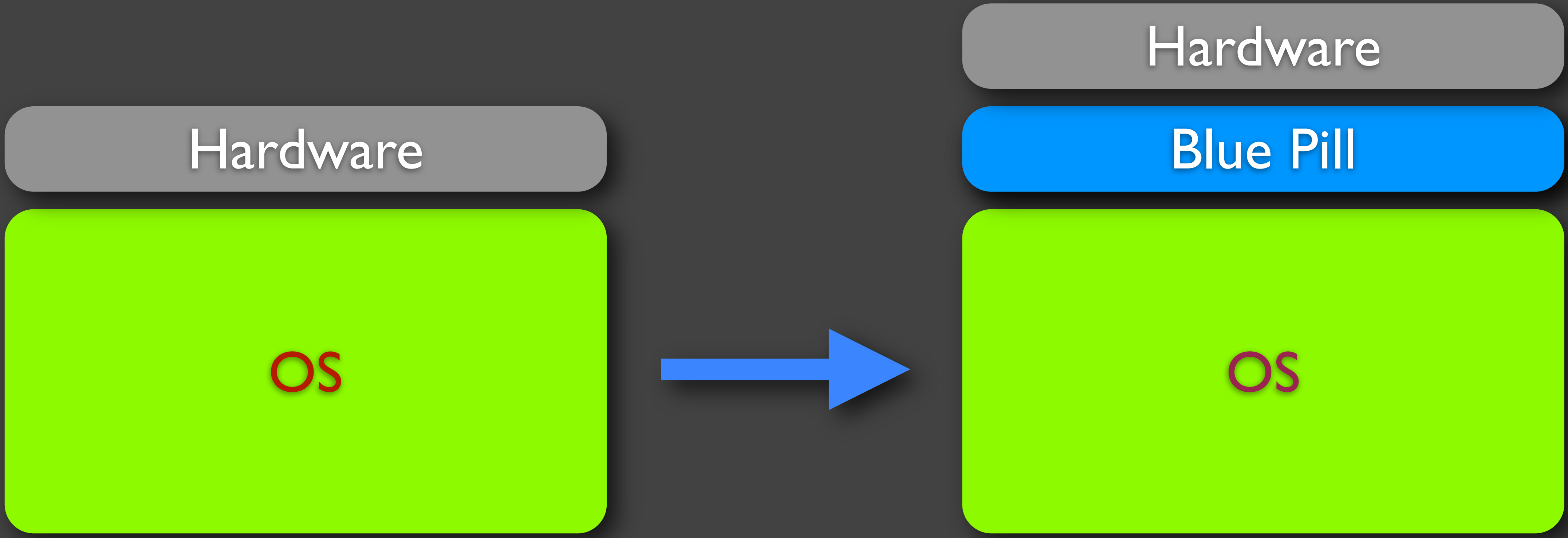
RSA Conference,
San Francisco, April 8th 2008

1 Virtualization-based **MALWARE**

2 Using Virtual Machines for **ISOLATION**

3 **NESTED** virtualization

Virtualization-based MALWARE



Hardware

OS

Hardware

Blue Pill

OS

AMD-V

Intel VTx

Blue Pill Characteristics

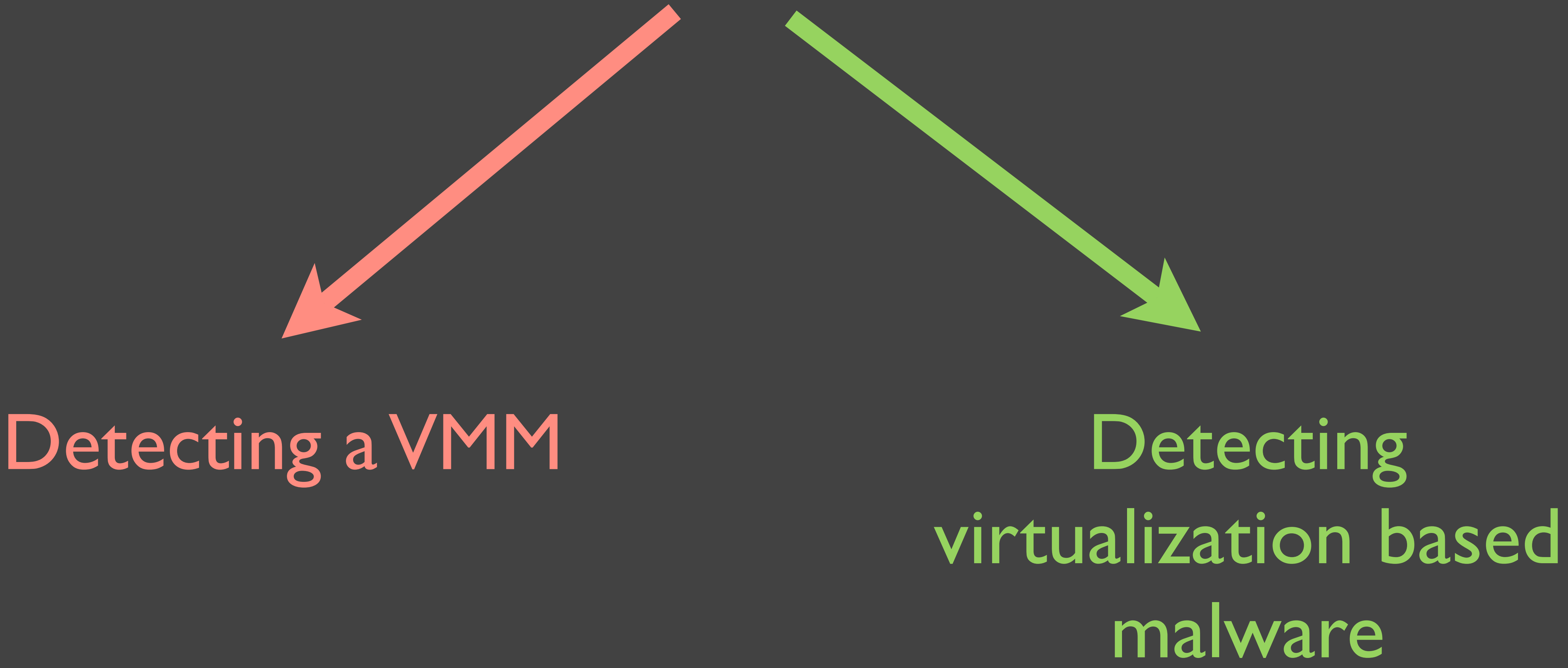
<u>NO HOOKS!</u> → Cannot be detected using any integrity scanner

On the fly installation → No boot/BIOS/etc modifications necessary

No I/O virtualization → Negligible performance impact (your brand new 3D card will still work!)

Blue Pill detection

Blue Pill detection



Detecting a VMM

Detecting virtualization based malware

VMM detection

Direct timing analysis

Guest time virtualization

HPET timers

Blue Chicken

CPU specific behavior

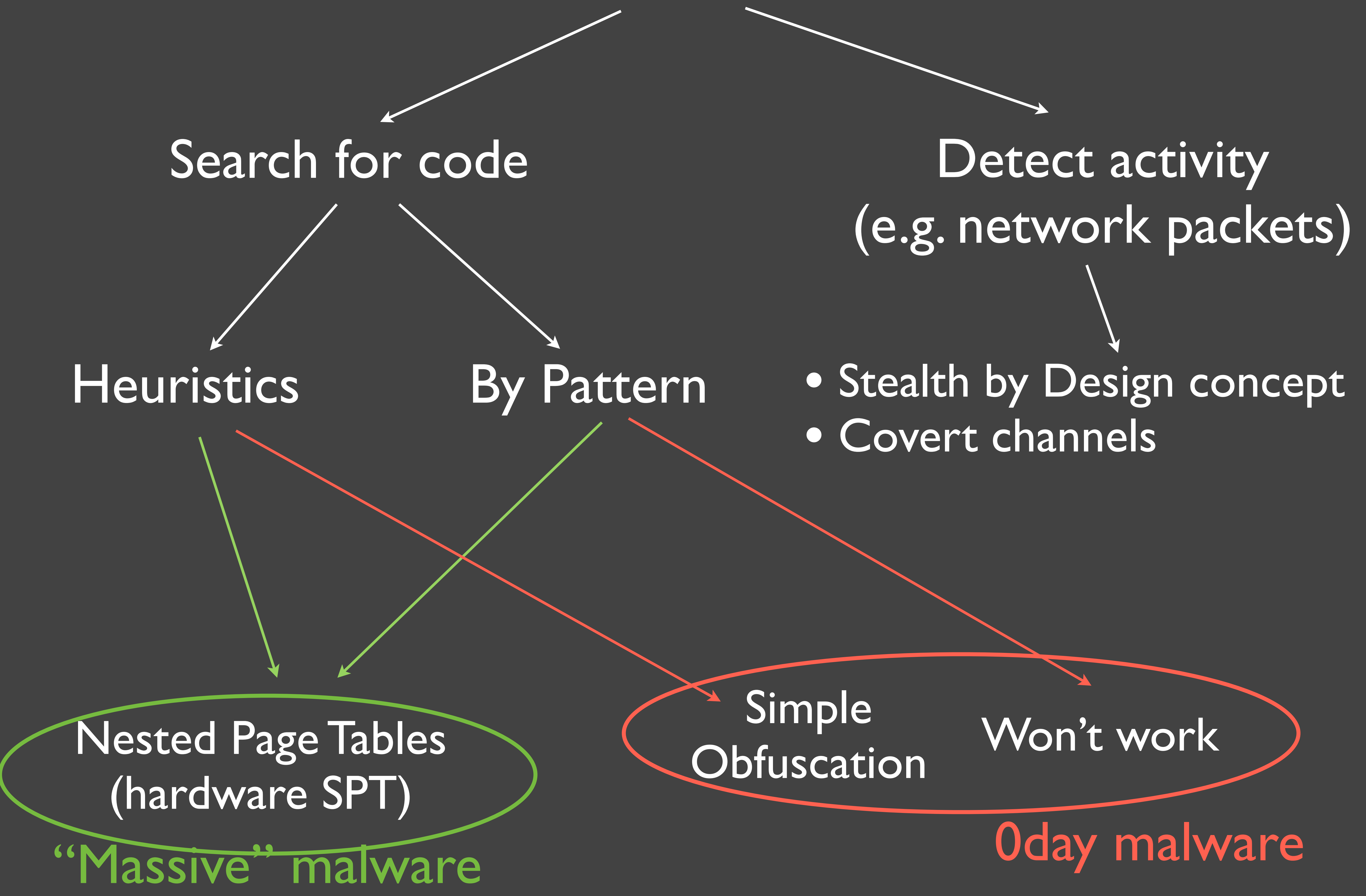
TLB profiling

VMM detection?

- Everything is going to be virtualized!
- Thus the information that "there is a hypervisor in the system"...
- ...would be pretty much useless...

Detecting virtualized malware?

No Hooks!



Search for code

Detect activity (e.g. network packets)

Heuristics

By Pattern

- Stealth by Design concept
- Covert channels

Nested Page Tables (hardware SPT)

Simple Obfuscation

Won’t work

“Massive” malware

0day malware

But why we can’t use obfuscation for “classic” malware?
Because it leaves hooks anyways!
And we can always find those hooks, no matter how obfuscated the classic malware is!

The whole big deal about Blue Pill is:

NO HOOKS in the system!

Blue Pill prevention

Disable virtualization?

How about also disabling your network card so you never got infected from the Internet?

Install a trusted hypervisor first?

Installing trusted hypervisor

Static Root of Trust Measurement

BIOS > MBR > VMM
e.g. MS Bitlocker

Dynamic Root of Trust Measurement

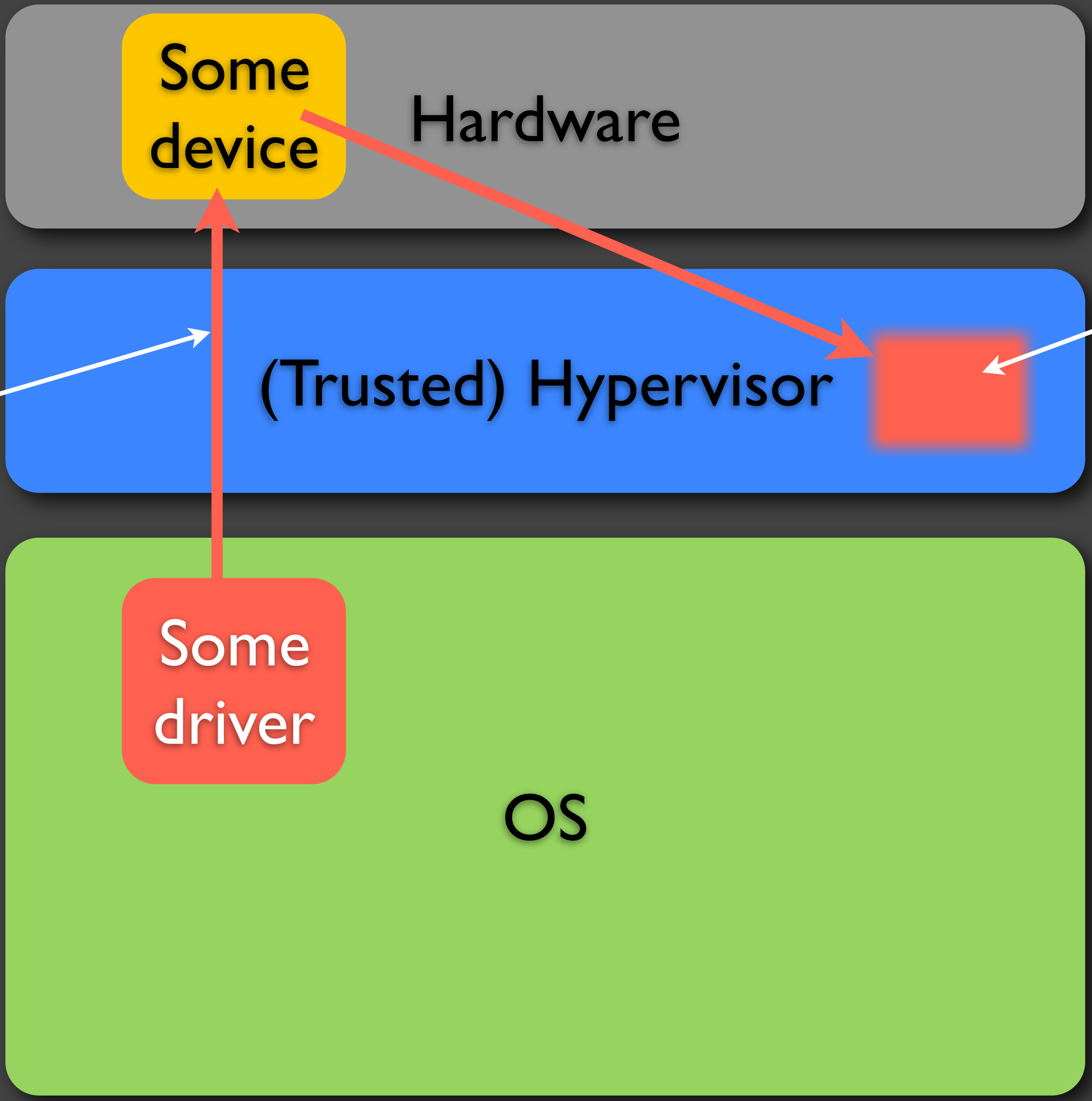
SENTER (Intel TXT)
SKINIT (AMD SVM)

Trusted vs. Secure?

- SRTM and DRTM only assures that what we load is trusted...
- ...at the moment of loading!
- 3 sec later... it could be exploited and get compromised!

Trusted != Secure (e.g. flawless)

E.g. #1: The famous DMA problem



I/O: asks the device to setup a DMA transfer

Read/Write memory access!

IOMMU

- Solution to the problem of "DMA attacks"
- Intel calls it: VT-d
- Not much PC hardware supports it yet
    - Expected to change soon
- No THIN HYPERVISORS without IOMMU!

Other problems with VMMs?

Stay tuned...

All in all: it’s not trivial to have a trusted & secure hypervisor installed...

... but for sure this is the proper way to go...

1 Virtualization-based **MALWARE**

2 Using Virtual Machines for **ISOLATION**

3 **NESTED** virtualization

Using Virtual Machines for ISOLATION

Originally ISOLATION was supposed to be provided by Operating Systems...

- Separate processes/address spaces,
- User accounts & ACLs...

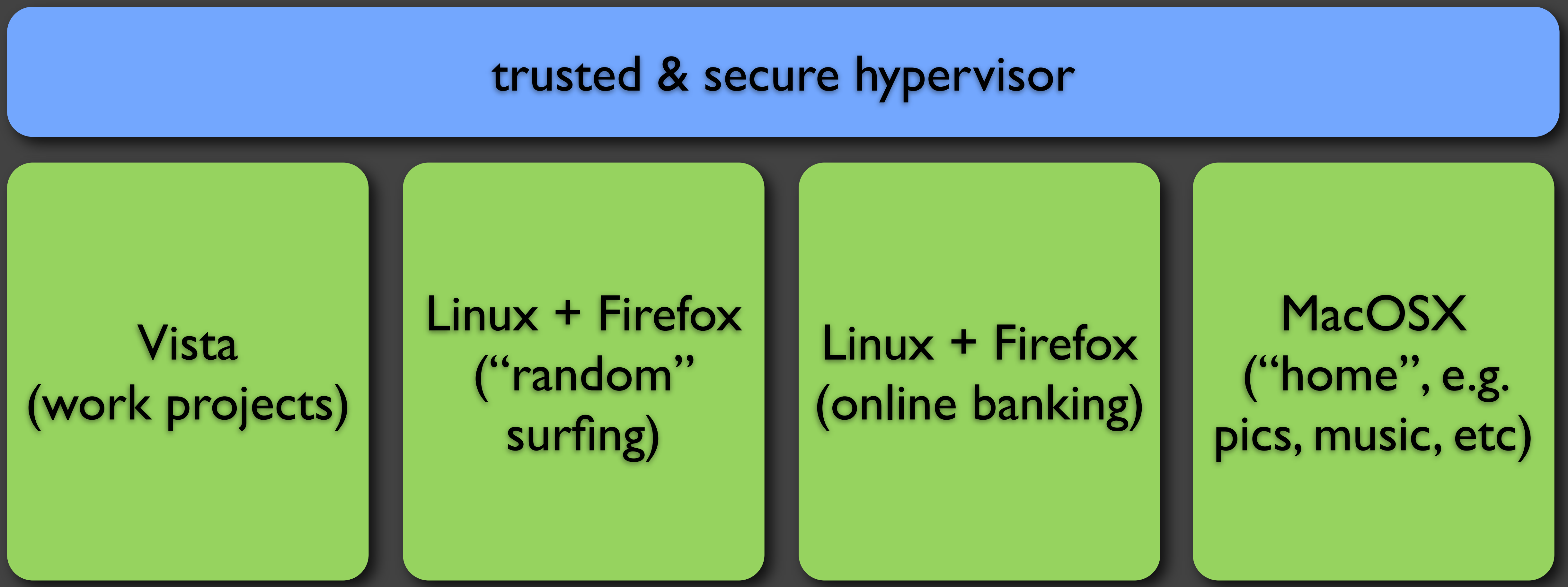
But in practice current OSes simply

fail at providing isolation!

Why OSes fail?

- Kernel bugs!
- Kernel bugs!!
- Kernel bugs!!!
- Bad design, e.g.:
  - XP and "all runs as admin" assumption
  - Vista's UAC assumes admin rights should be granted to every installer program!

VMMs for the rescue!



trusted & secure hypervisor

Vista
(work projects)

Linux + Firefox
(“random”
surfing)

Linux + Firefox
(online banking)

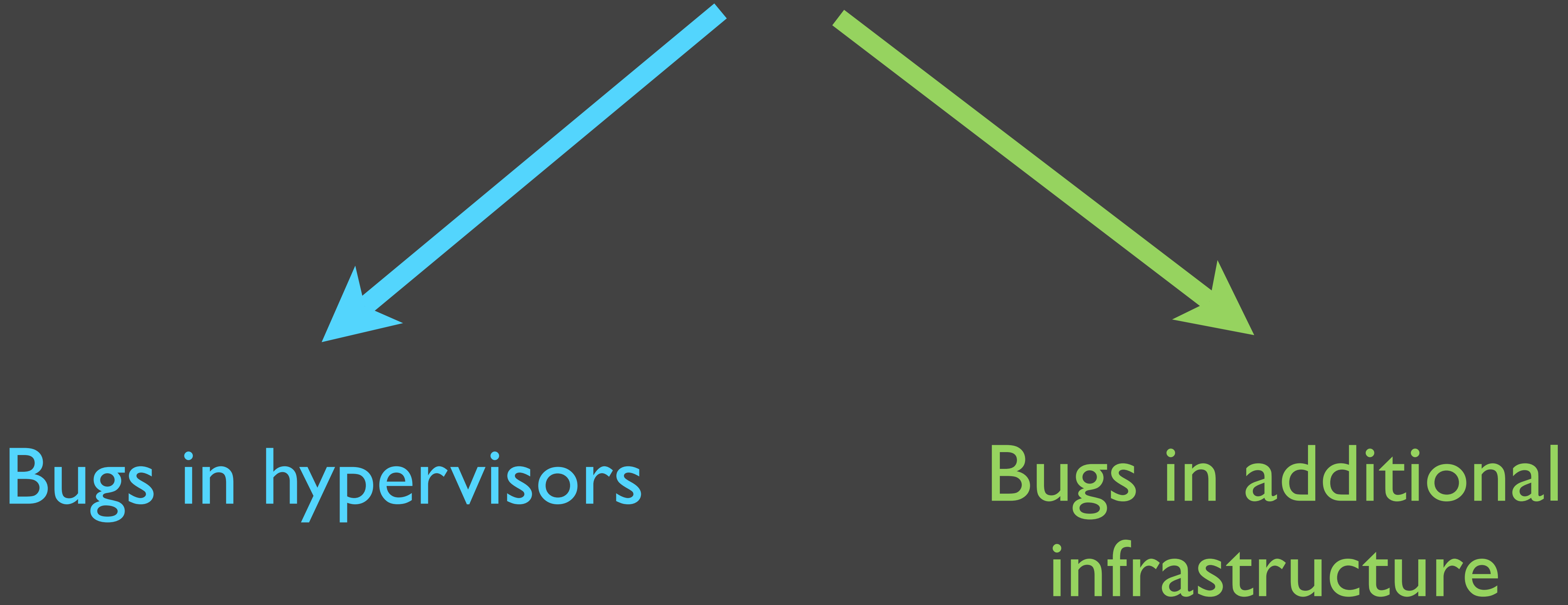
MacOSX
(“home”, e.g.
pics, music, etc)

Challenges

- Performance
- Why is VMM/hypervisor going to be more secure then OS's kernel?

VMM bugs?

VMM Bugs



Bugs in hypervisors

Bugs in additional infrastructure

E.g. #1: CVE-2007-4496

- VMWare ESX 3.0.1
  - http://www.vmware.com/support/vi3/doc/esx-8258730-patch.html
- Found by Rafal Wojtczuk (McAfee)
- September 2007
- Guest OS can cause memory corruption on the host and *potentially* allow for arbitrary code execution on the host

E.g. #2: CVE-2007-0948

- Microsoft Virtual Server 2005 R2
  - http://www.microsoft.com/technet/security/bulletin/ms07-049.mspx
- Found by Rafal Wojtczuk (McAfee)
- August 2007
- Heap-based buffer overflow allows guest OS to execute arbitrary code on the host OS

E.g. #3: CVE-2007-4993

- Xen 3.0.3
  - http://bugzilla.xensource.com/bugzilla/show_bug.cgi?id=1068
- Found by Joris van Rantwijk
- September 2007
- By crafting a grub.conf file, the root user in a guest domain can trigger execution of arbitrary Python code in domain 0.

E.g. #4: Various Bugs

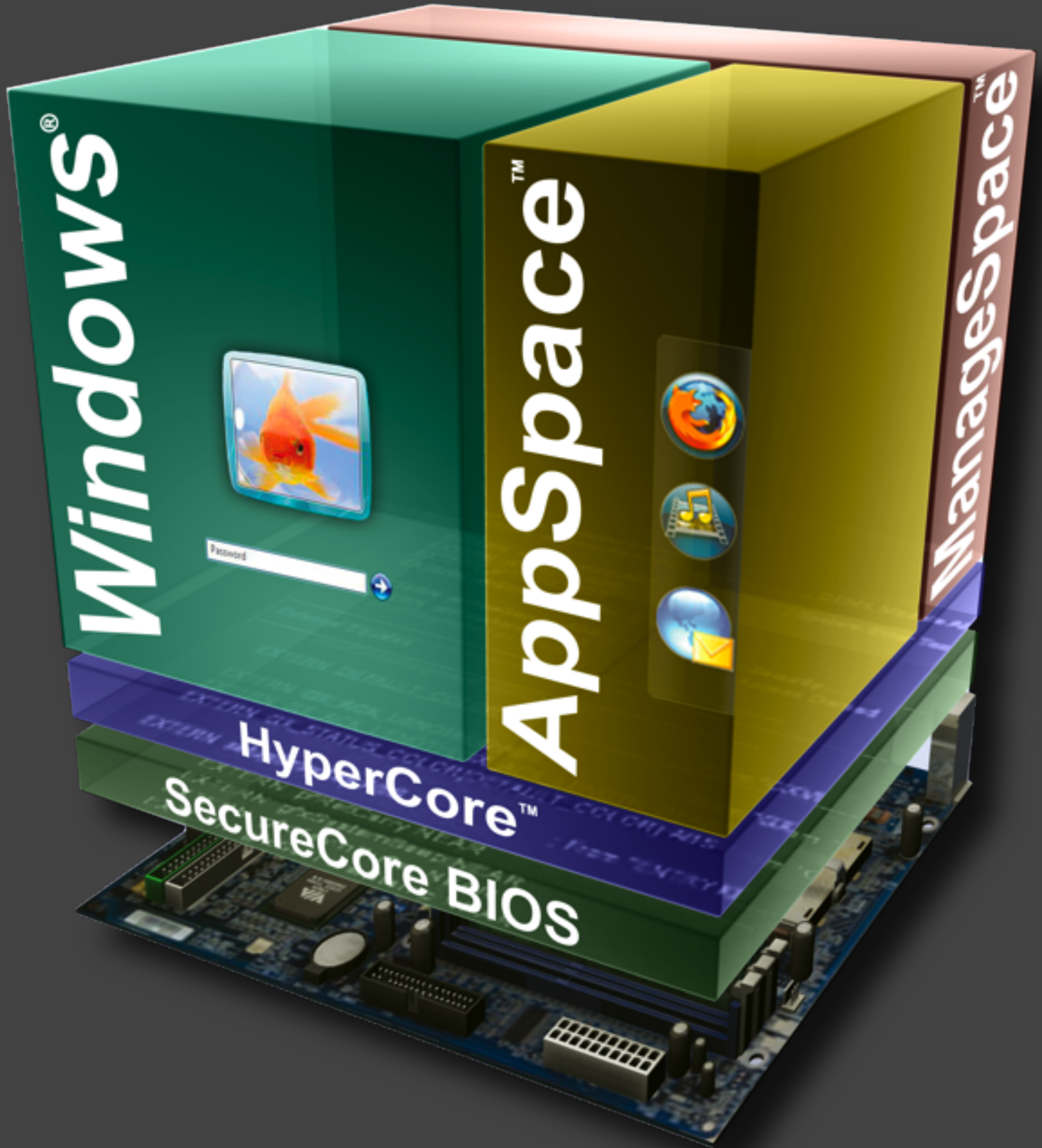
- Paper by Tavis Ormandy (Google)
  - http://taviso.decsystem.org/virtsec.pdf
- April 2007
- Disclosed bugs in VMWare, XEN, Bochs, Virtual PC, Prallels
- A simple fuzzers for:
  - Instruction parsing by VMMs
  - I/O device emulation by VMMs

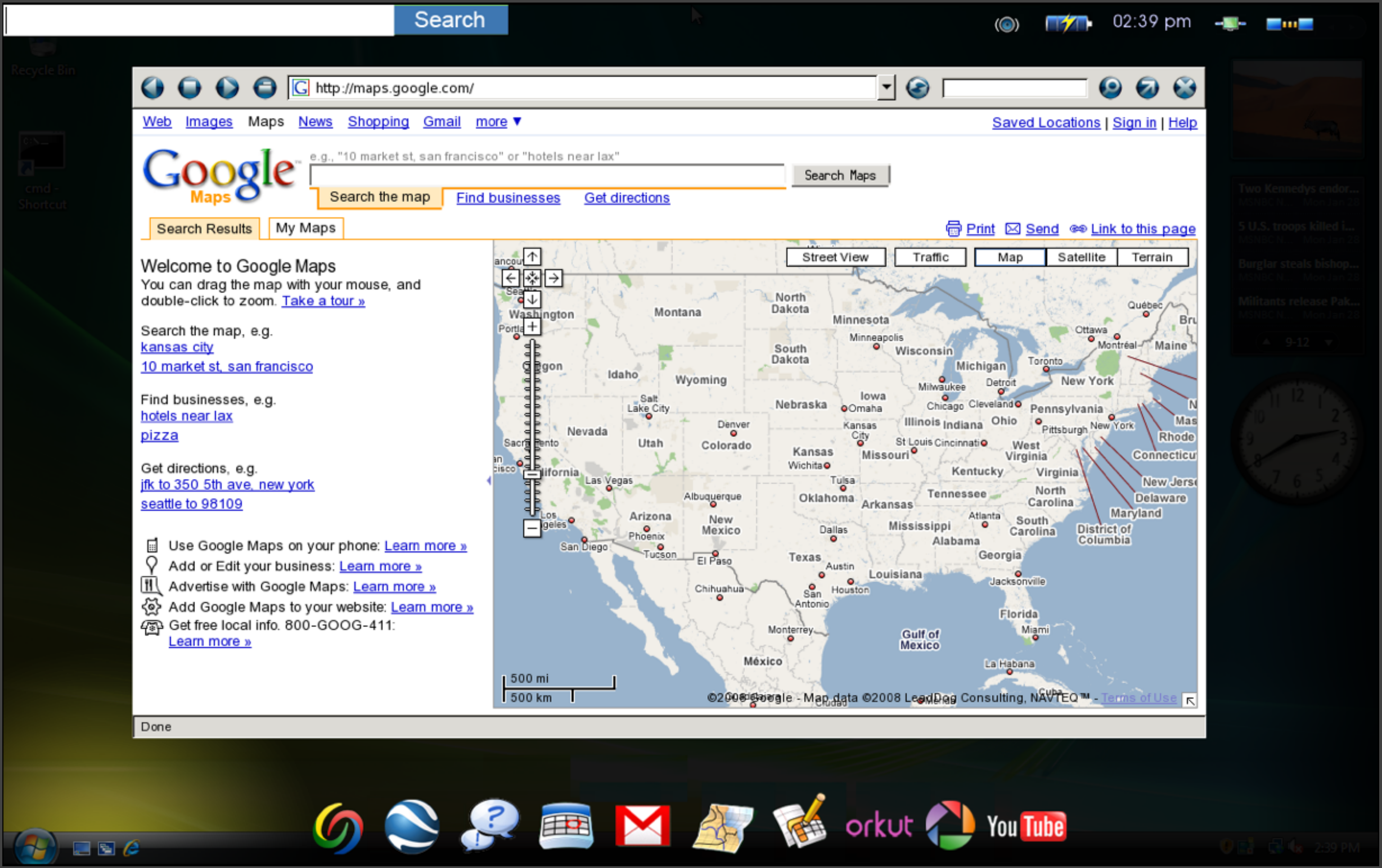
As you see current VMMs are far from being flawless...

To make VMMs more secure we need to keep them

ultra-thin and small!

Phoenix HyperSpace


Windows®
Password
AppSpace™
ManageSpace™
HyperCore™
SecureCore BIOS



Welcome to Google Maps

You can drag the map with your mouse, and double-click to zoom. Take a tour »

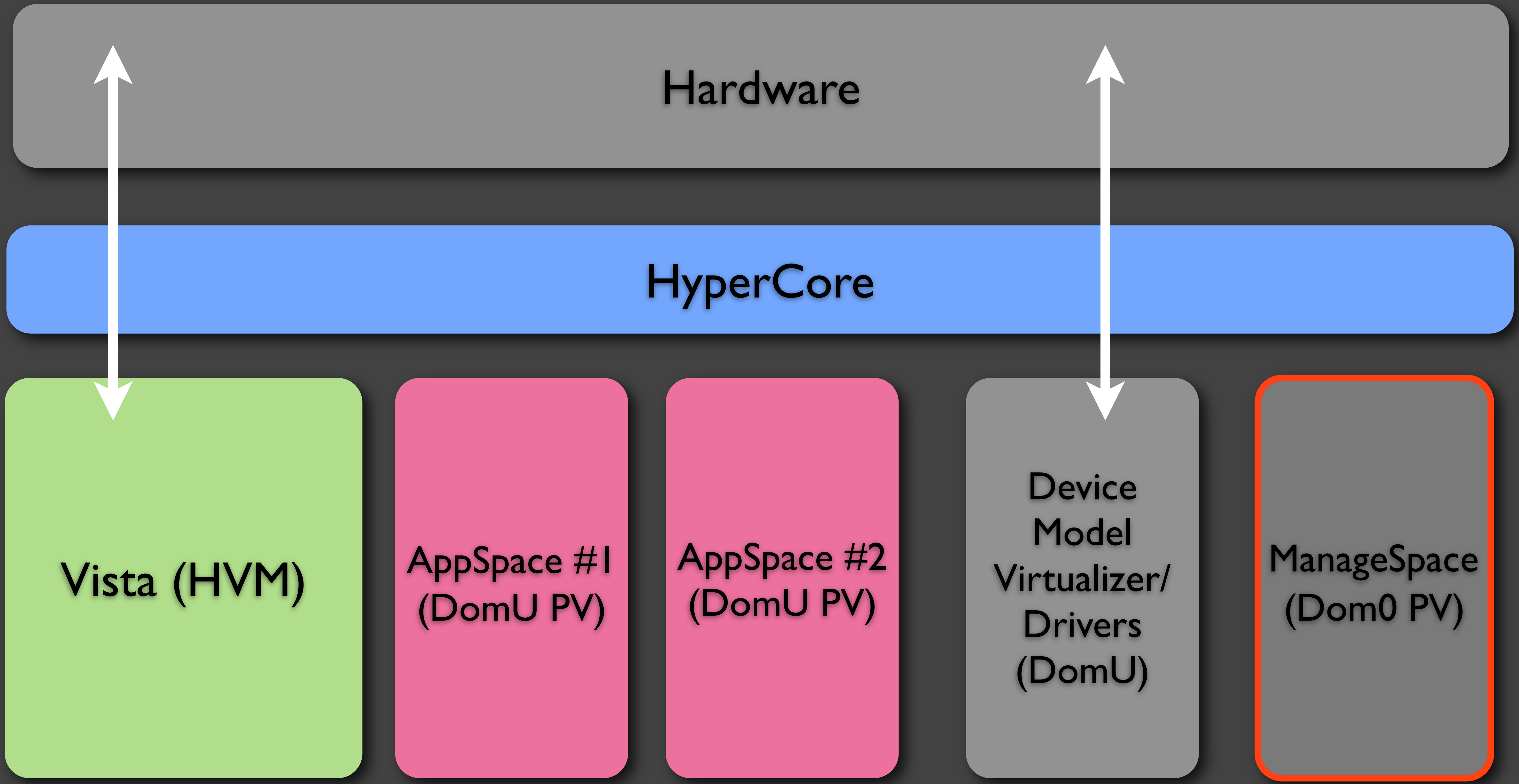
Search the map, e.g.
kansas city
10 market st, san francisco

Find businesses, e.g.
hotels near lax
pizza

Get directions, e.g.
jfk to 350 5th ave, new york
seattle to 98109

Use Google Maps on your phone: Learn more »
Add or Edit your business: Learn more »
Advertise with Google Maps: Learn more »
Add Google Maps to your website: Learn more »
Get free local info. 800-GOOG-411: Learn more »

HyperCore:
the type I hypervisor used for HyperSpace


Hardware
HyperCore
Vista (HVM)
AppSpace #1 (DomU PV)
AppSpace #2 (DomU PV)
Device Model Virtualizer/ Drivers (DomU)
ManageSpace (Dom0 PV)

The HyperCore

- Targets desktop/laptop systems
- Guest OS execute at near-native performance (including fancy graphics)
- Support for full ACPI (Power Management)
- Integrity: loaded via SecureCore BIOS (Static Root of Trust Measurement)
- Very thin - easy to audit!

Speeding things up

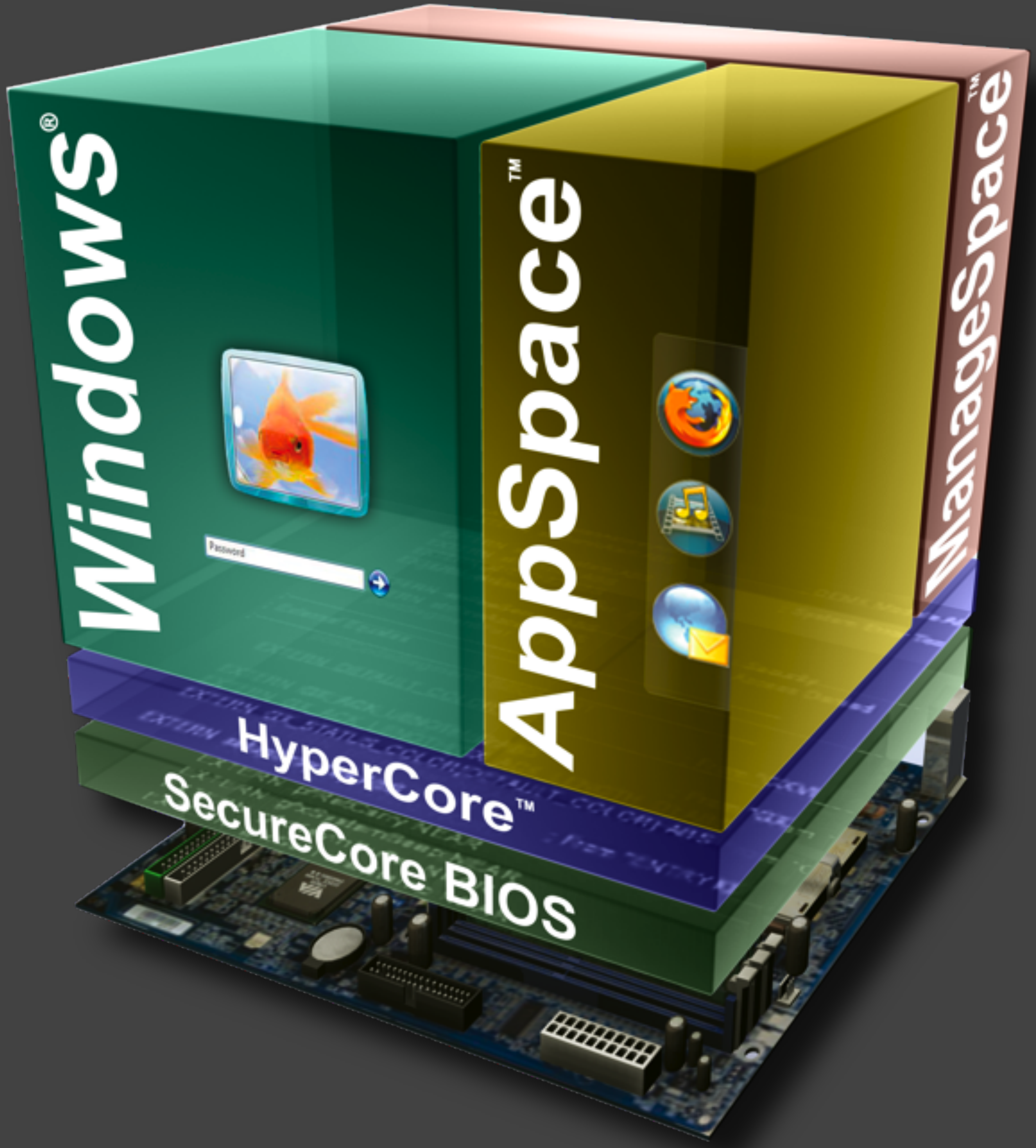
- Pass through for most devices
- SPT: 1-1 mapping for most pages for the Primary OS

Power Management

- ACPI tables exposed to the Primary OS, so that the overall power performance is optimized
- Efficient intercepts for power management control

Integrity

- Static RTM via Phoenix's SecureCore BIOS
- Dynamic RTM via Intel's TXT/AMD's SKINIT
- SMM-based watchdog for HyperCore code


Windows®
Password
AppSpace™
ManageSpace™
HyperCore™
SecureCore BIOS

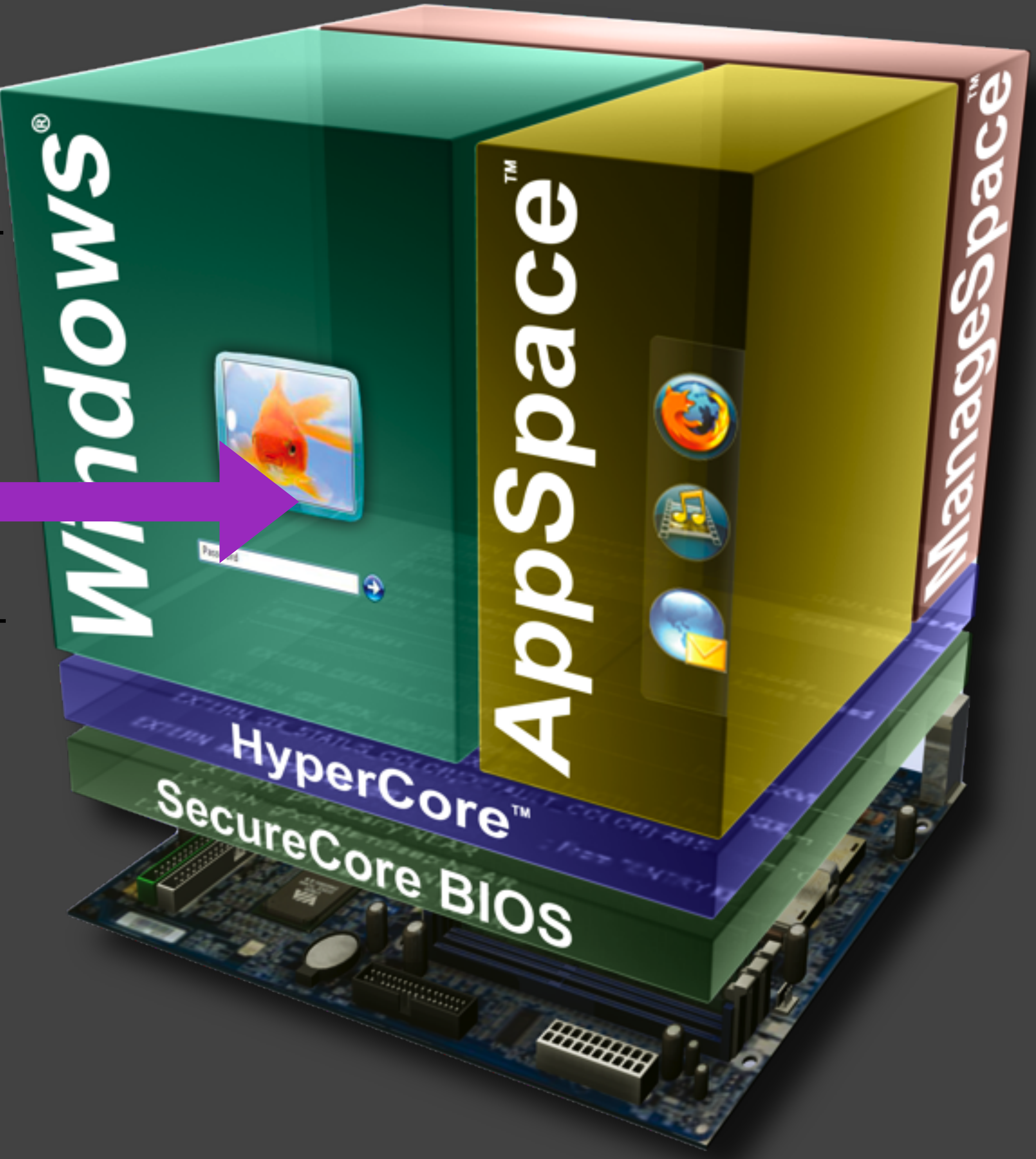
1 Virtualization-based **MALWARE**

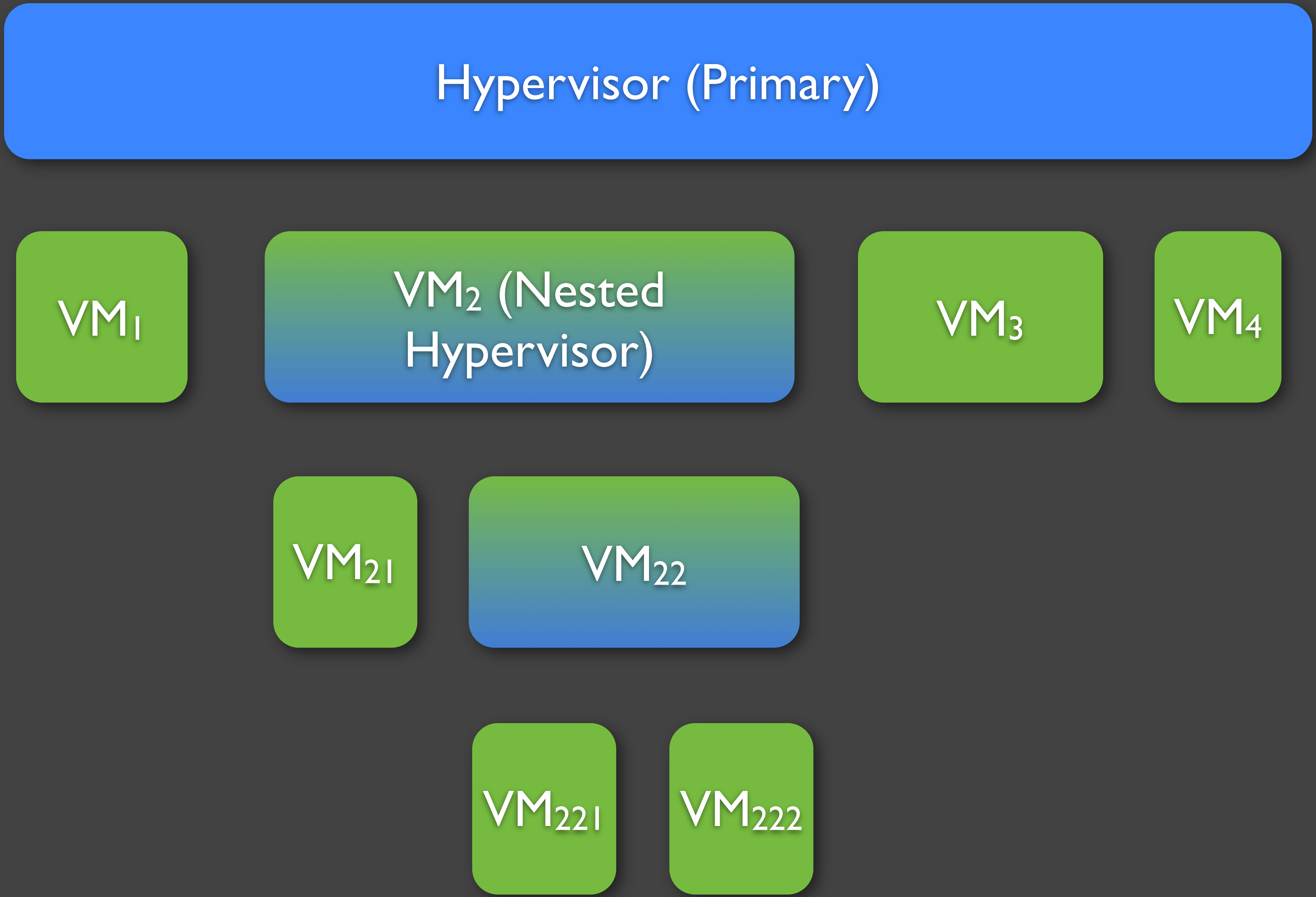
2 Using Virtual Machines for **ISOLATION**

3 **NESTED** virtualization

NESTED virtualization

What if a user wants to run e.g. Virtual PC here?





Hypervisor (Primary)

VM$_1$

VM$_2$ (Nested Hypervisor)

VM$_3$

VM$_4$

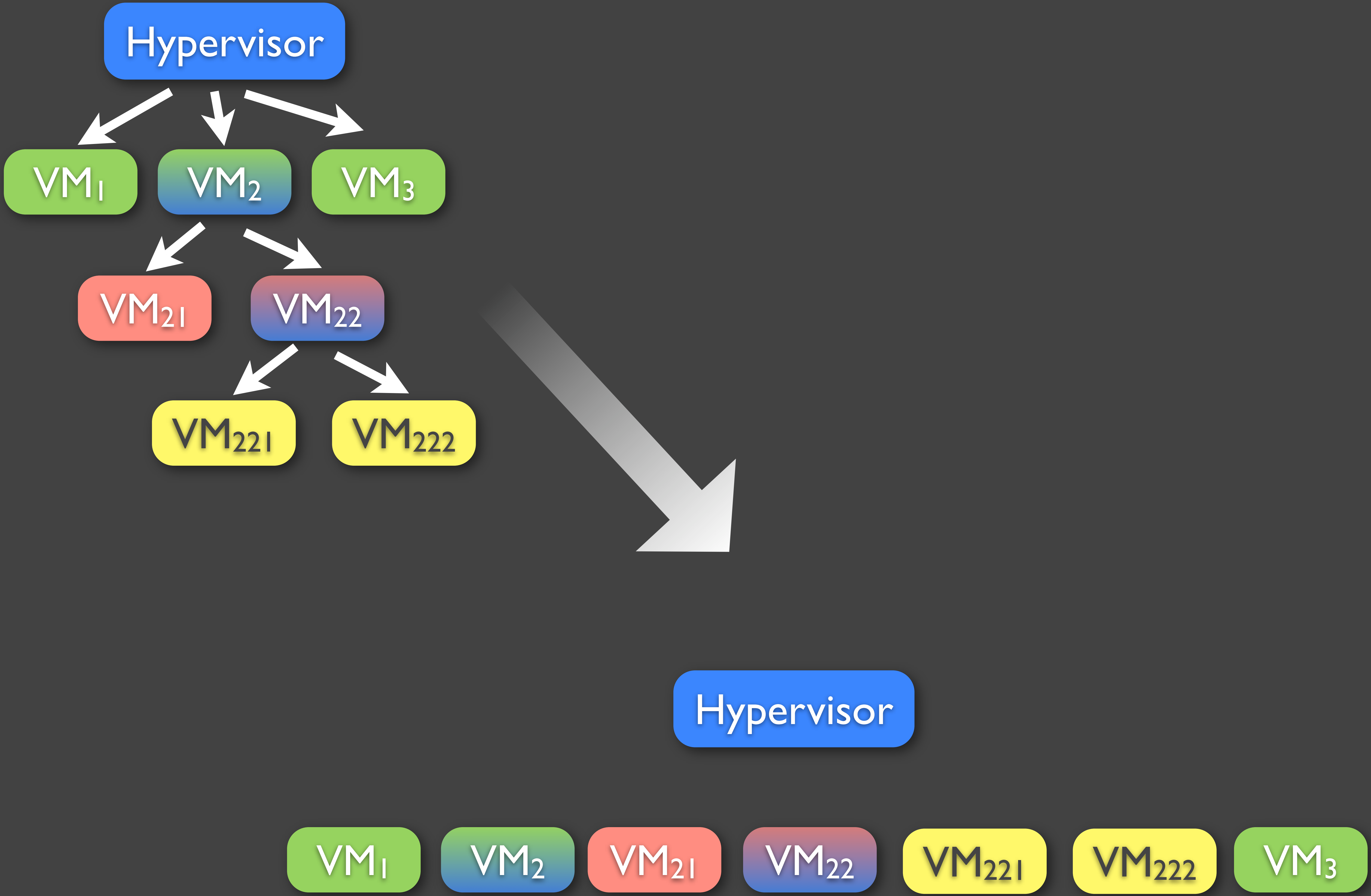
VM$_{21}$

VM$_{22}$

VM$_{221}$

VM$_{222}$

Idea of how to handle this situation...



Hypervisor

VM$_1$ VM$_2$ VM$_3$

VM$_{21}$ VM$_{22}$

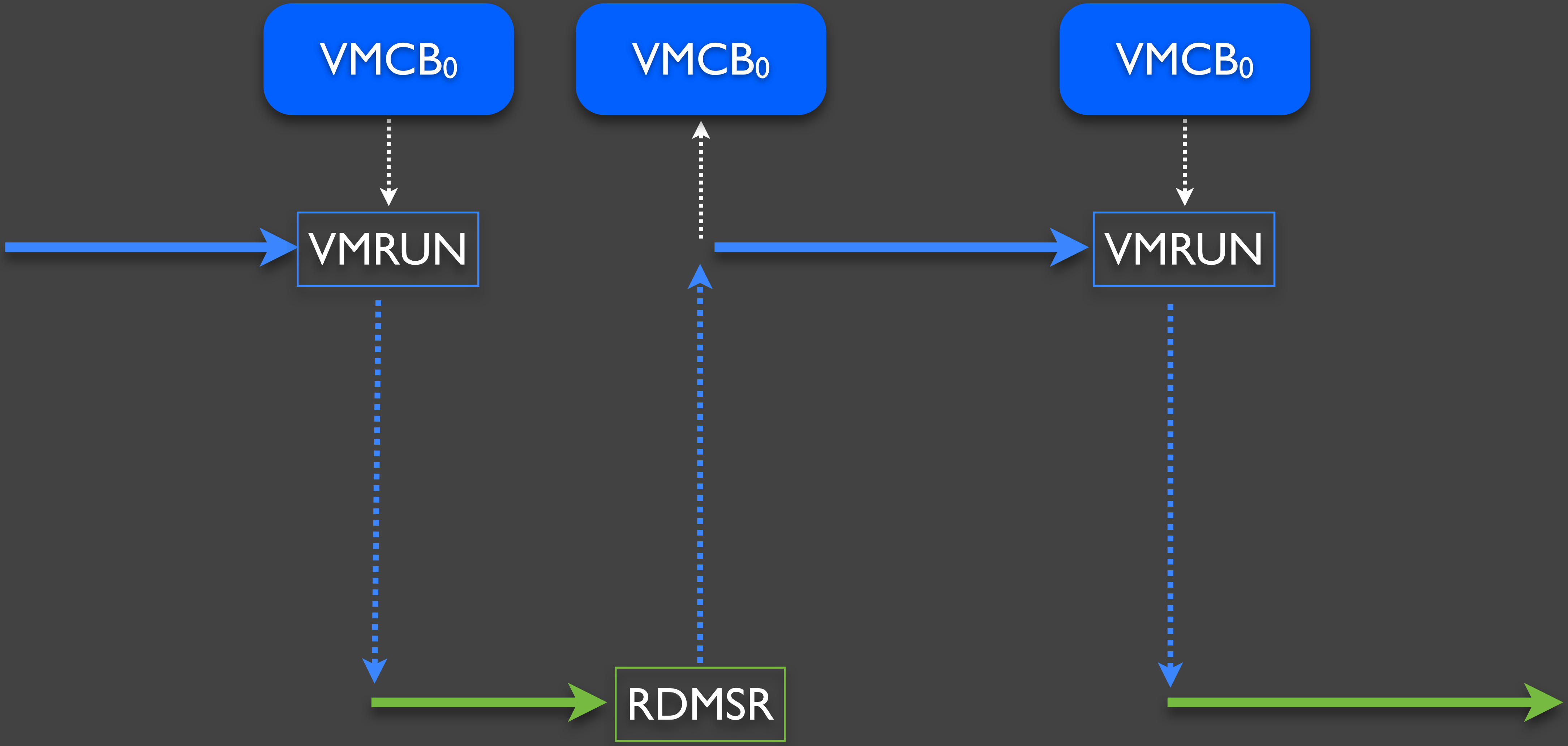
VM$_{221}$ VM$_{222}$

Hypervisor

VM$_1$ VM$_2$ VM$_{21}$ VM$_{22}$ VM$_{221}$ VM$_{222}$ VM$_3$

Now, lets look at the actual details :)

Let’s start with AMD-V...



VMCB$_0$

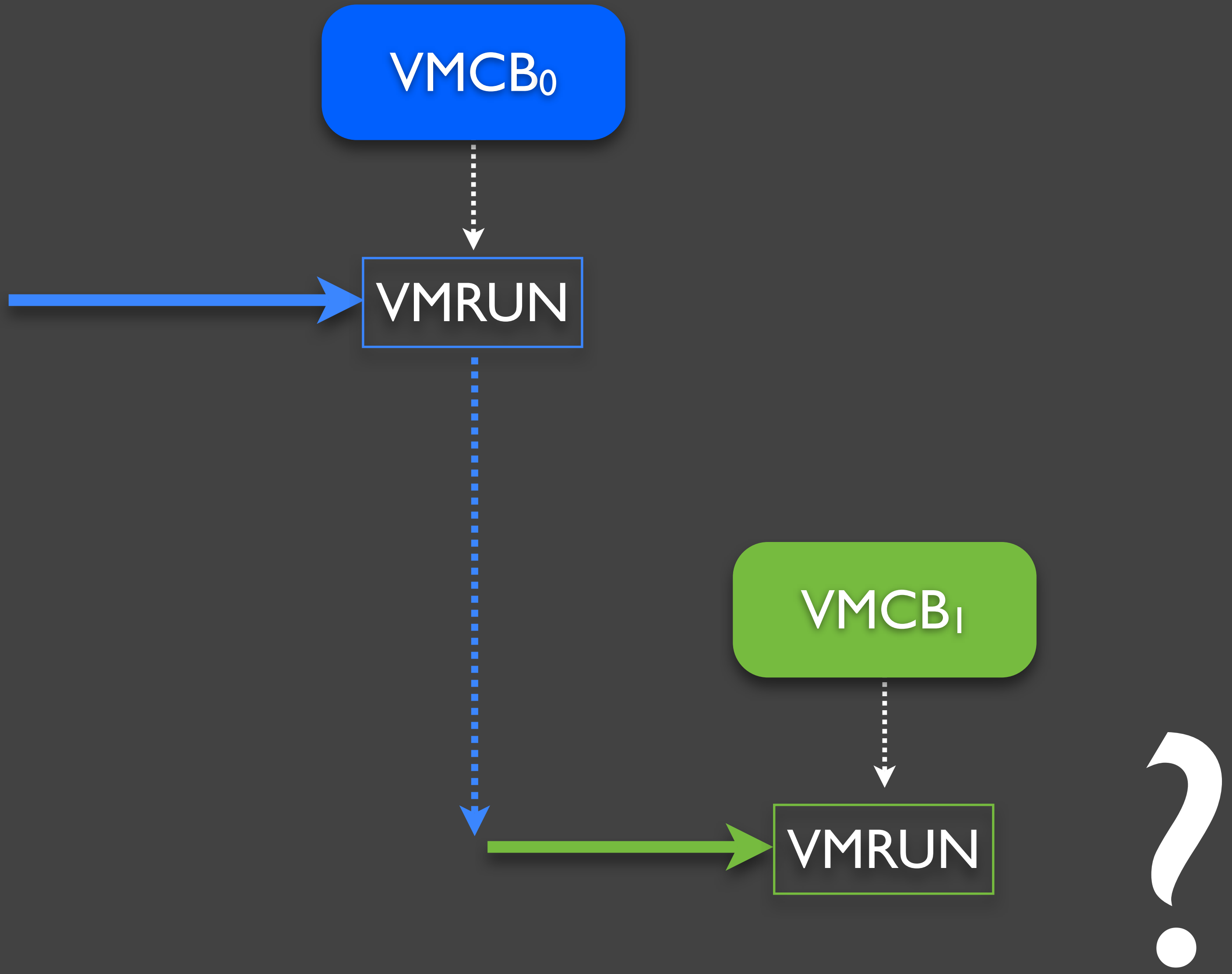
VMCB$_0$

VMCB$_0$

VMRUN

VMRUN

RDMSR



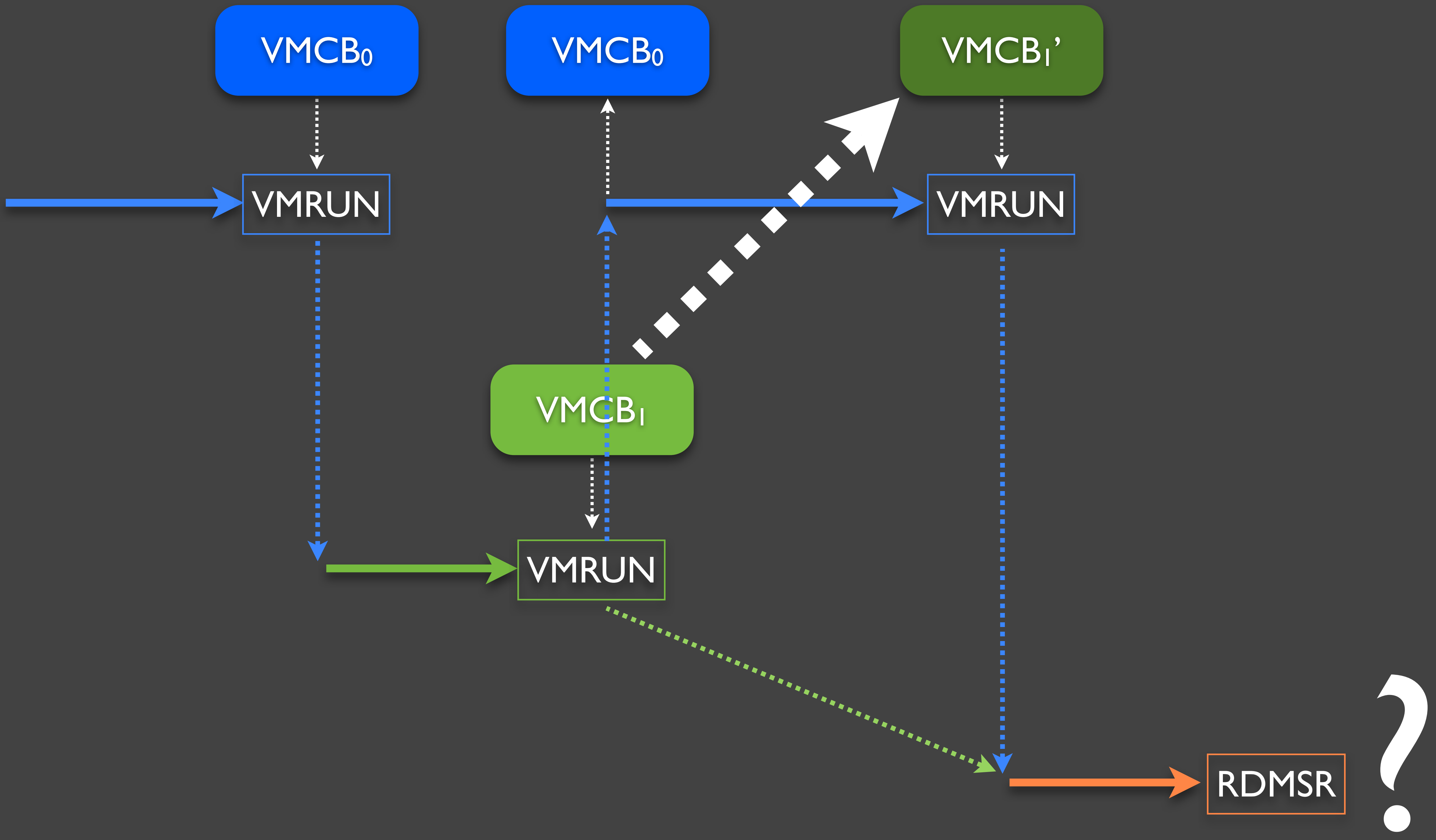
VMCB$_0$

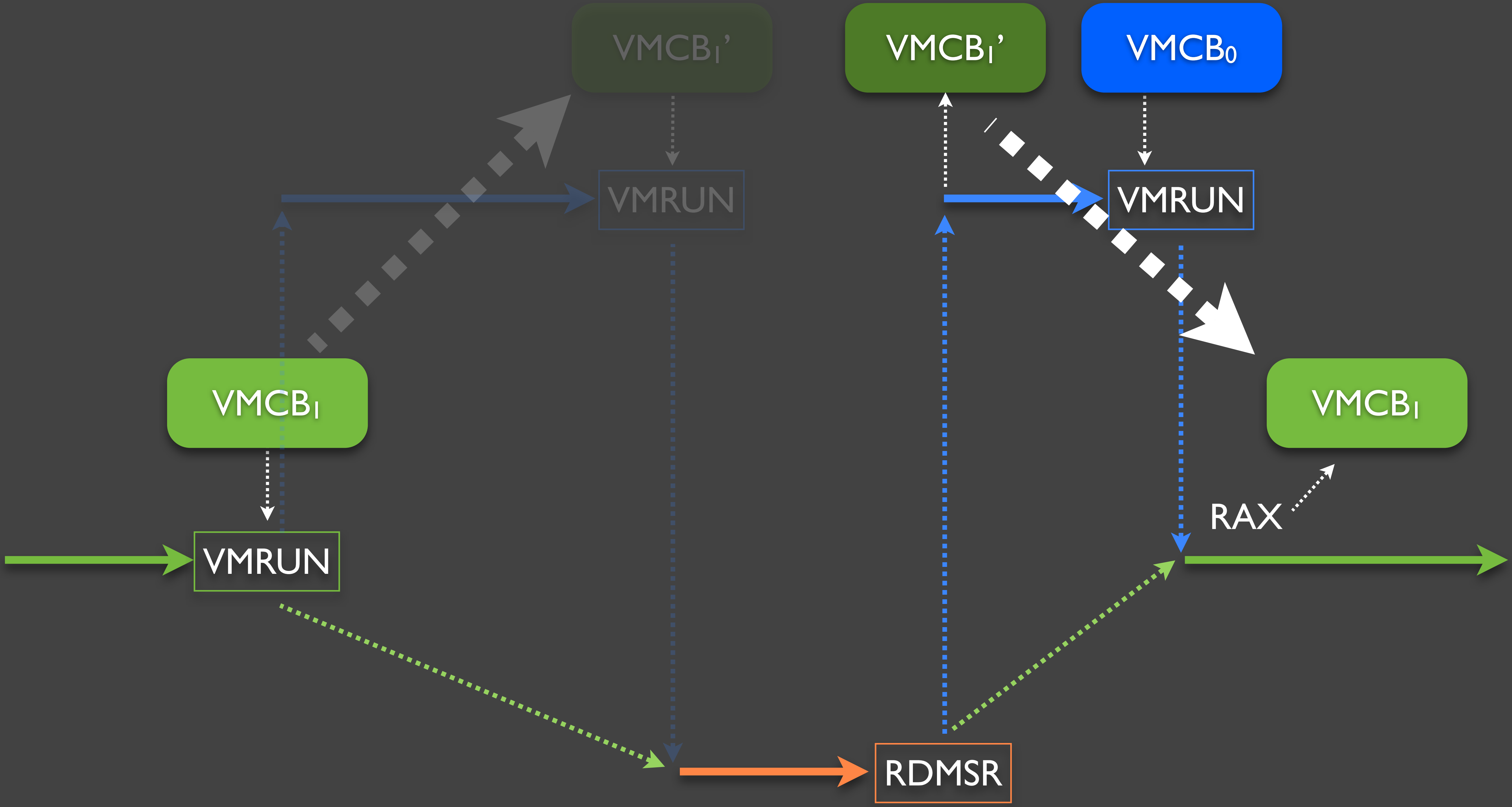
VMRUN

VMCB$_1$

VMRUN

?


VMCB0
VMCB0
VMCB1'
VMRUN
VMRUN
VMCB1
VMRUN
RDMSR
?



VMCB$_1$

VMRUN

VMCB$_1$'

VMRUN

RDMSR

VMCB$_1$'

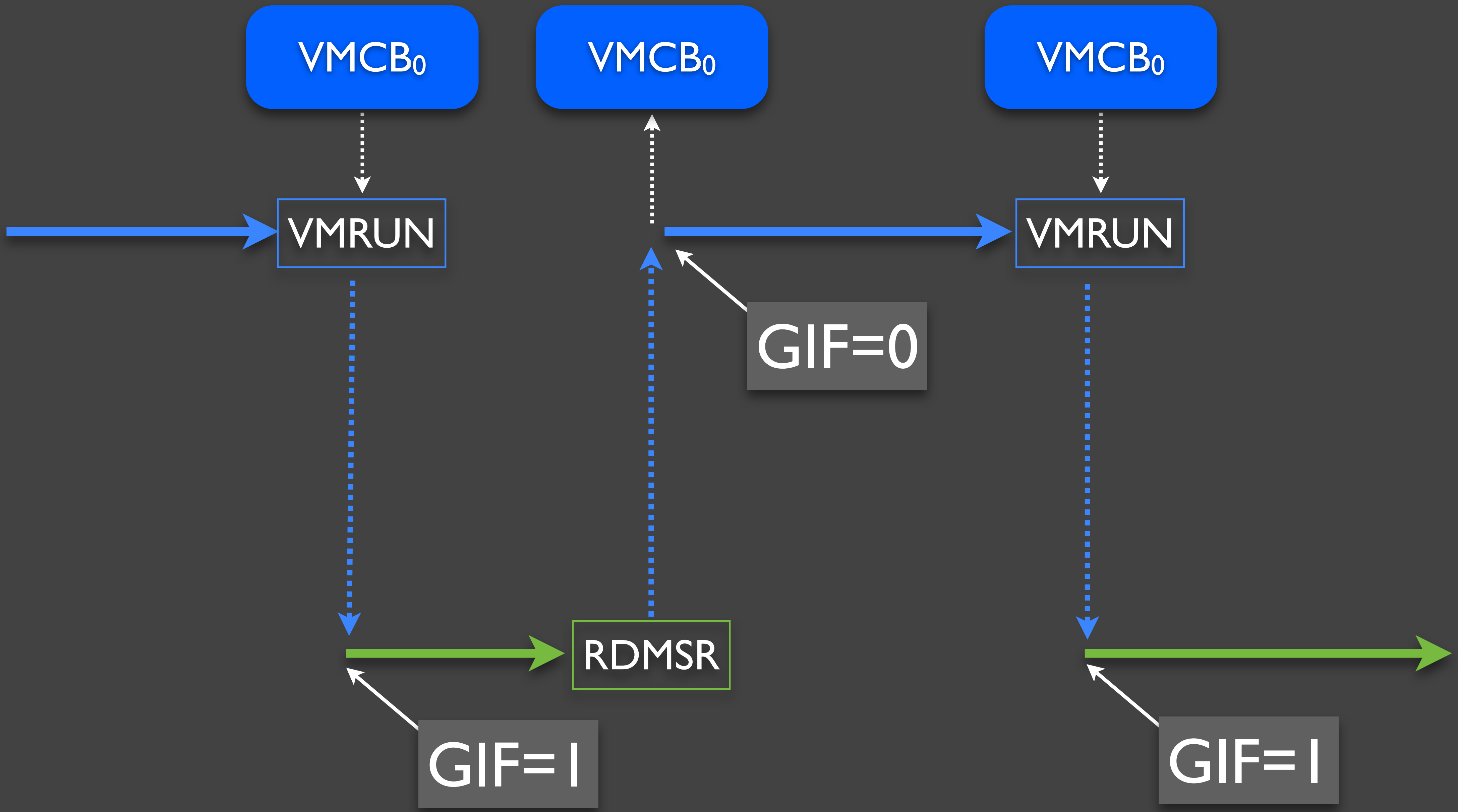
VMCB$_0$

VMRUN

RAX

VMCB$_1$

Looks convincing but won’t work with more complex hypervisors...



VMCB$_0$

VMCB$_0$

VMCB$_0$

VMRUN

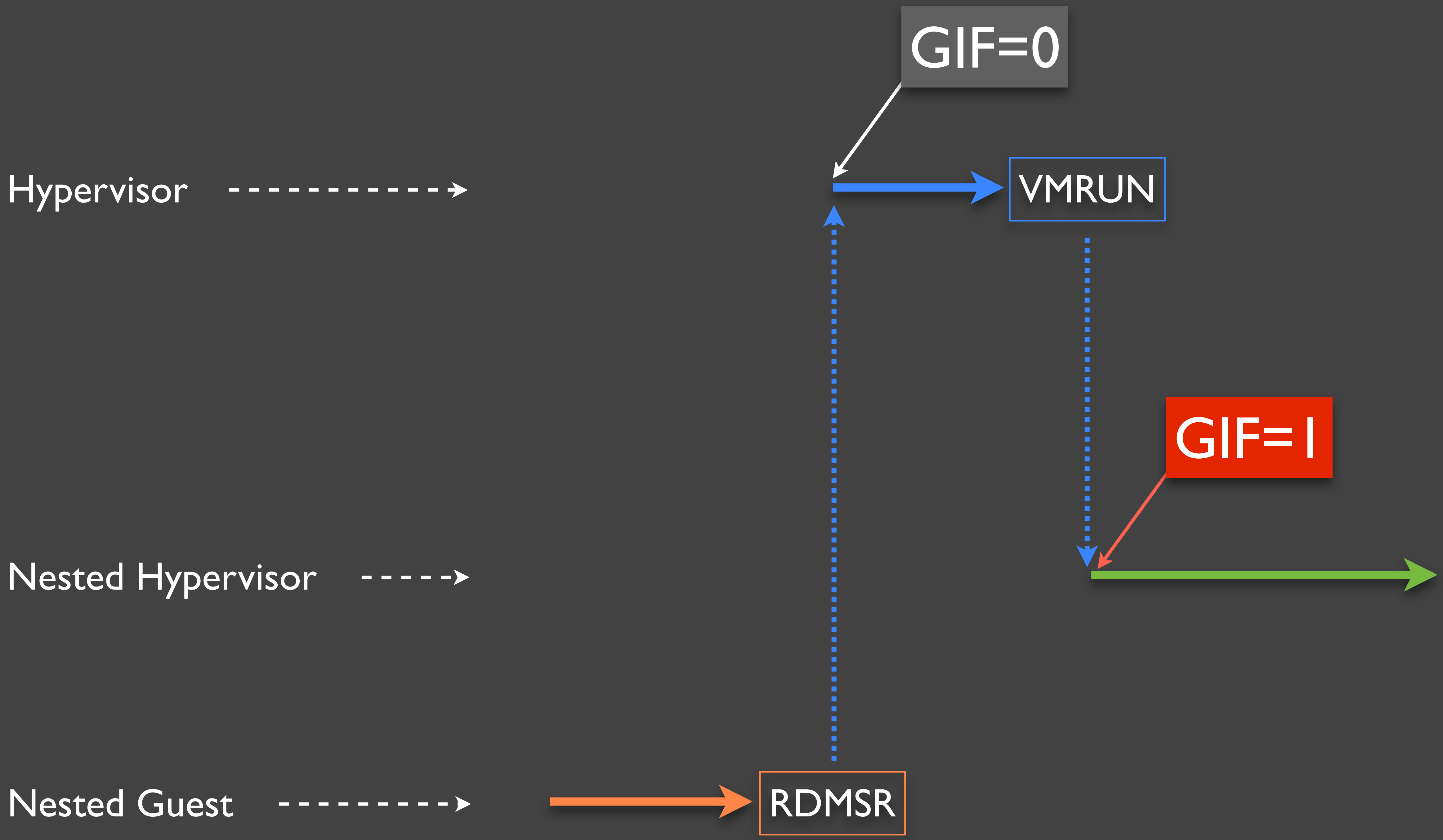
VMRUN

GIF=0

RDMSR

GIF=1

GIF=1



GIF=0
Hypervisor
VMRUN
GIF=1
Nested Hypervisor
Nested Guest
RDMSR

- Hypervisors expect to have GIF=1 when VMEXIT occurs...
    - They might not be prepared to handle interrupts just after VMEXIT from guests!
- ... but when we resume the nested hypervisor CPU sets GIF=1, because we do this via VMRUN, not VMEXIT...

Getting around the "GIF Problem"

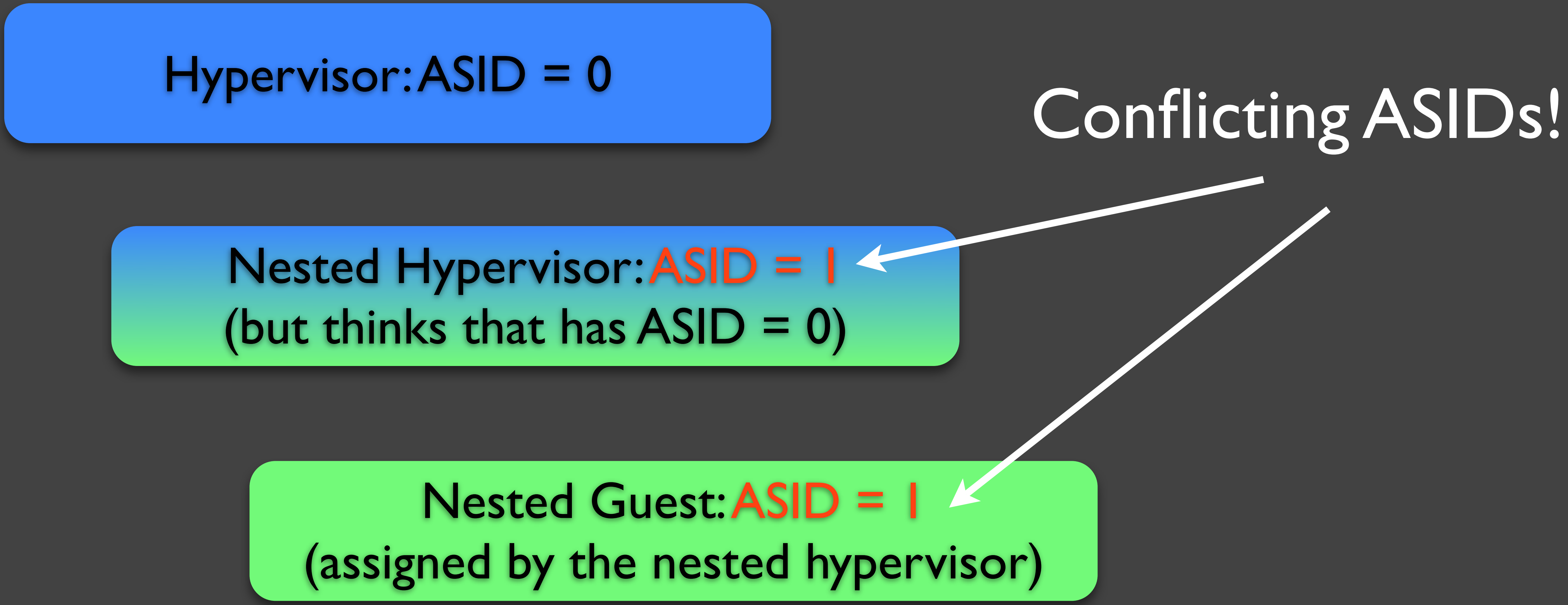
- We need to "emulate" that GIF is 0 for the nested hypervisor
- We stop this emulation when:
  - The nested hypervisor executes STGI
  - The nested hypervisor executes VMRUN
- How do we emulate it?

GIF0 emulation

- VMCB_1'.V_INTR_MASKING = 1
- Host's RFLAGS.IF = 0
- Intercept NMI, SMI, INIT, #DB and held (i.e. record and reinject) or discard until we stop the emulation

Additional details

- Need to also intercept VMLOAD/VMSAVE
- Need to virtualize VM_HSAVE_PA
- ASID conflicts



Hypervisor: ASID = 0

Nested Hypervisor: ASID = 1
(but thinks that has ASID = 0)

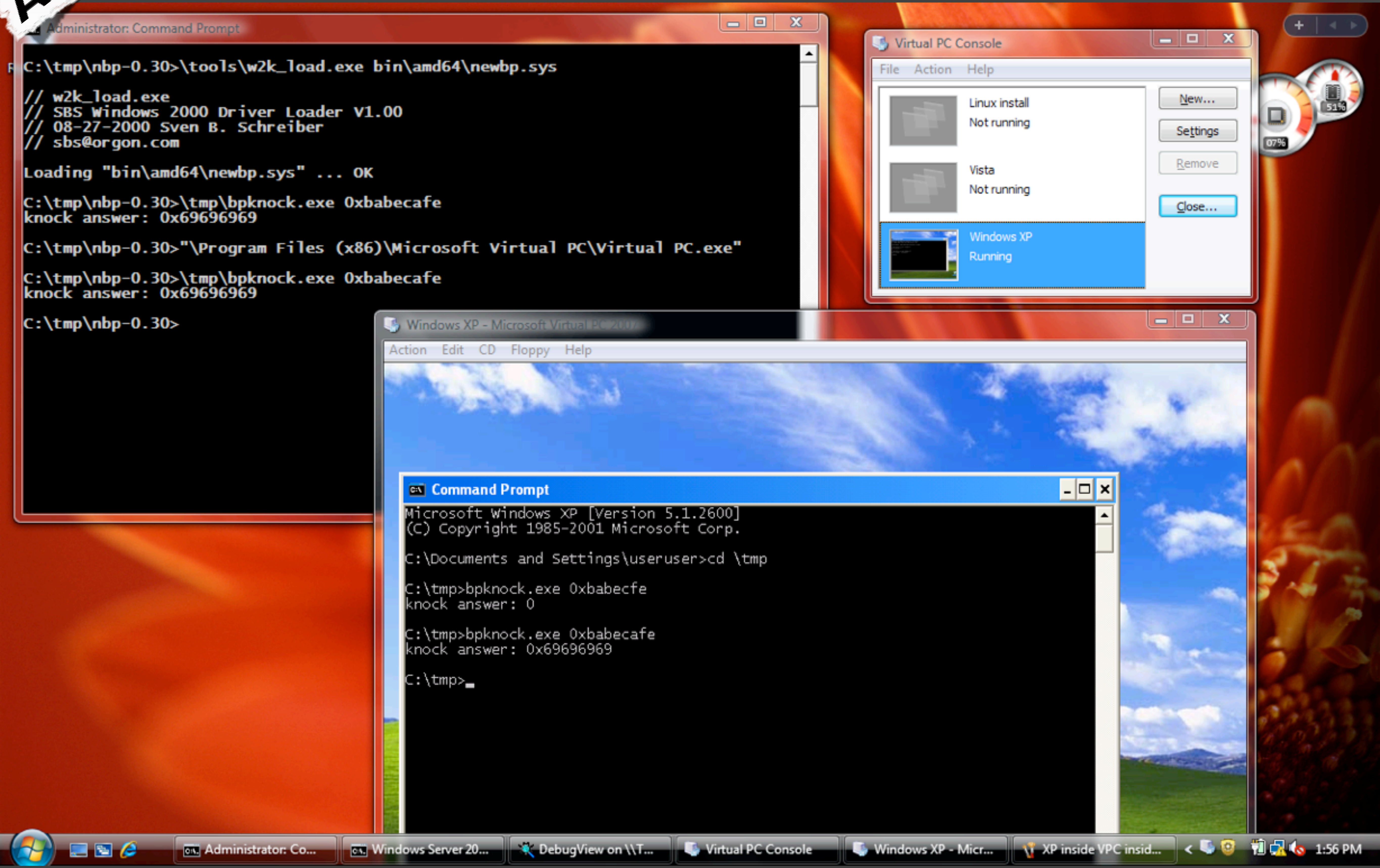
Nested Guest: ASID = 1
(assigned by the nested hypervisor)

Conflicting ASIDs!

But we can always reassign the ASID in the VMCB “prim” that we use to run the nested guest.

Performance Impact

- One additional #VMEXIT on every #VMEXIT that would occur in a non-nested scenario
- One additional #VMEXIT when the nested hypervisor executes: STGI, CLGI, VMLOAD, VMSAVE
- Lots of space for optimization though



```
C:\tmp\nbp-0.30>\tools\w2k_load.exe bin\amd64\newbp.sys

// w2k_load.exe
// SBS Windows 2000 Driver Loader V1.00
// 08-27-2000 Sven B. Schreiber
// sbs@orgon.com

Loading "bin\amd64\newbp.sys" ... OK

C:\tmp\nbp-0.30>\tmp\bpknock.exe 0xbabecafe
knock answer: 0x69696969

C:\tmp\nbp-0.30>"\Program Files (x86)\Microsoft Virtual PC\Virtual PC.exe"

C:\tmp\nbp-0.30>\tmp\bpknock.exe 0xbabecafe
knock answer: 0x69696969

C:\tmp\nbp-0.30>
```

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\useruser>cd \tmp

C:\tmp>bpknock.exe 0xbabecfe
knock answer: 0

C:\tmp>bpknock.exe 0xbabecafe
knock answer: 0x69696969

C:\tmp>_
```

http://bluepillproject.org

How AMD could help?

- AMD could add an additional field to VMCB: “EmulateGif0ForGuest”
- Additionally: virtualize STGI and CLGI when the above field is set to improve performance
- Seems simple to do: just a few additional lines in the microcode... :)

Further thinking...

- Virtualizing DEV for the nested hypervisor that makes use of DEV?
- Virtualizing IOMMU for the IOMMU-aware nested hypervisor?
- Virtualizing Nested Paging mechanism for the NP-aware nested hypervisor?

How about Intel VT-x?

Nested virtualization on VT-x

- No GIF bit - no need to emulate “GIF0” for the nested hypervisor :)
- No Tagged TLB - No ASID conflicts :)
- However:
  - VMX instructions can take memory operands - need to use complex operand parser
  - No tagged TLB - potentially bigger performance impact

Nested VT-x: Status

- We “pretty much” have that working already
- Code is messy and should be rewritten
    - e.g. the operand parser

What Intel could do?

- Extend info provided by:

  VMCS.VMX_INSTRUCTION_INFO

  So that we don't need to parse memory operand manually
- Tagged TLB for better performance
- Other optimization?

Who else does Nested (hardware-based) Virtualization?

IBM z/VM hypervisor on IBM System z™ mainframe

"Running z/VM in a virtual machine (that is, z/VM as a guest of z/VM, also known as "second-level" z/VM) is functionally supported but is intended only for testing purposes for the second-level z/VM system and its guests (called "third-level" guests)."

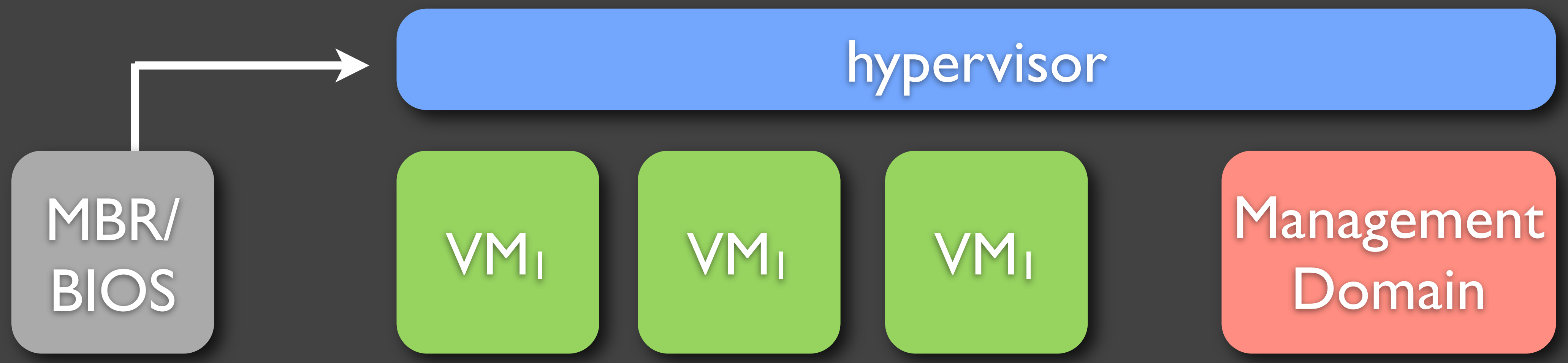
-- http://www.vm.ibm.com/pubs/hcsf8b22.pdf

IBM System z10, source: ibm.com

Confusion

- AMD Nested Page Tables != Nested Virtualization!
- NPT is a hardware alternative to Shadow Page Tables (a good thing, BTW)
- NPT is also called: Rapid Virtualization Indexing

Nested Virtualization: Security Implications



hypervisor

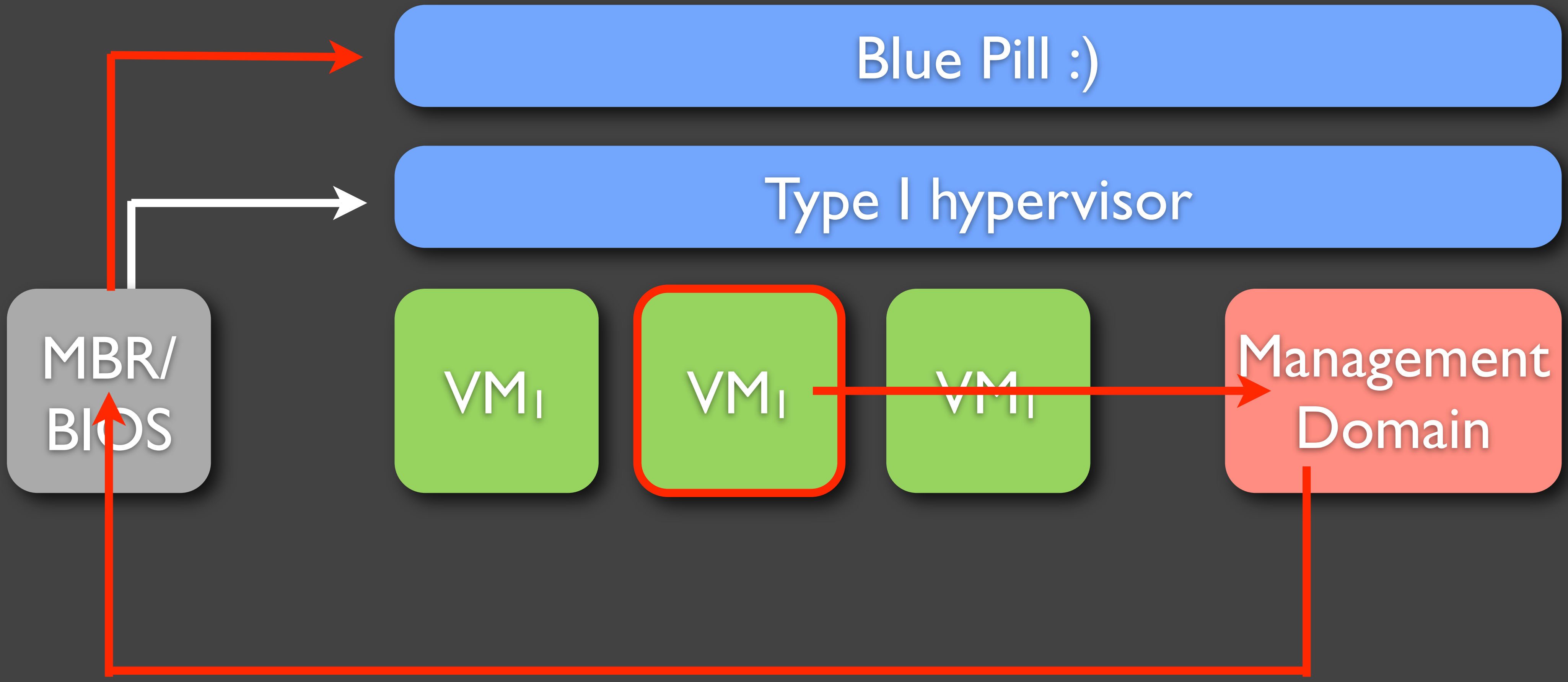
MBR/
BIOS

VM$_1$

VM$_1$

VM$_1$

Management
Domain



Blue Pill :)

Type I hypervisor

MBR/ BIOS

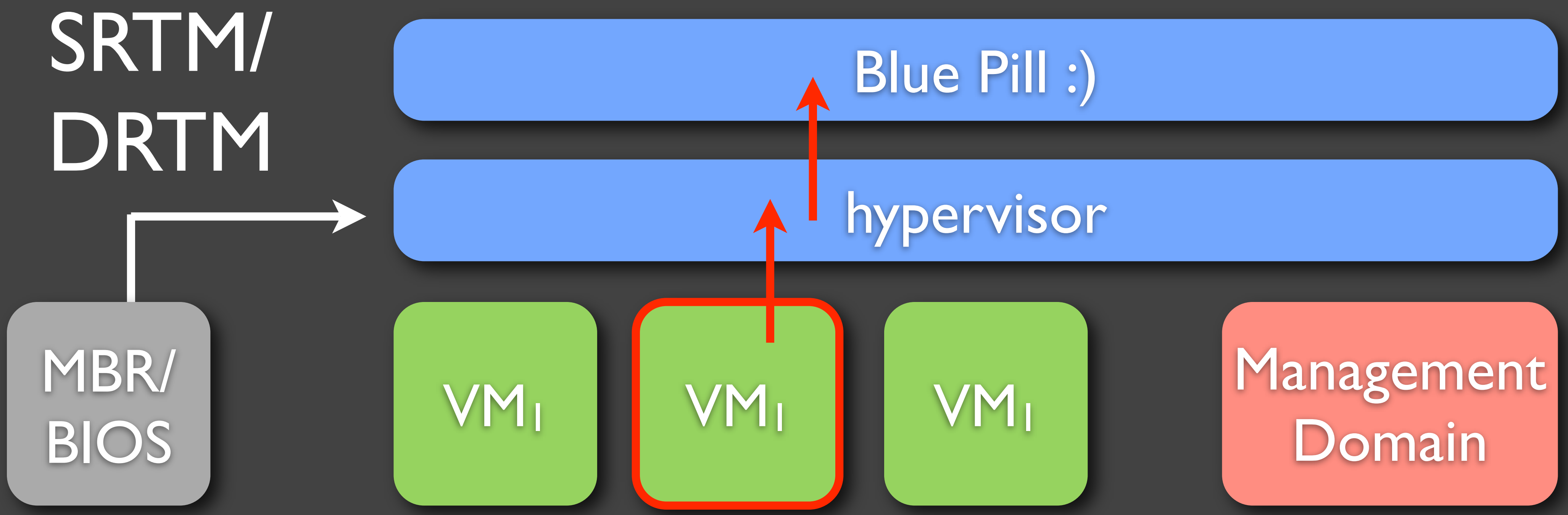
VM$_I$

VM$_I$

VM$_I$

Management Domain

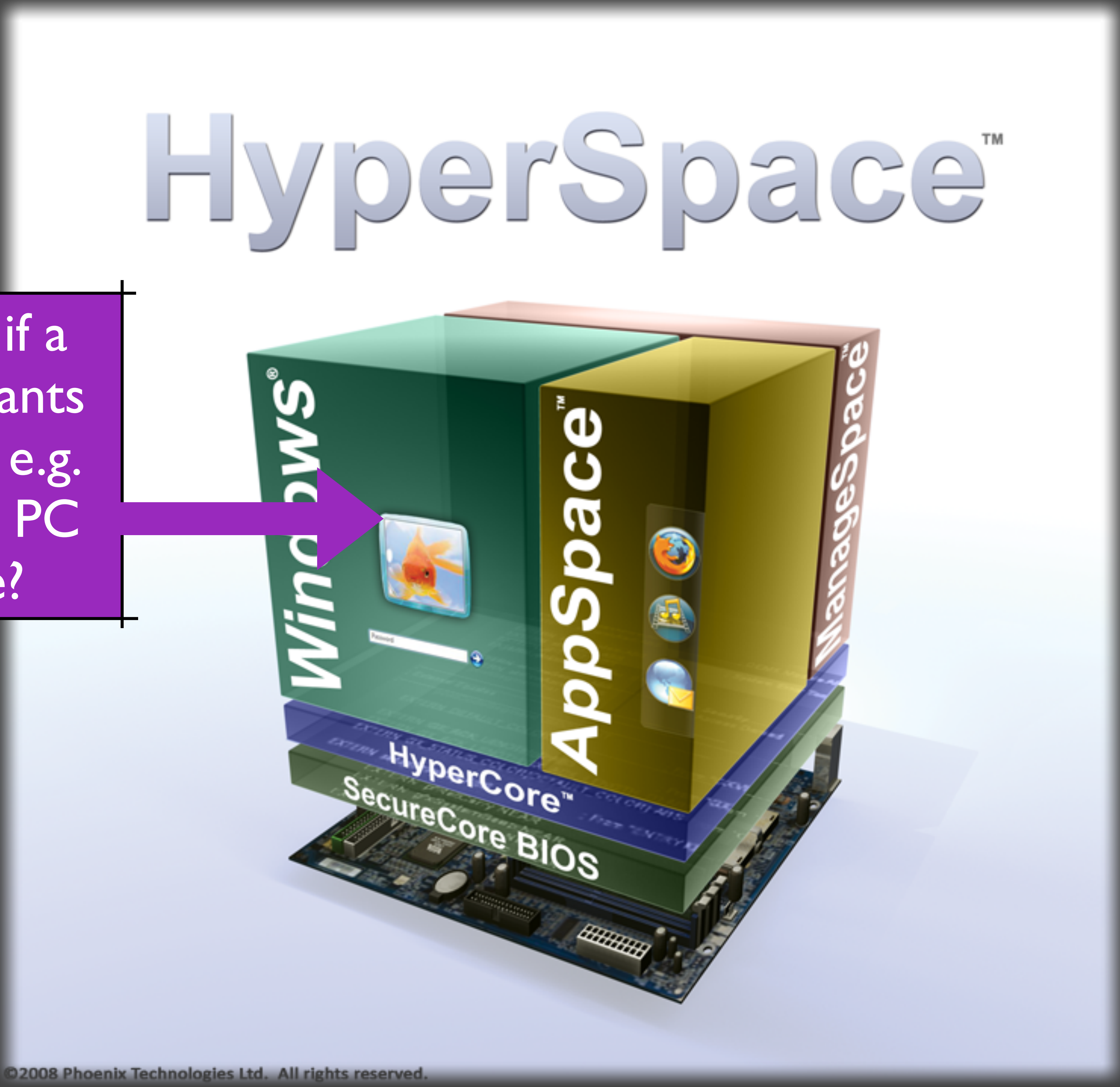
Solution: ensure hypervisor integrity via SRTM or DRTM



SRTM/DRTM do not protect the already loaded hypervisor, from being exploited if it is buggy!

Keep hypervisors very slim!
Do not put drivers there!

Nested Virtualization:
Useful Applications



©2008 Phoenix Technologies Ltd. All rights reserved.

What if a user wants to run e.g. Virtual PC here?

Phoenix Technologies has supported the research on nested hypervisors since Fall 2007

1 Virtualization-based **MALWARE**

2 Using Virtual Machines for **ISOLATION**

3 **NESTED** virtualization

Summary

- Virtualization technology could be used to improve security on desktop systems
- However there are non-trivial challenges in making this all working well...
- ... and not to introduce security problems instead...
- Virtualization is cool ;)

Invisible Things Lab

http://invisiblethingslab.com