

**If k8ns makes you uncomfortable
You're going to enjoy the rest**

**K8ns:
the inevitable result
of dedication to
the inner platform effect**

OOPS WE DID IT AGAIN
WE FORGOT ABOUT STORAGE
AND NETWORKING TOO

What we aren't talking about



What Kubernetes is in detail



Boiling the ocean



I've missed some stuff



Come educate me!

A silhouette of a person standing on a dark horizon, with their arms raised in a 'V' shape. The background is a vibrant sunset sky with warm orange and yellow tones, transitioning to a darker blue at the top. The person's shadow is cast on the ground in front of them.

Welcome to a story of wanderlust

For adventure

For growth

For insanely poor decisions

So I asked GPT to write my intro

Kubernetes is a platform for managing containerized applications across a cluster of nodes. It automates the deployment, scaling, and maintenance of these applications, and provides features such as service discovery, load balancing, storage orchestration, and self-healing. Kubernetes is based on Google's experience with Borg, and is open-source and extensible. Kubernetes is widely used for cloud-native and hybrid-cloud workloads, and has a large and growing ecosystem of tools and services.

A group of business professionals in a meeting, looking at a tablet. The image is dimly lit and has a dark overlay. The text "Let's talk about SharePoint" is centered in white.

Let's talk about SharePoint



“The inner platform effect”



When you attempt to design something so general that you essentially re-implement the tool you're using to build your system, but worse in every conceivable way.

What about the other stuff



SharePoint is an application server

That comes with collaboration storage out of the box



SalesForce is an application server

That comes with CRM out of the box



Kubernetes is a state management backplane

That comes with container management out of the box

Let's start with some questions

What tech won the virtualization wars?

- The real winner was inside us all along

Who likes Kubernetes?

What are the two kinds of multitasking?

- Bad and worse

Who had a 386 that wasn't their first computer?

- What did you think of it?

A large, fluffy brown teddy bear is the central focus of the image, sitting on a wooden floor. The bear is made of soft, shaggy fur and has a friendly expression. The background is a plain, light-colored wall. The text is overlaid on the right side of the bear.

It is now time for our
Build-a-Bear
workshop
but for workload
orchestration

The stuffing is made out of real bear



The history of processes

1960



TRANSISTOR COMPUTING



IBM SYSTEM/360

1971: It's pronounced "Unnix"



MULTI-SESSION TIME-SHARING
MAINFRAMES



COMPUTING EXPERIENCES THAT
WOULD BE FAMILIAR TODAY



A brief 11-year interlude
where not much happened

1982

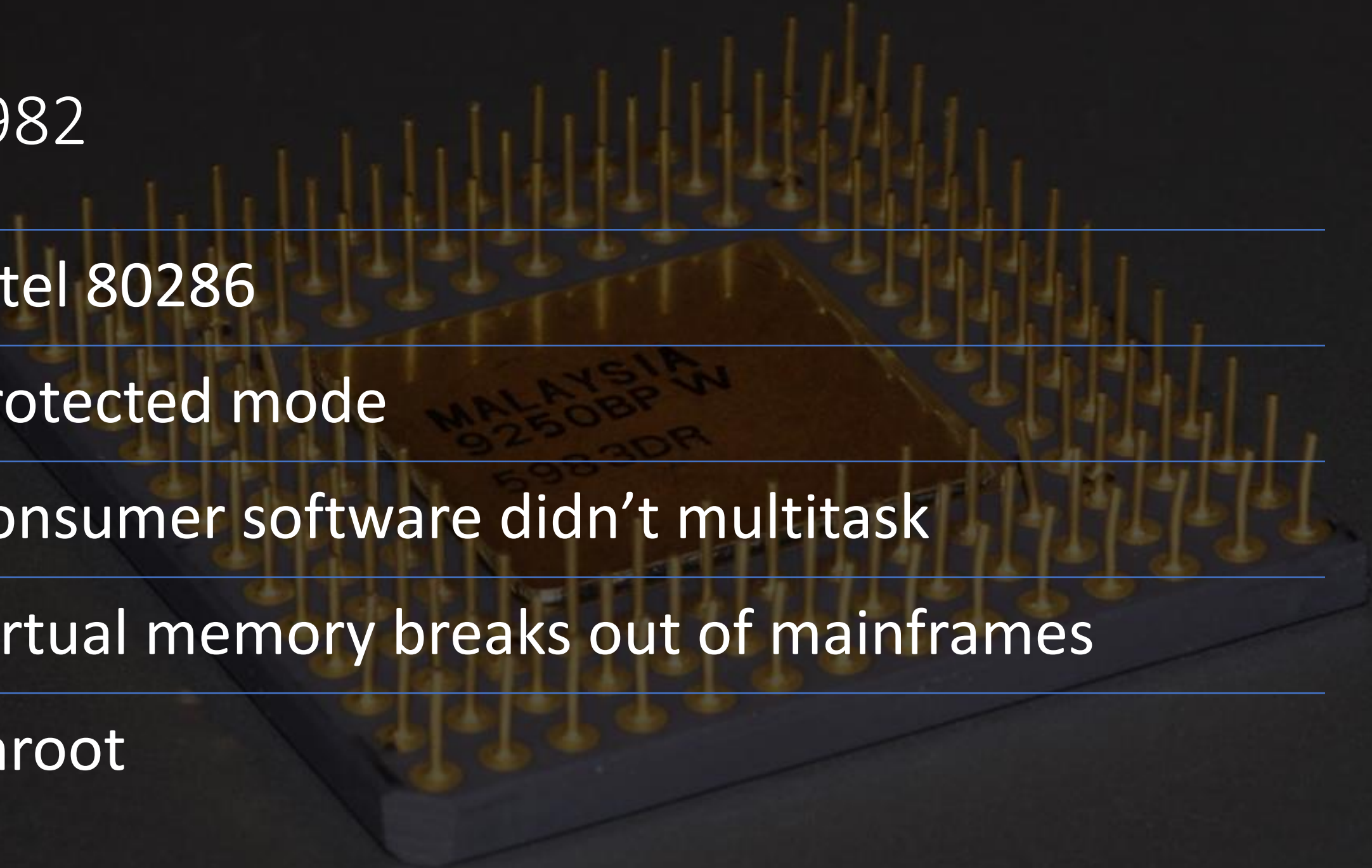
Intel 80286

Protected mode

Consumer software didn't multitask

Virtual memory breaks out of mainframes

chroot





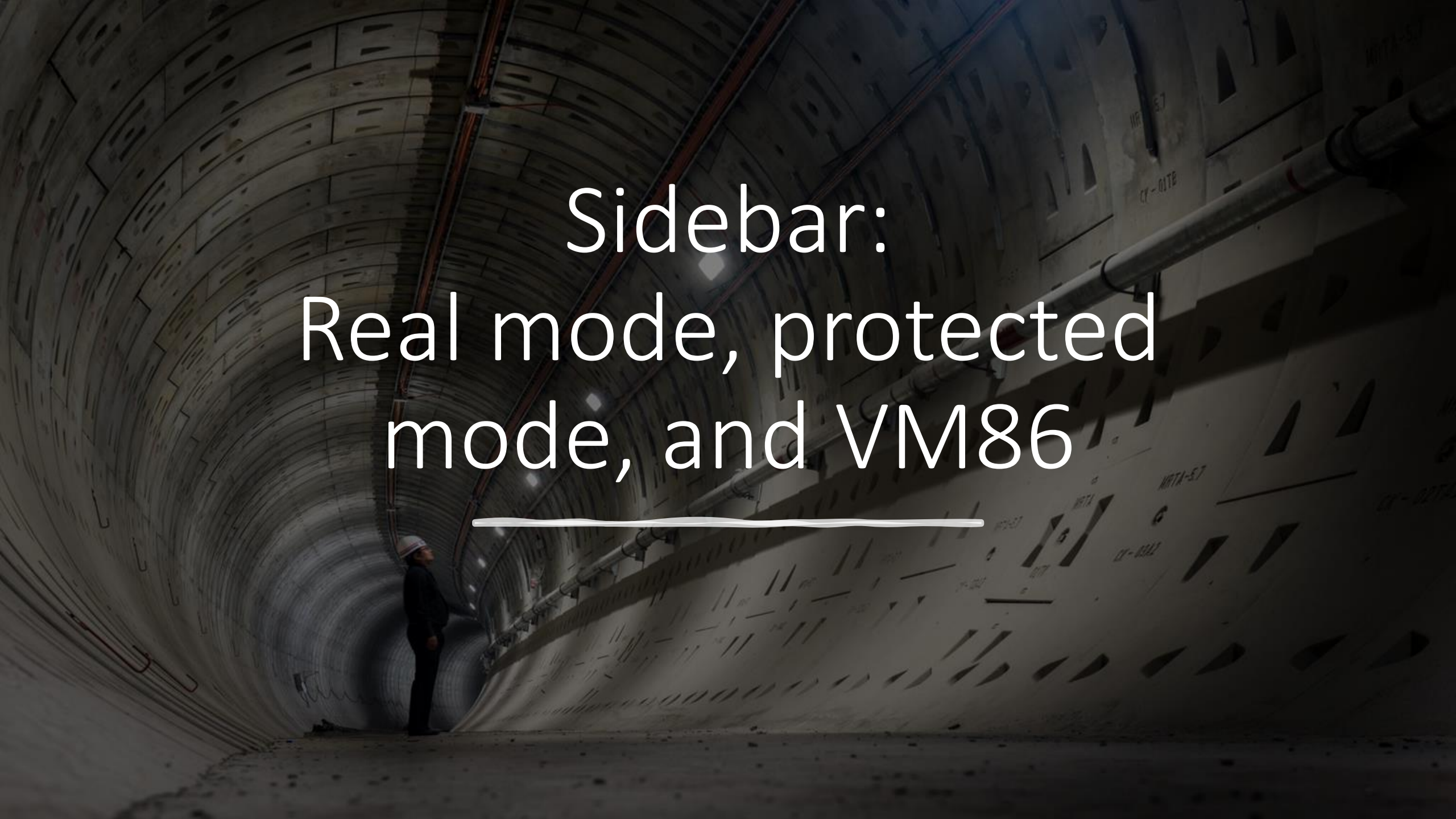
1985

Intel 386

Virtual memory actually works

The first consumer VM: VM86

Enabled basic cooperative multitasking

A large, dimly lit tunnel under construction. The walls are lined with concrete segments, and various pipes and cables run along the ceiling. A person wearing a hard hat and dark clothing stands in the distance on the left side of the tunnel, providing a sense of scale. The overall atmosphere is industrial and somewhat somber due to the low lighting.

Sidebar: Real mode, protected mode, and VM86

Real (slim shady) mode



NAÏVE COMPUTING



EVERY PROCESS CAN ACCESS
AND USE EVERYTHING



THIS IS DANGEROUS FOR
MULTITASKING

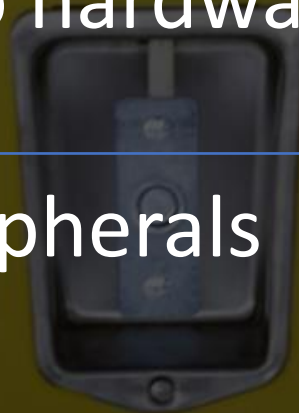
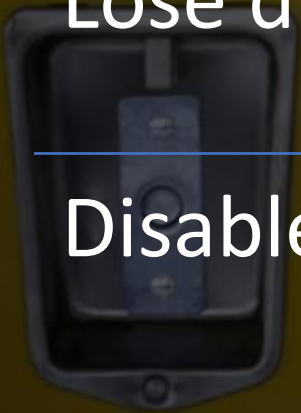
Protected Mode

Provide each process a sandbox for memory

Gate access to shared resources through a parent process

Lose direct mapping to hardware as the tradeoff

Disables access to peripherals

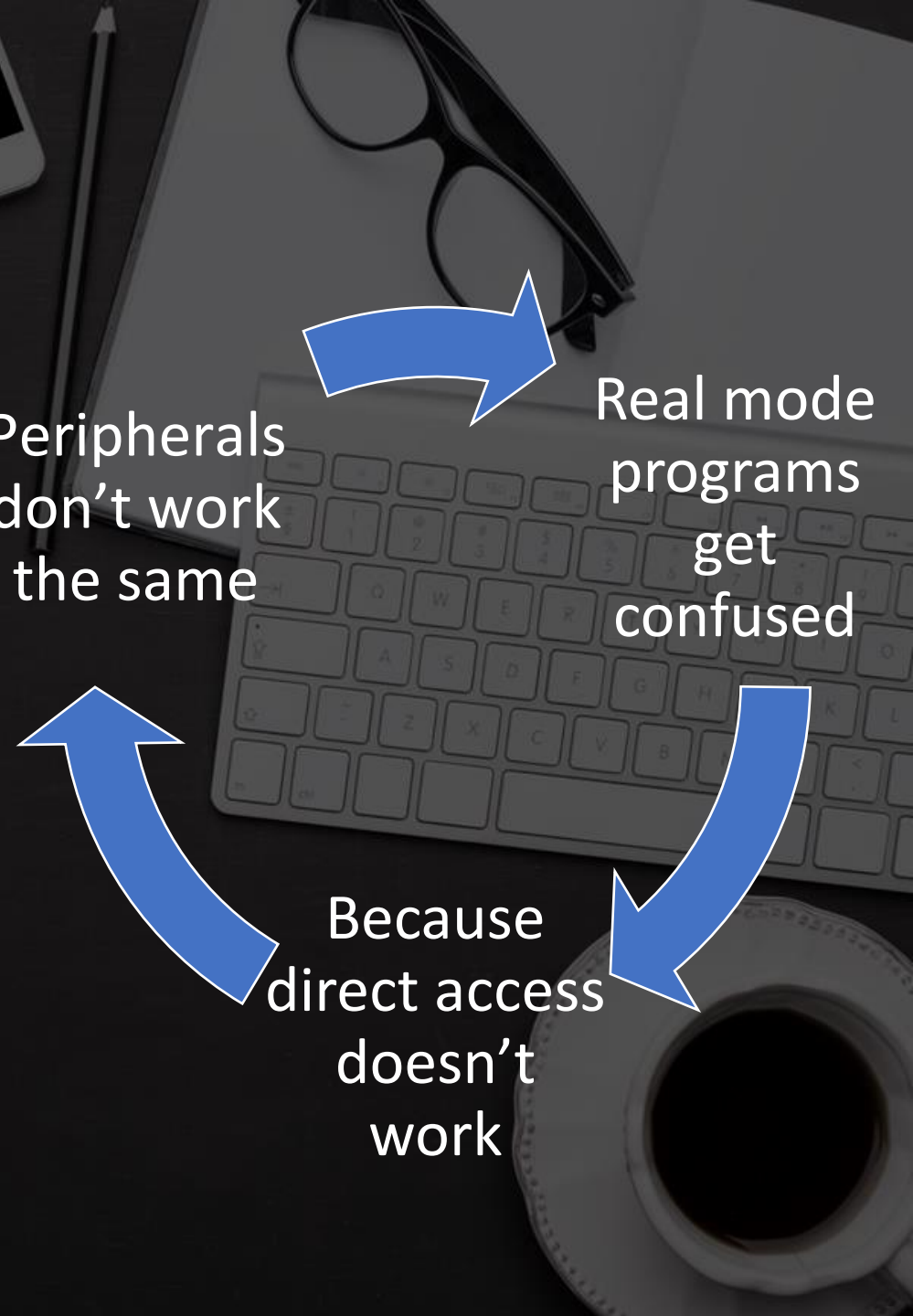


Challenges of Protected Mode

Peripherals don't work the same

Real mode programs get confused

Because direct access doesn't work



VM86 mode for protected processes



Exposes all functions in protected mode



Traps sensitive instructions

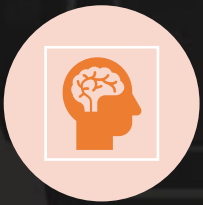


Traps trigger a monitor for the VM86 process

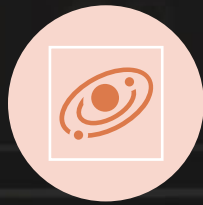


Monitor proxies functionality

1992



PREEMPTIVE
MULTITASKING
COMING IN
WAVES



SOLARIS
2.0/SUNOS 5.0



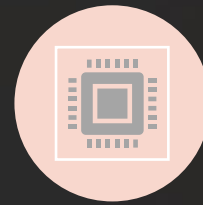
WINDOWS 3.1
PROTECTED/EN
HANCED



MAC OS
COOPERATIVE
MULTITASKING



LINUX NOT YET
ON THE SCENE



BUILDING ON
WORK FROM
MAINFRAMES,
MOTOROLA,
AMIGA



What made
the Pentium
special?

"WEIRD AL" YANKOVIC



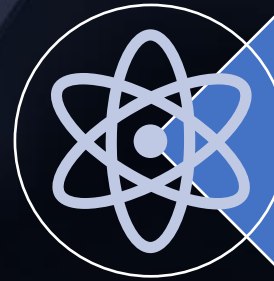
What made the Pentium special?



MMU
Shadowing



Superscalar
architectur
e



CMPXCHG
is born!



Why was Windows NT special?

Full preemption in user and kernel space

1993



PENTIUM IS
RELEASED



ENABLES EFFICIENT
MMU SHADOWING



EFFICIENT PAGE
TABLE MAPPING



WINDOWS NT



Why was Windows 95 so good?



Full preemption for
consumer use



Driving mainframe
stability, performance,
responsiveness to the
home



1995



Windows 95 for consumer use



Mac would need another <Soon™>
years



1999

VMWare and others are taking on virtualization full tilt

Take trap-and-emulate for x86 and do it in software



Sidebar: Virtualization



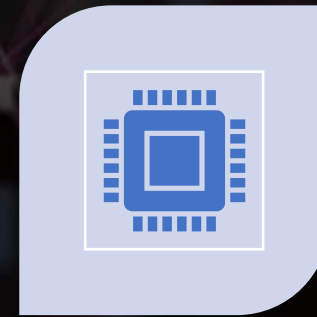
Types of virtualization



FULL VIRTUALIZATION



PARA-
VIRTUALIZATION



HARDWARE-ASSISTED
VIRTUALIZATION

Full virtualization



Simulate everything a real
computer has



Trap problematic instructions
where possible



Translate binaries to mitigate
problem instructions



Binary translation

X86 has so many instructions

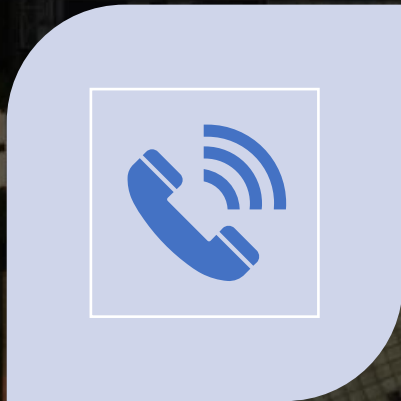
And no hardware for them

Many pierce the virtualization veil

Translate them into a new set of instructions

New instructions are benign or trappable

What if some parts were virtualization aware



TRAPPING CALLS TO IO IS NOT
PERFORMANT



WHAT IF WE USED VIRT-AWARE DRIVERS
(VMWARE TOOLS)



DISK, NETWORKING, PERIPHERALS

Para-Virtualization

What if the entire OS was aware

Compiled not to contain problematic instructions

Cooperatively trapped itself



Hardware assisted virtualization

What if the
hardware traps
were added



Hardware traps
are better, but
not free



Made PV
unnecessary, but
still beneficial

Popek and Goldberg

What is needed for virtualization?

Sensitive instructions

Privileged instructions



A comparison of architectures



PowerPC

- Everything works the same in system and user modes, or not at all

A comparison of architectures

PowerPC

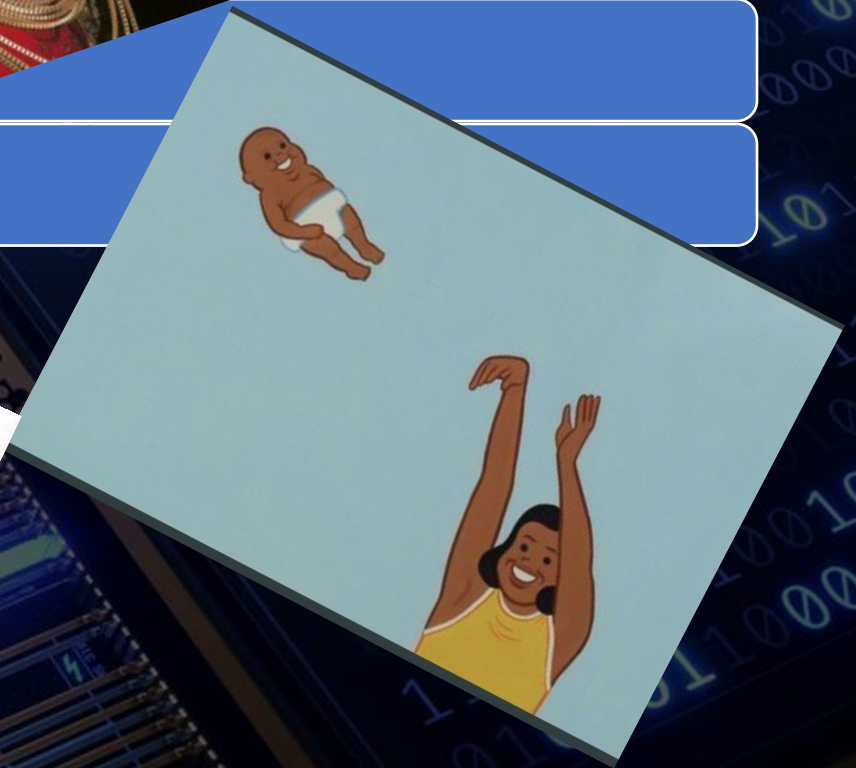
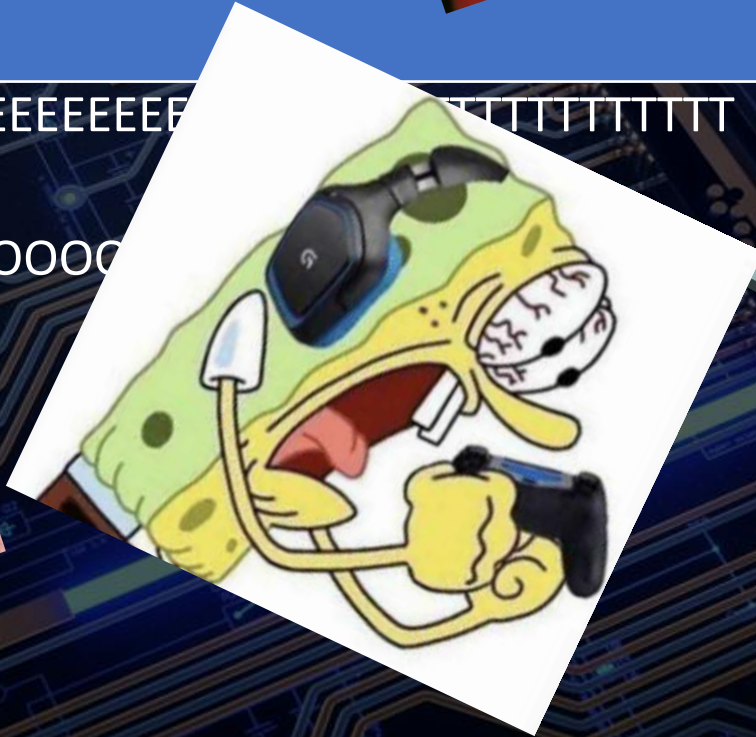
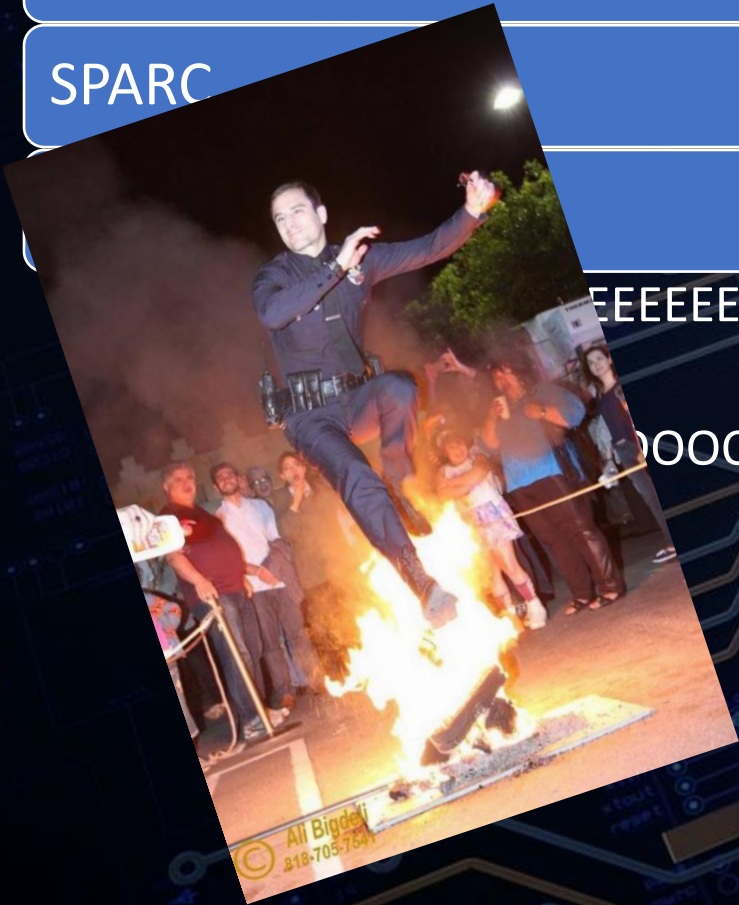
SPARC

- Designed from the start not to make virtualization harder in the future

A comparison of arch

PowerPC

SPARC



2000
for the ones that
survived Y2K

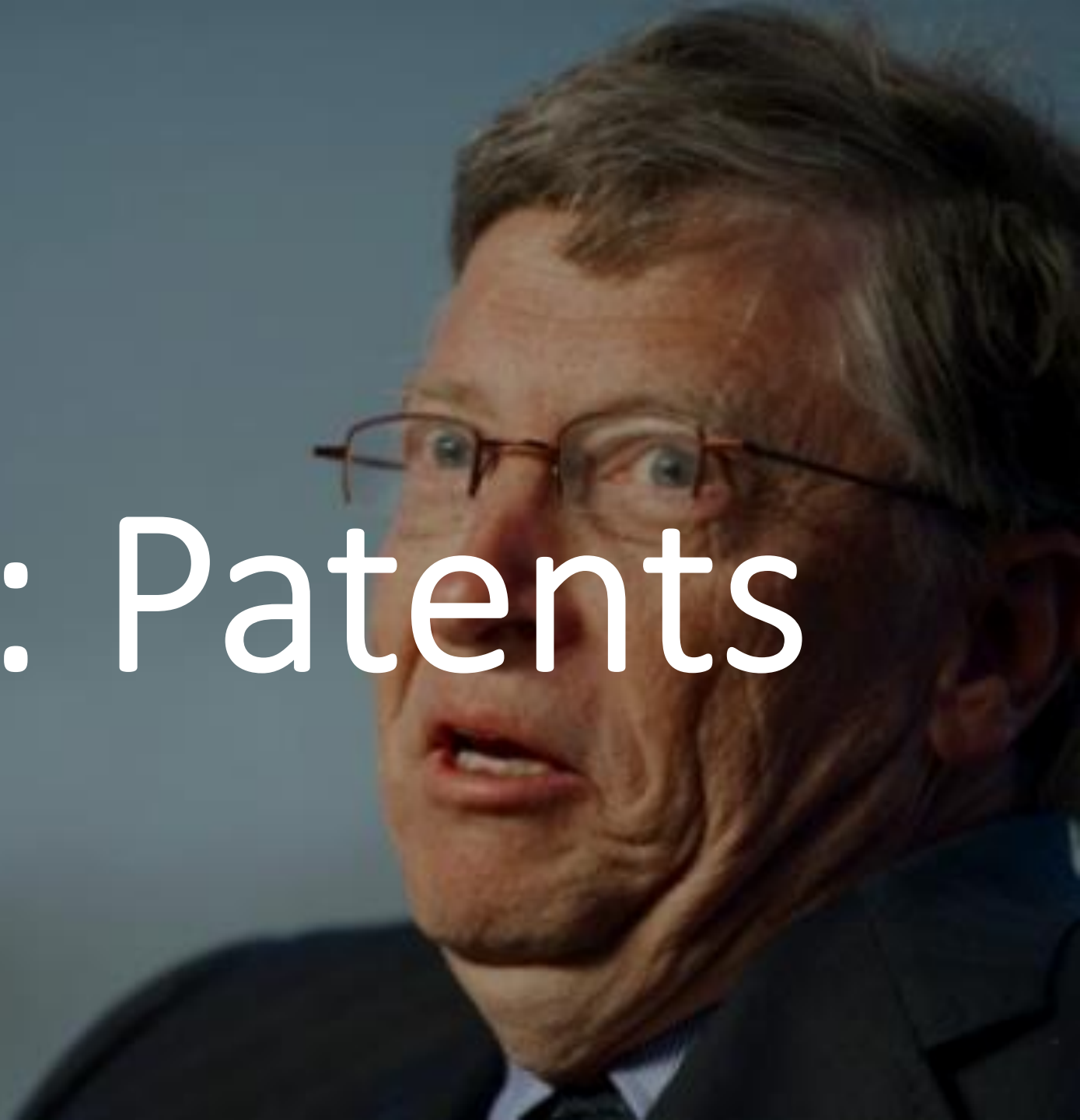
FreeBSD 4.0 adds jail and
makes chroot viable for
PV

VMWare Server released
and explodes

Critical patents for
storage, memory, and CPU
virtualization

Sidebar: Patents

Everyone's favourite topic



Virtualization



US6397242 for VMWare for full virtualization including VMM and binary translation in 1998



US7516247B2 AMD holds several foundational patents on IOMMUs



US7356817B1 to Intel for real-time scheduling of multiple guests with nested scheduling



US7623134B1 GPU hardware page fault management to Nvidia in 2006

Storage

US6928526 to DataDomain for deduplication in 2005

US6289356 to NetApp for snapshotable filesystem in 2001

US7146524 to Isilon for distributed storage with hot spares

US8,266,099 to VMWare for VMFS and clustered storage using the shared storage provider as the compute queue, no server-server communication

US8650359 to VMWare for VVols and storage-servers being VM client aware

US7546307B2 to Nvidia in 2009 for efficient storage of block devices as files in filesystems

Networking

Nvidia holds a pile of networking related patents from 2000-2004


GB2413872A held by Nvidia lays groundwork for something like HPE Moonshot

US7107359B1 to Intel in 2000 for a HFA that can logically partition itself for DMA

Peripherals and accelerators

US6920484B2 held by Nvidia lays some groundwork for PCIe SR-IOV





Why was Linux 2.4 boiled trash?

No preemption in the kernel

2001



Linux gets preemption




Apple sucks less with Mac
OSX mutant love child of
BSD and NeXTSTEP



OS preemption and virtual
memory is a level playing
field now



PCI-SIG ratifies SR-IOV



Sidebar: PCIe SR-10V



SR-IOV

**NETWORK
CARD**



**"NETWORK
CARD?"**

**"ALSO A
NETWORK
CARD?"**

YOU GET A NETWORK INTERFACE



YOU ALL GET A NETWORK INTERFACE

PCIe devices are functions

Functions of the
hardware

You tell hardware
what to do,
it does the thing

But what if you want more than one



What if you could tell the hardware you wanted two of it



You could address each one individually

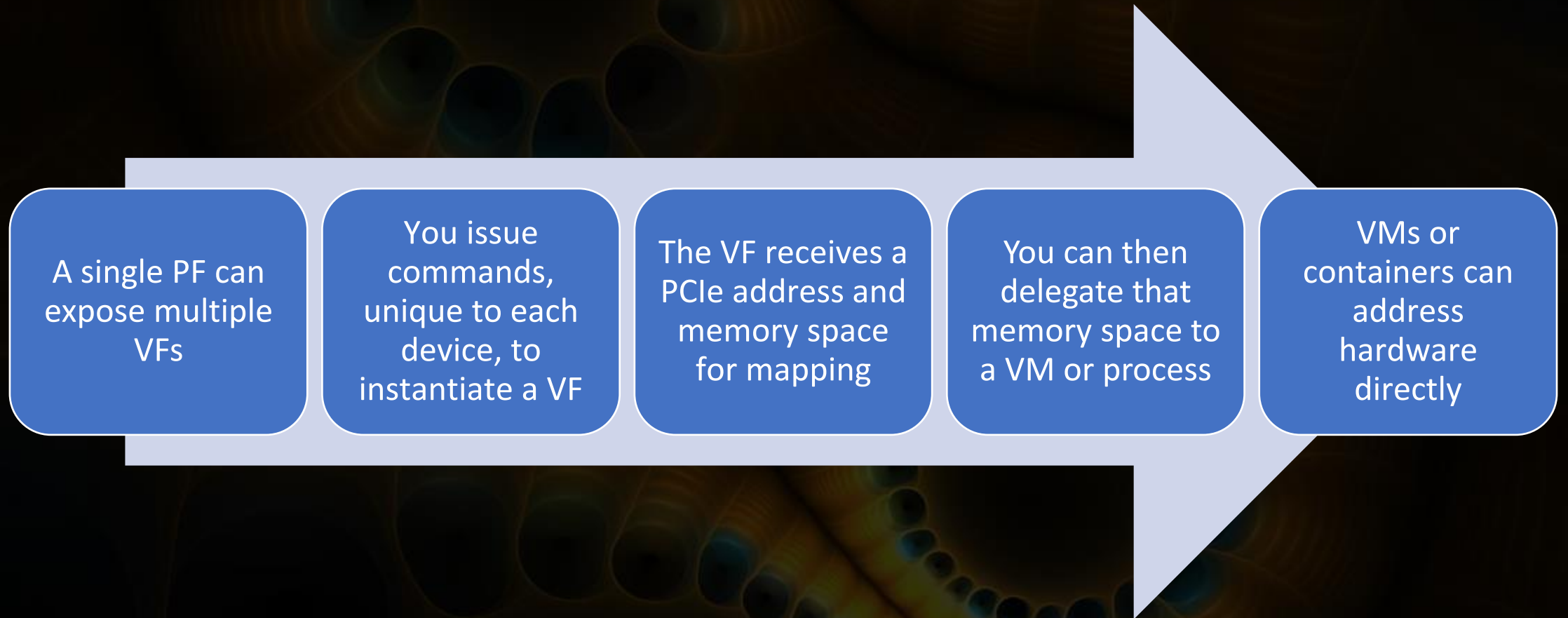


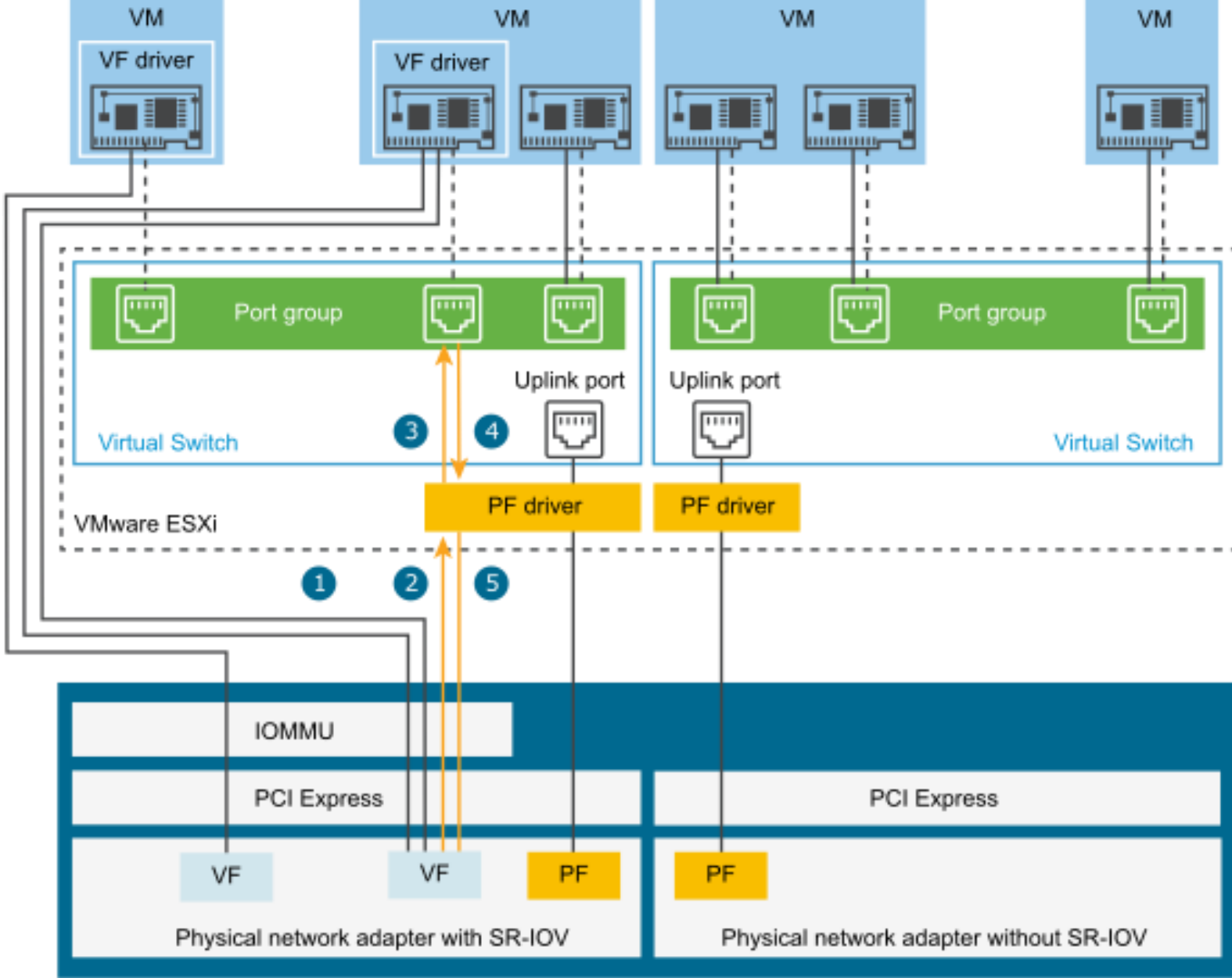
The hardware would sort out how



You just tell it what

PCIe Virtual Functions





Basic PCIe SR-IOV

- Guest workload
- VF Driver
- Guest
- PF Driver
- Hypervisor
- IOMMU
- Virtual functions
- Physical functions
- Physical devices

Examples

Network devices

- Discrete NICs with distinct address scopes

FPGAs

Nvidia GRID on pre-Ampere

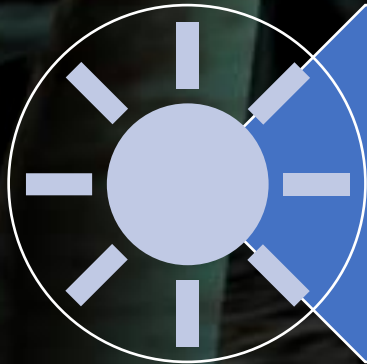
- Needed driver support and configuration for accessing partitions

NVIDIA MIG on Datacentre A100 and newer

- Exposes pieces of the GPU as PCIe devices



Xen hypervisor
released
supporting PV
guests



Solaris zones;
jails with
actual
management!

2004





2005

Intel VT-x included in
Pentium 4 662/672

Xen 3.0 supports HVM
guests

2006



AMD includes AMD-V in Athlon 64, 64 X2, and 64 FX



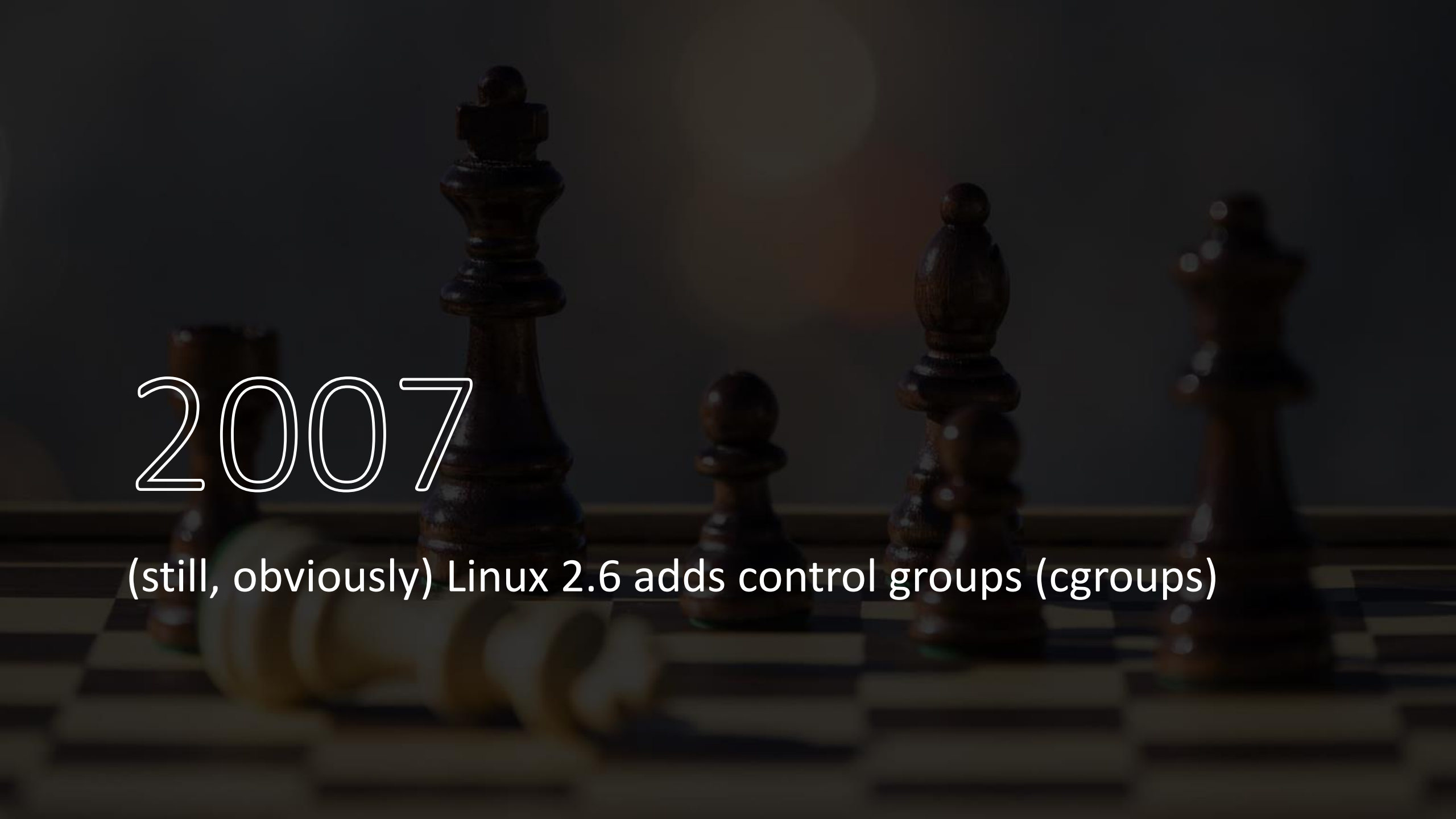
VT-x and AMD-V are first-gen hardware assists



Linux 2.6 adds mount namespace



AWS launches EC2 with Xen PV instances



2007

(still, obviously) Linux 2.6 adds control groups (cgroups)



2008

Finally get SLAT and page table shadowing in hardware

Intel VT-d and EPT in Nehalem

AMD SVM and RVI in Barcelona Opteron

IOMMU for MMU shadowing and directed IO



LXC

No way to remap user
IDs in Linux yet, so
root is root

2008



2010

PCIe SR-IOV available in mass-produced products

Not much happens after this for x86 virtualization hardware

Linux user namespace

Docker wrapping LXC released almost
instantly after

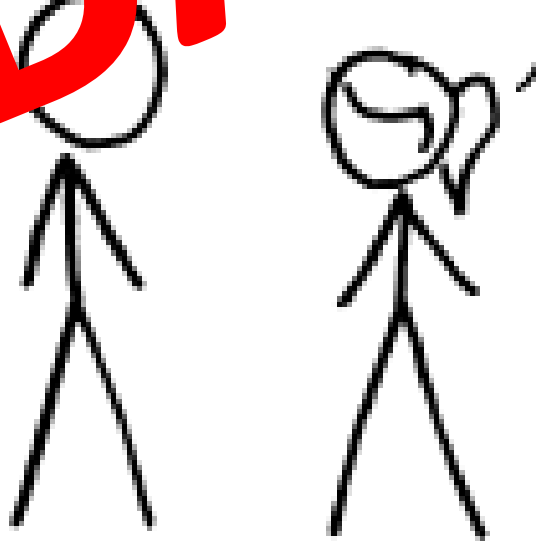
2013



HOW STANDARDS PROLIFERATE: (SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)

SITUATION:
THERE ARE
14 COMPETING
STANDARDS.

14?! RIDICULOUS!
WE NEED TO DEVELOP
ONE UNIVERSAL STANDARD
THAT COVERS EVERYONE'S
USE CASES. YEAH!



SITUATION:
THERE ARE
15 COMPETING
STANDARDS.



STANDARDS

2015



DOCKER SPLITS
CONTAINERD RUNTIME
FROM DOCKER-CLI



OCI IS FOUNDED



OCI RATIFIES CRI



KUBERNETES 1.0
RELEASED

2016

Docker 1.10
drops LXC for CRI
runtimes

OCI releases runc
and libcontainer
as reference
implementations

Docker swarm
splats onto the
scene like a wet
dog

Kubernetes 1.2
formally adopts
OCI standards

also 2016

- Kubernetes realizes workload state is a thing
- Kubernetes 1.3 hold my beer...



2017

Kubernetes still
struggles with
storage

And also
networking

Sorry

Just commit to
master its fine
(in-tree modules)

2018



CSI AND CNI FOR OUT-
OF-TREE STORAGE AND
NETWORKING IS IN BETA



OH.



RIGHT.



BLOCK STORAGE.



HANG ON.



KUBERNETES 1.11 HAS
ALPHA SUPPORT,
THERE, BE HAPPY



3 long years



2021

Major storage providers finally have GA CSIs

2023



Modern Kubernetes can do *literally anything*



..... Cough cough



Its been 63
years

Its been 63 years

And we have finally reimplemented computers

4 times

And made the same mistakes

Every time

Inner platform effect

A blurred high-speed train is shown at a platform, illustrating the inner platform effect. The train is moving quickly, creating a sense of motion blur. The platform is visible on the right side of the frame, and the background is dark and out of focus.

again

How can we put these blocks together?



Bottlerocket (AWS)

Hardened Kubernetes OS



Kata containers (Openinfra, Microsoft, Intel)

Containers in VMs



Firecracker (AWS)

MicroVM segmentation



KubeVirt (RedHat)

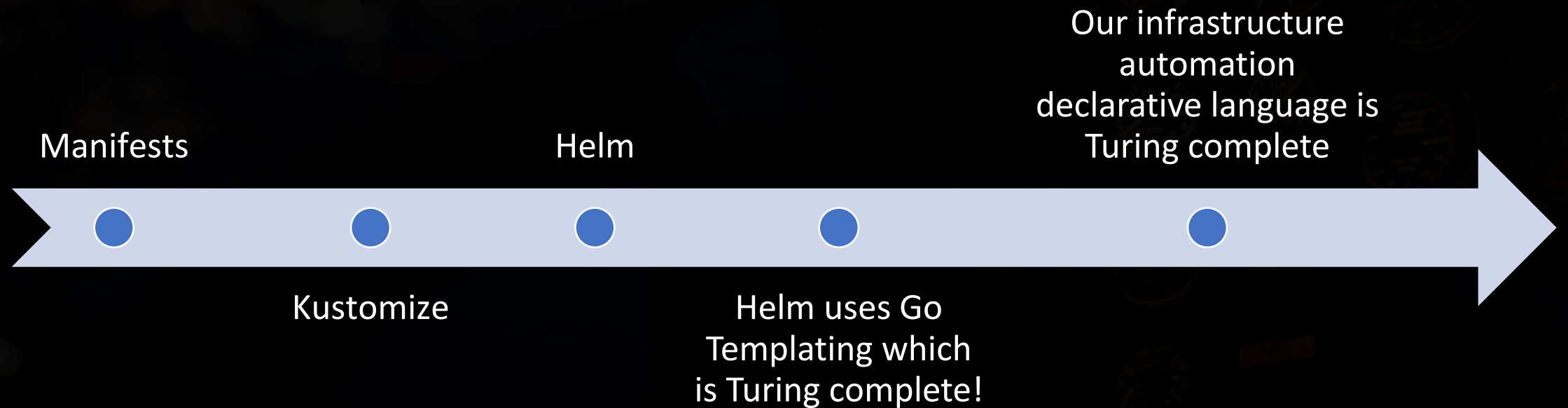
VMs managed as containers

Workloads need other stuff Kubernetes can do





Let's make it turing complete!



And so here we find ourselves



Swimming in an ocean of
complexity.



But all we wanted was some milk
with our cheerios.



I'll now take questions

