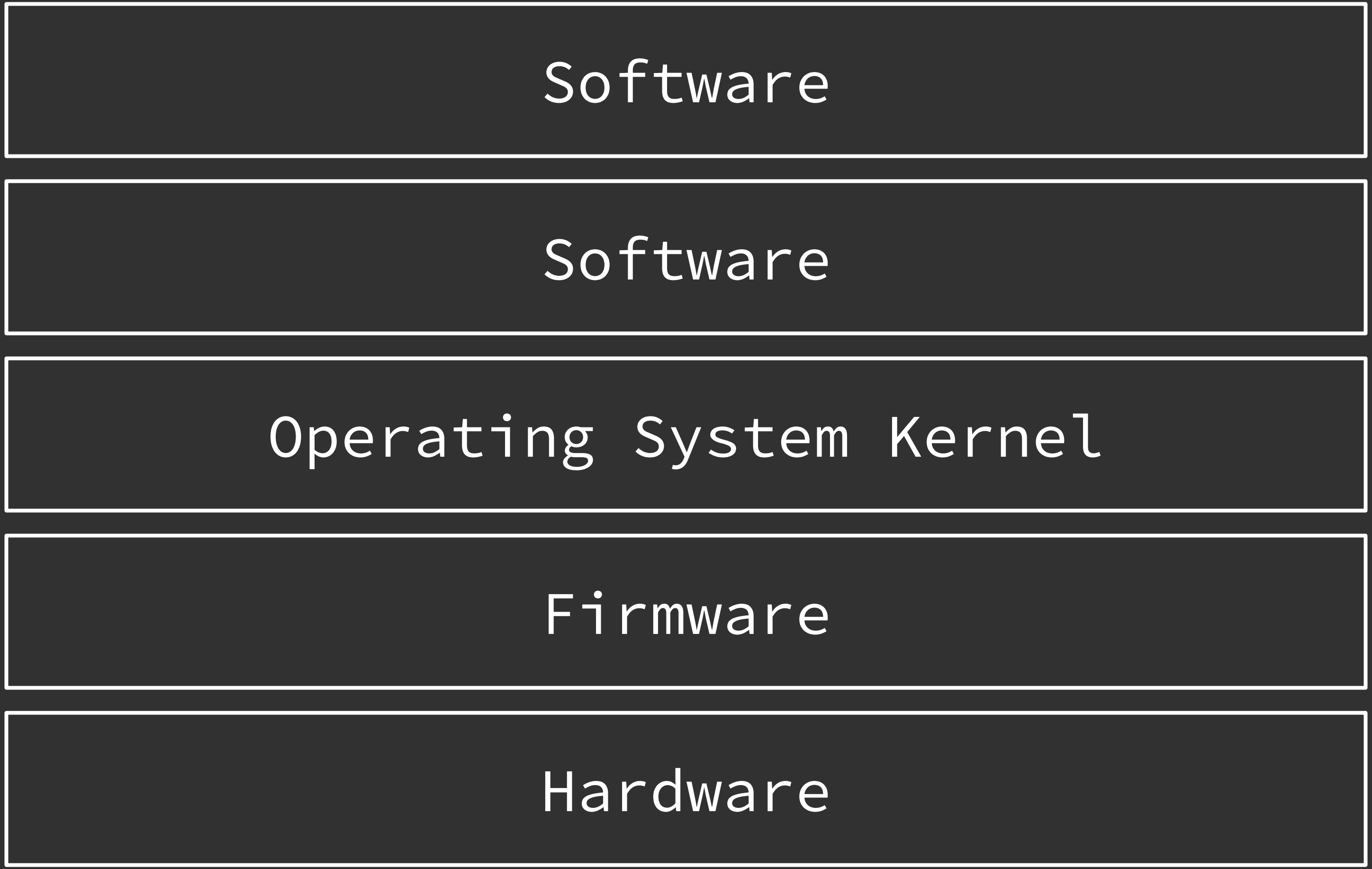
Why open source firmware is important

Jessie Frazelle – @jessfraz

Points of View

1. Security
2. Usability
3. Visibility

First Point of View: **Security...**



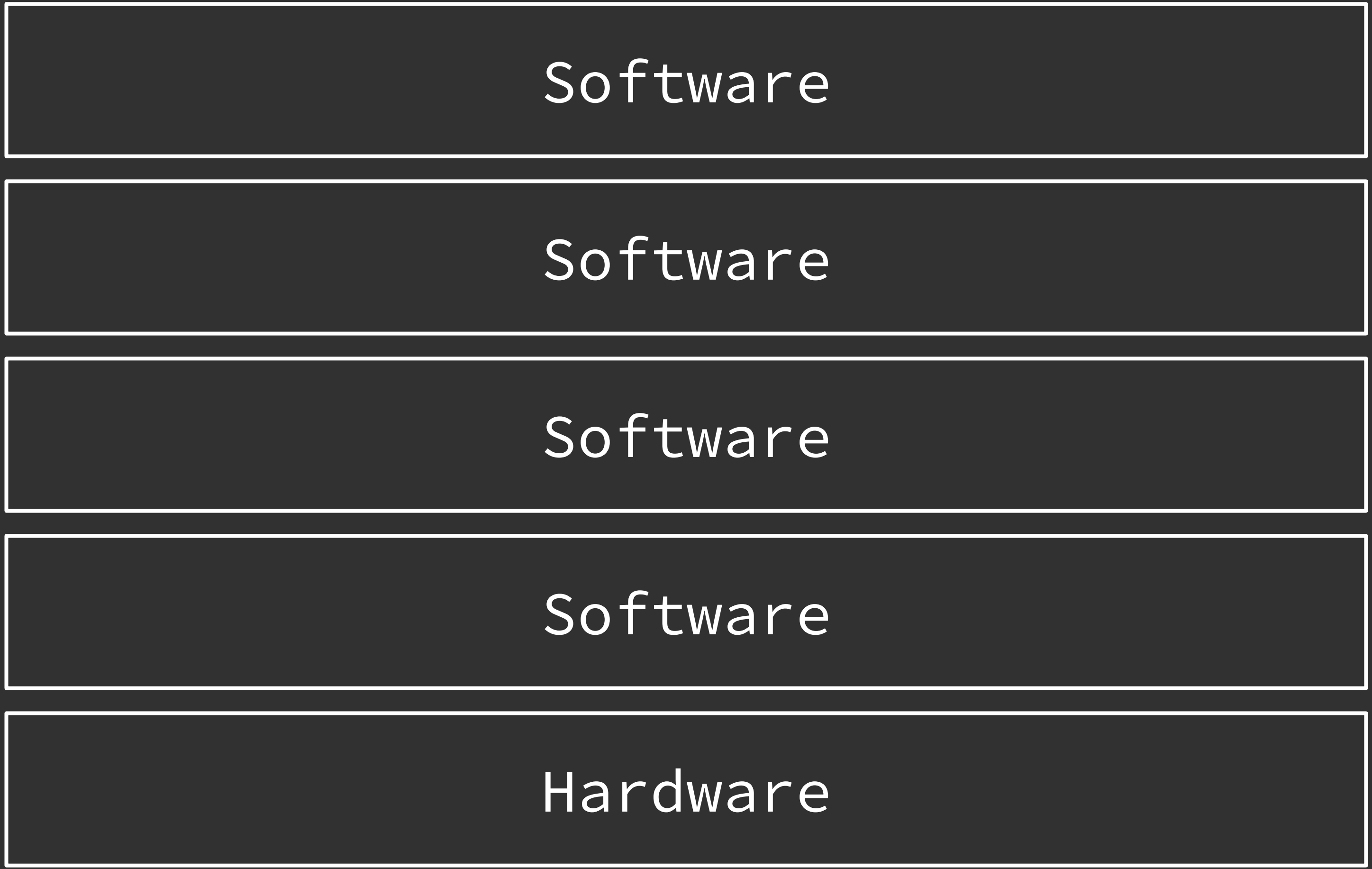
Software

Software

Operating System Kernel

Firmware

Hardware



Software

Software

Software

Software

Hardware

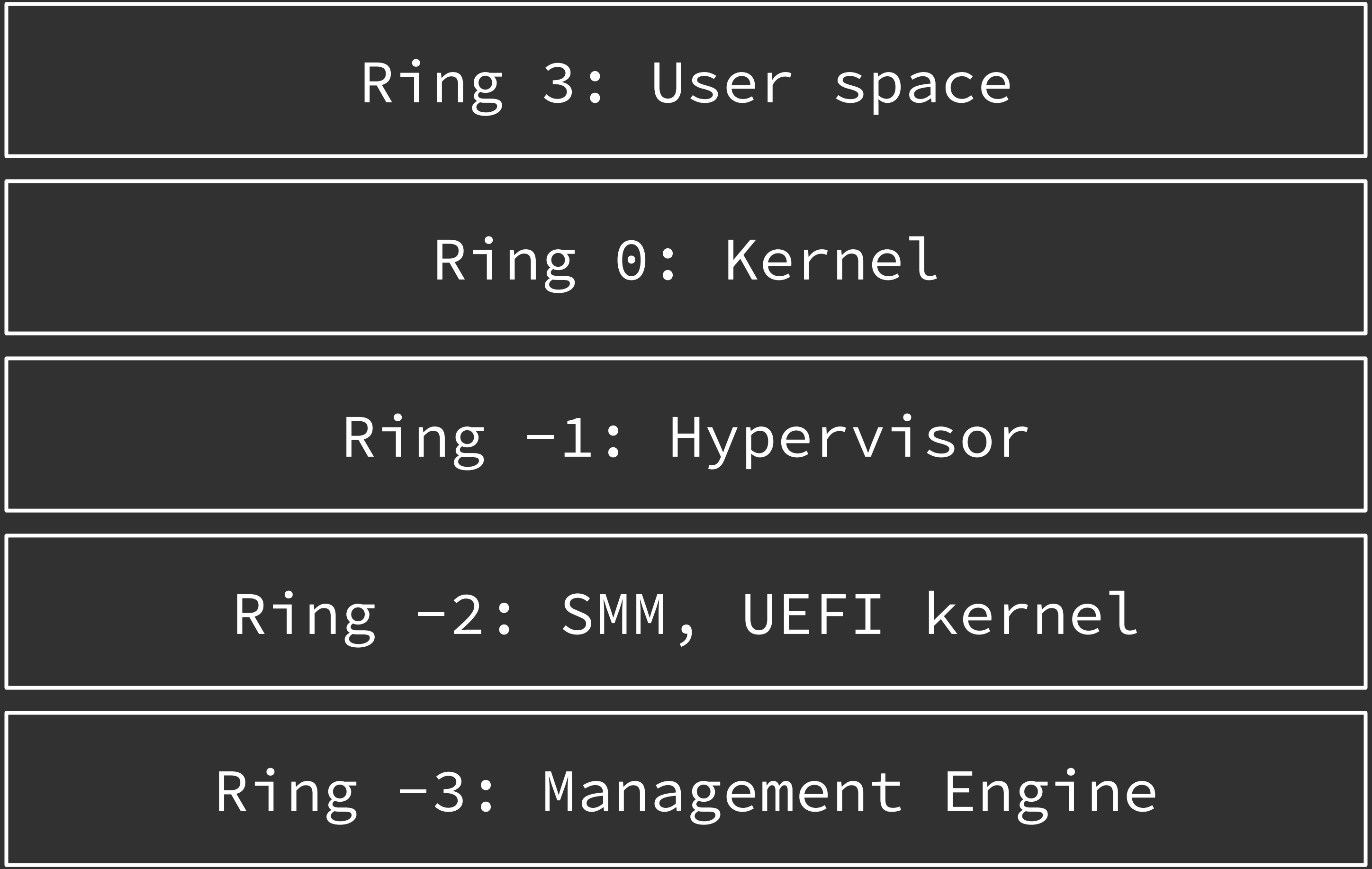
Soft💩are

Soft💩are

Soft💩are

Soft💩are

Hard💩are



Ring 3: User space

Ring 0: Kernel

Ring -1: Hypervisor

Ring -2: SMM, UEFI kernel

Ring -3: Management Engine

The code we don’t know about...

Ring -2: SMM, UEFI kernel

Ring -3: Management Engine

System Management Mode

- Originally used for power management
- System hardware control
- Proprietary designed code
- Place where vendors add new features
- Handle system events like memory or chipset errors
- ½ kernel

Ring -2: SMM, UEFI kernel

UEFI Kernel

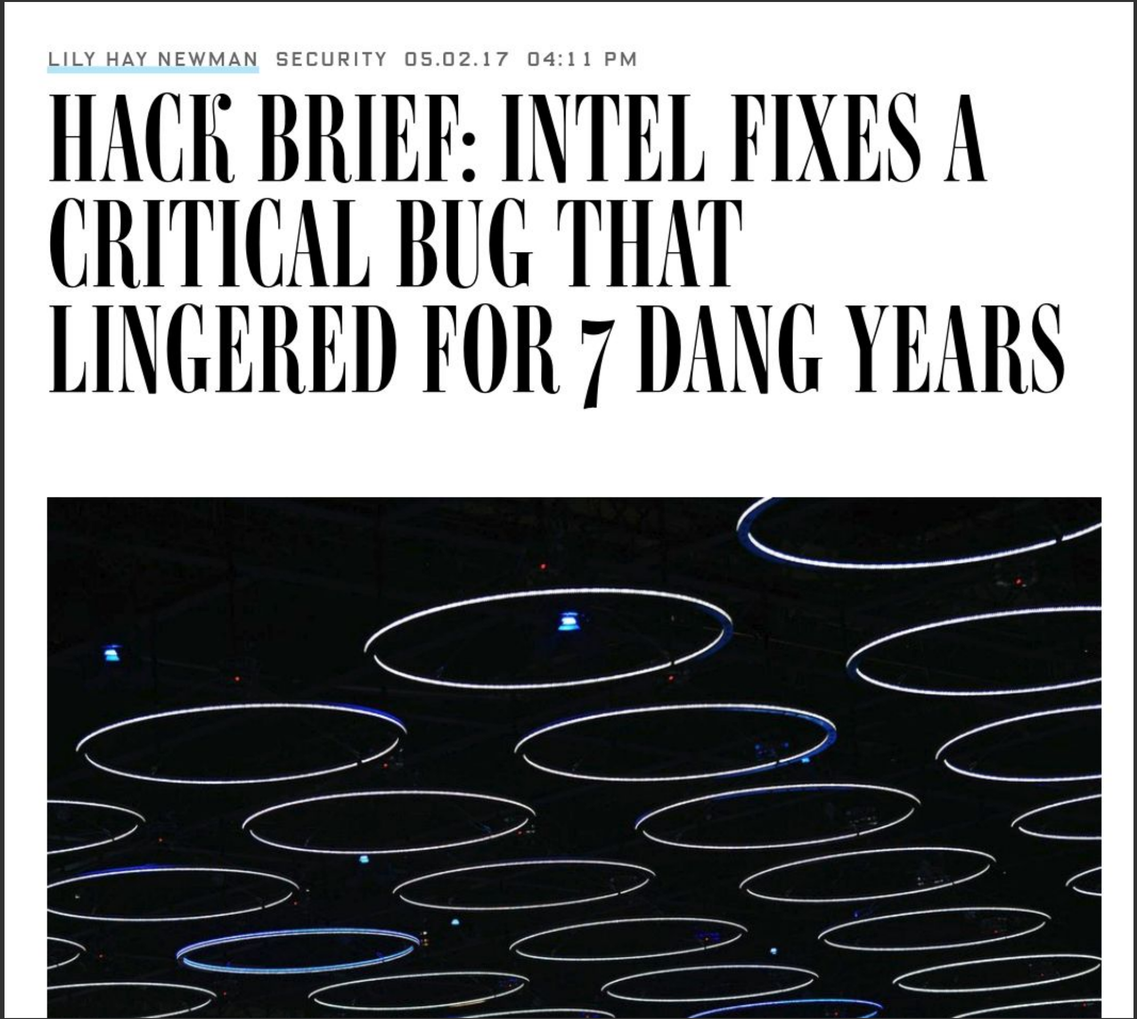
- Extremely complex
- Millions of lines of code
- UEFI applications are active after boot
- Security from obscurity
- A bajillion features, extremely complex

Ring -2: SMM, UEFI kernel

Management Engine

- Networking management
- KVM management
- Intel proprietary features
- Can reimage your device even if it’s powered off
- Can turn on node invisibly
- Minux
- SO MUCH MORE

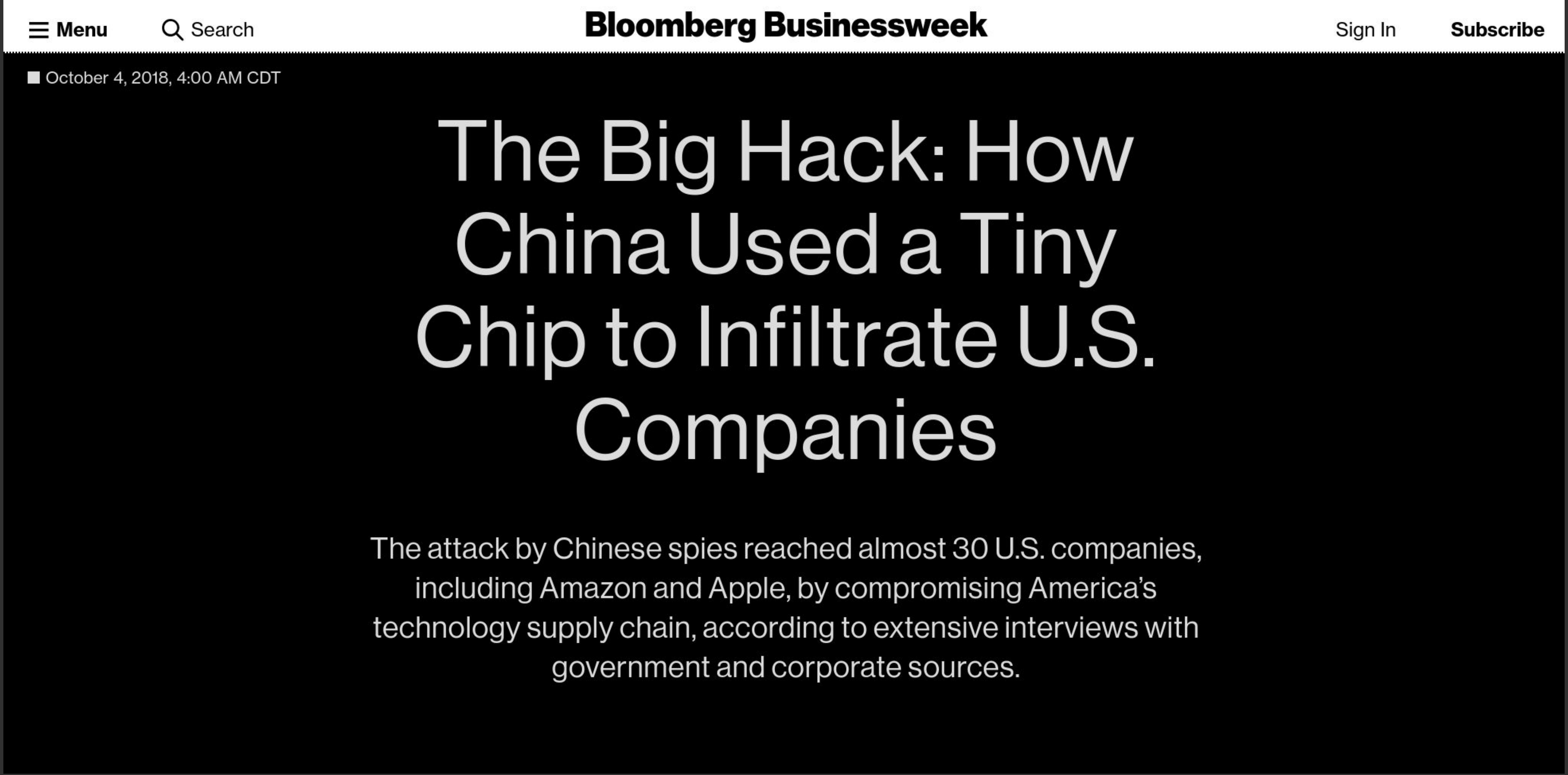
Ring -3: Management Engine



LILY HAY NEWMAN SECURITY 05.02.17 04:11 PM

HACK BRIEF: INTEL FIXES A CRITICAL BUG THAT LINGERED FOR 7 DANG YEARS

That’s just one example of a bad attack but if you google you can easily find others...



■ October 4, 2018, 4:00 AM CDT

The Big Hack: How China Used a Tiny Chip to Infiltrate U.S. Companies

The attack by Chinese spies reached almost 30 U.S. companies, including Amazon and Apple, by compromising America's technology supply chain, according to extensive interviews with government and corporate sources.

This is bad.

It gets even worse.

Intel Boot Guard

Adds up to: 2½ other kernels/OSes…

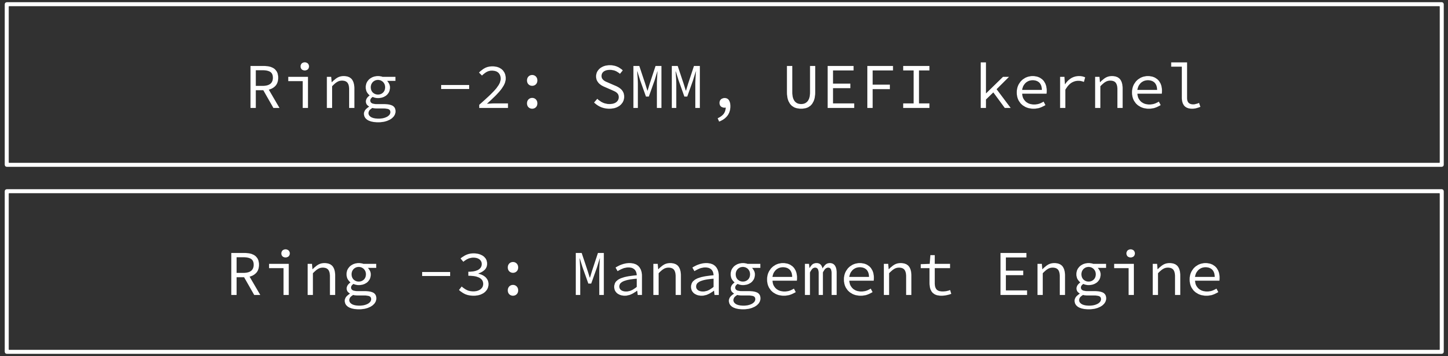
- They each have their own networking stacks, web servers (wtf)
- The code can modify itself and persist across power cycles and reinstalls

Ring -2: SMM, UEFI kernel

Ring -3: Management Engine

Adds up to: 2½ other kernels/OSes…

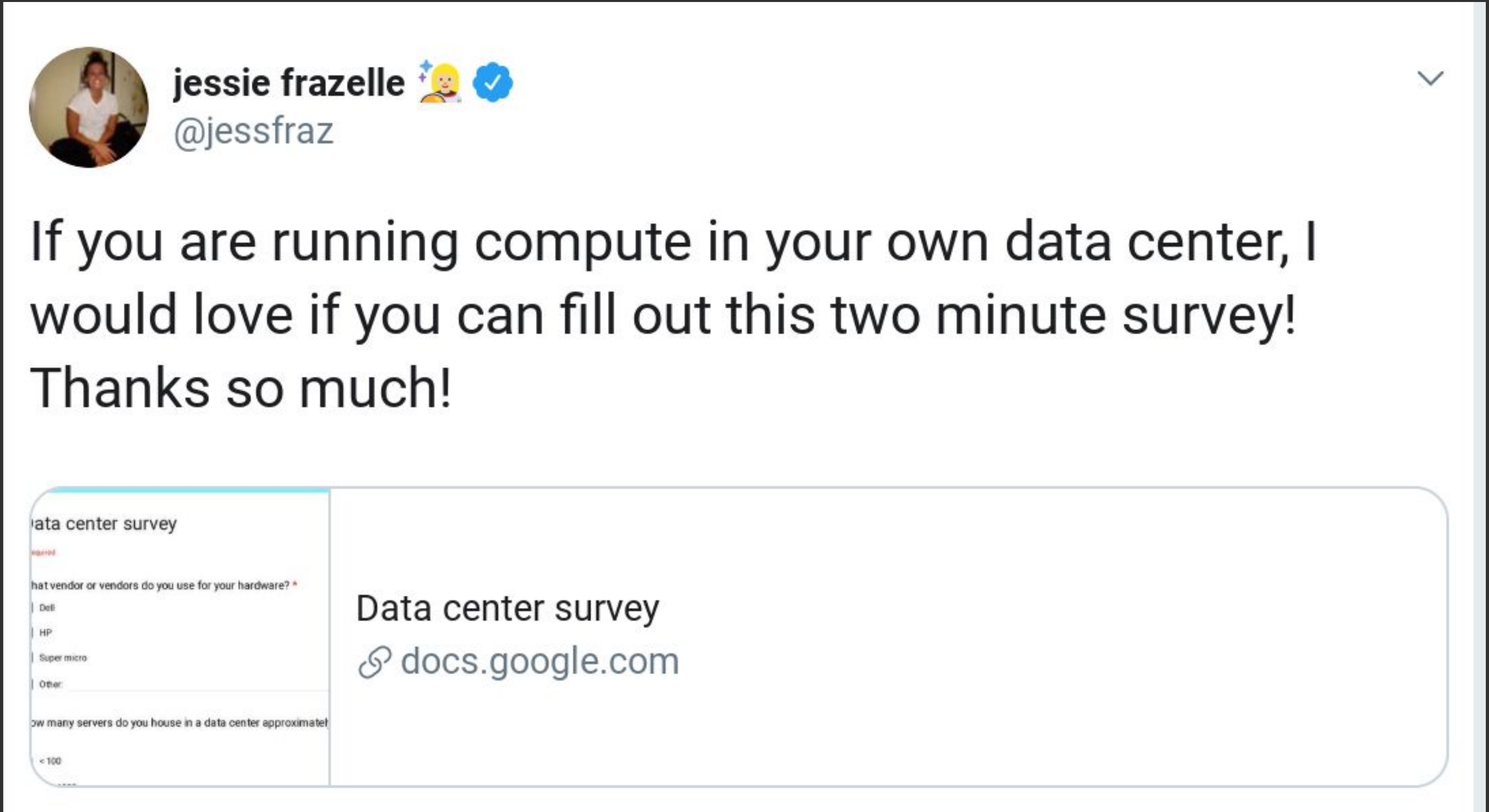
- They are all incredibly and unnecessarily complex
- **THEY ALL HAVE EXPLOITS!**



Ring -2: SMM, UEFI kernel

Ring -3: Management Engine

Second Point of View: **Usability...**



jessie frazelle ✨👩‍🚀 ✓
@jessfraz

If you are running compute in your own data center, I would love if you can fill out this two minute survey! Thanks so much!

ata center survey

equired

hat vendor or vendors do you use for your hardware? *

Dell

HP

Super micro

Other:

ow many servers do you house in a data center approximatel

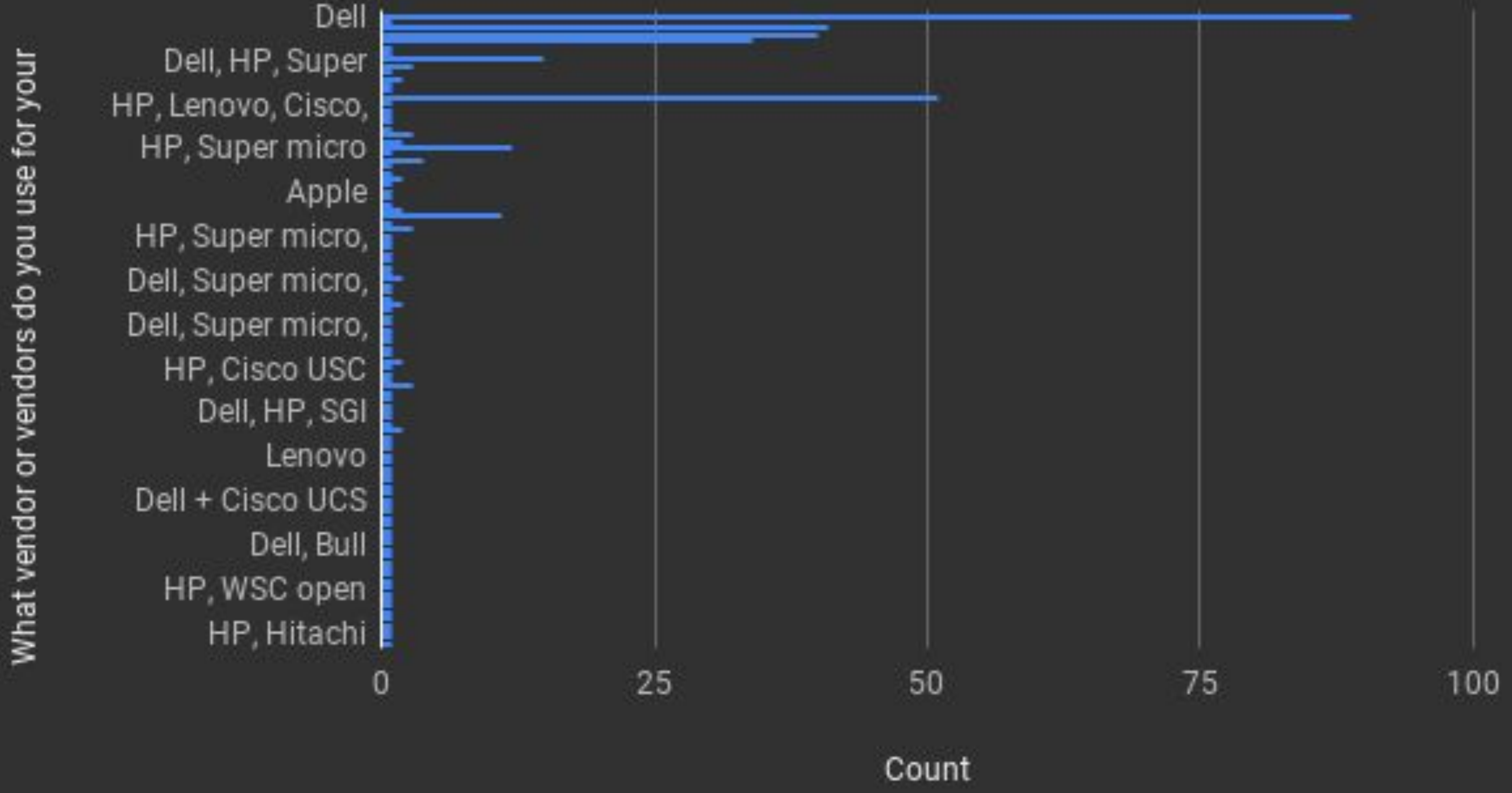
< 100

Data center survey

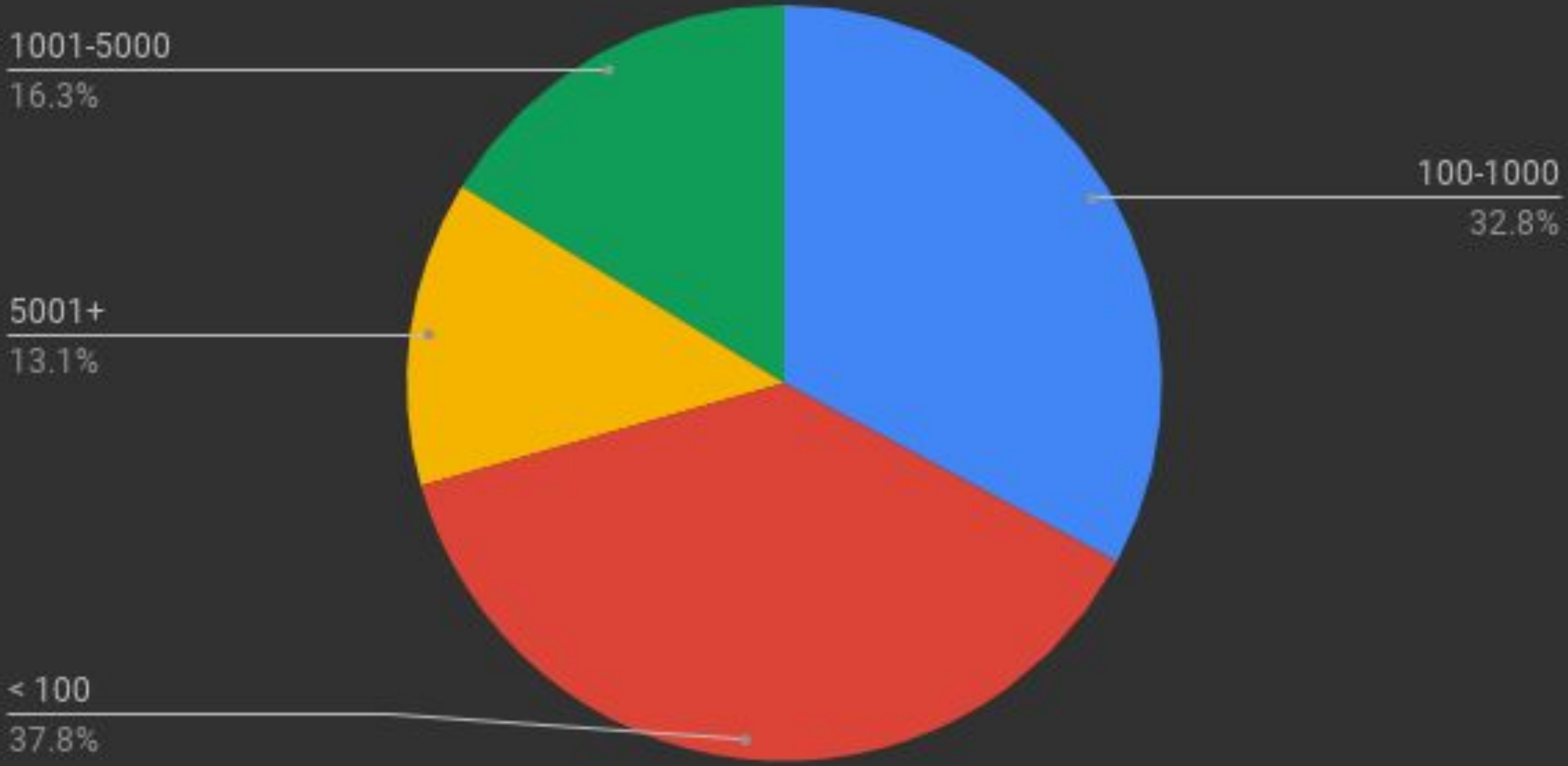
docs.google.com

The results

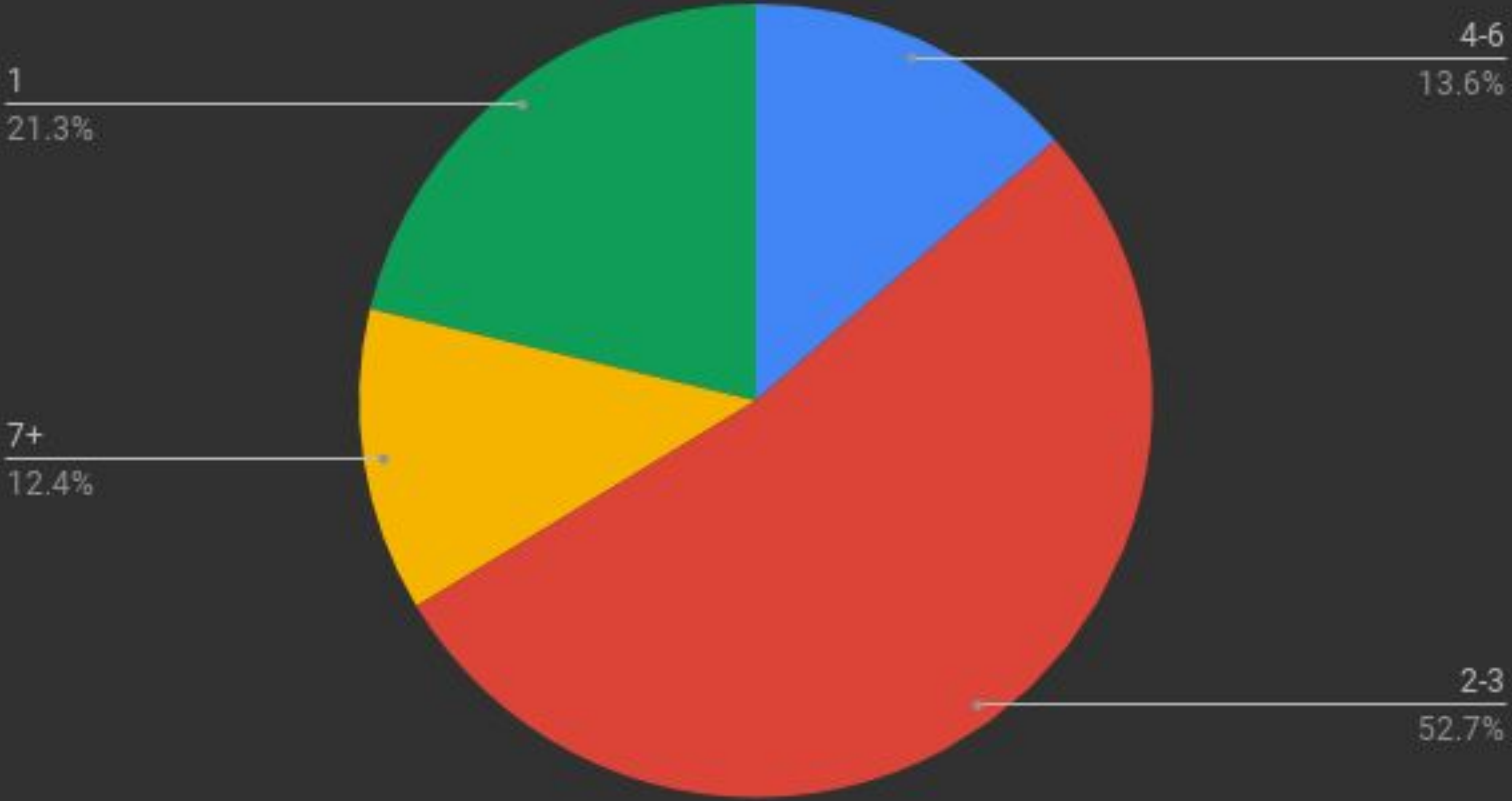
What vendor or vendors do you use for your hardware?



How many servers do you house in a data center approximately?



How many data centers do you have?

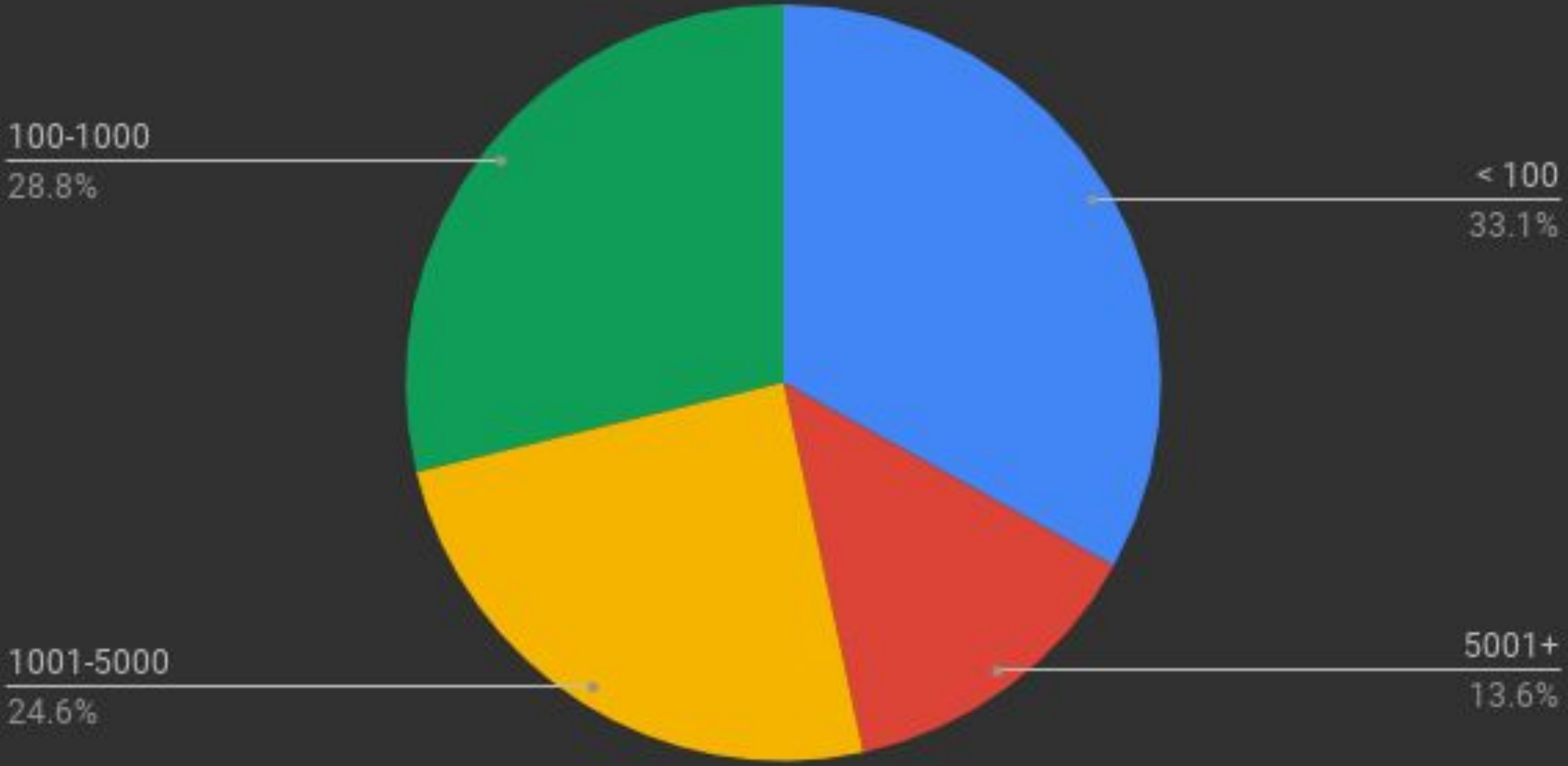


29.2%

Mentioned Firmware as a Pain Point

So at what scale is firmware a pain?

How many servers do you house in a data center approximately?



My hypothesis...

Once you need to deal with the firmware it becomes a pain...

Third Point of View: **Visibility...**

Conway’s Law

From the perspective of hardware engineers...

“You’d be crazy to think hardware was ever intended to be used for isolating multiple users safely..”

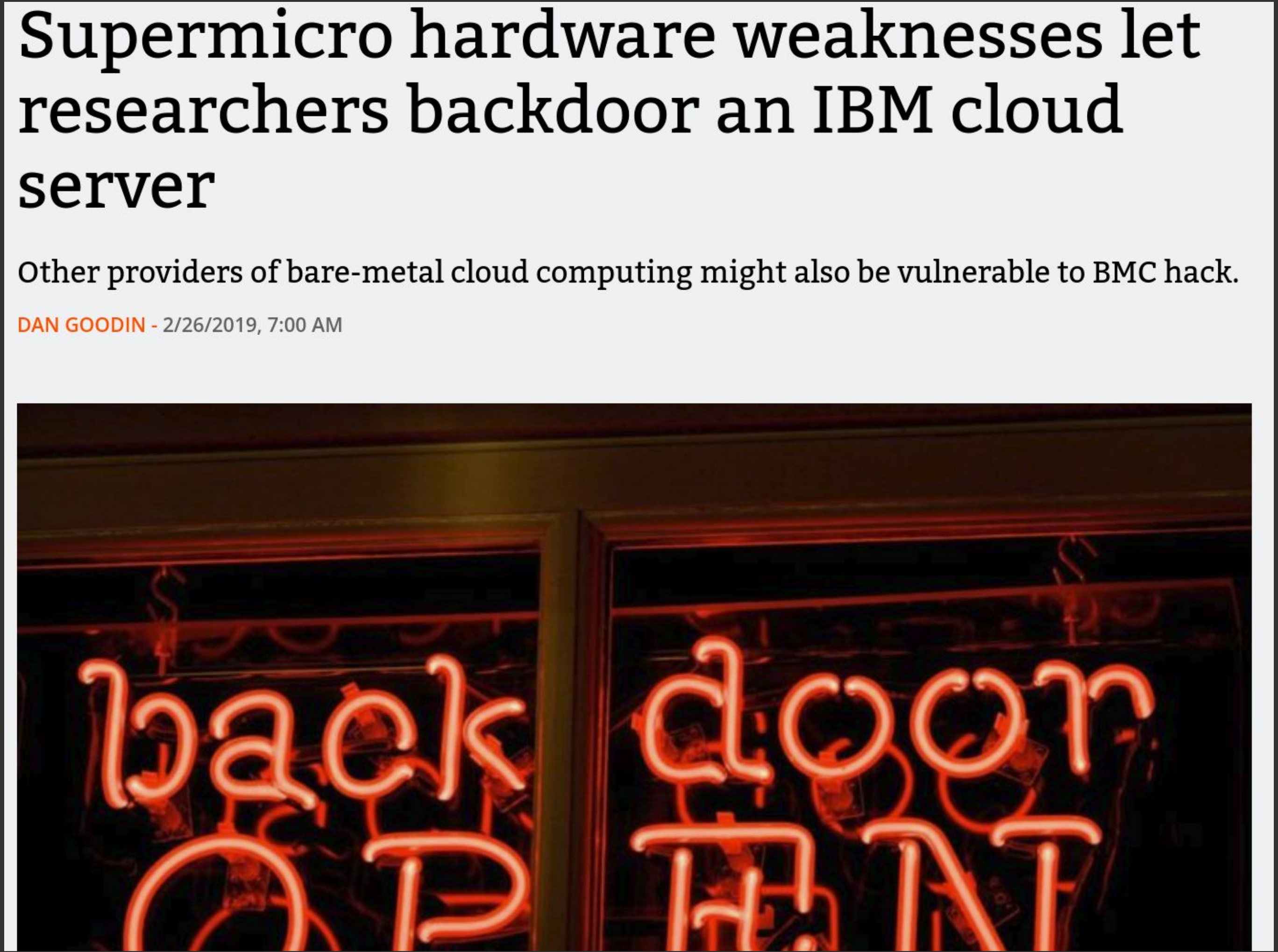
Spectre and Meltdown proved this to be true as well.

From the perspective of firmware and kernel engineers…

They want vendors to make their firmware do less, or give up the control to them.

Vendors can rarely debug firmware issues…

Oversights and
lack of
communication
leads to...



Supermicro hardware weaknesses let researchers backdoor an IBM cloud server

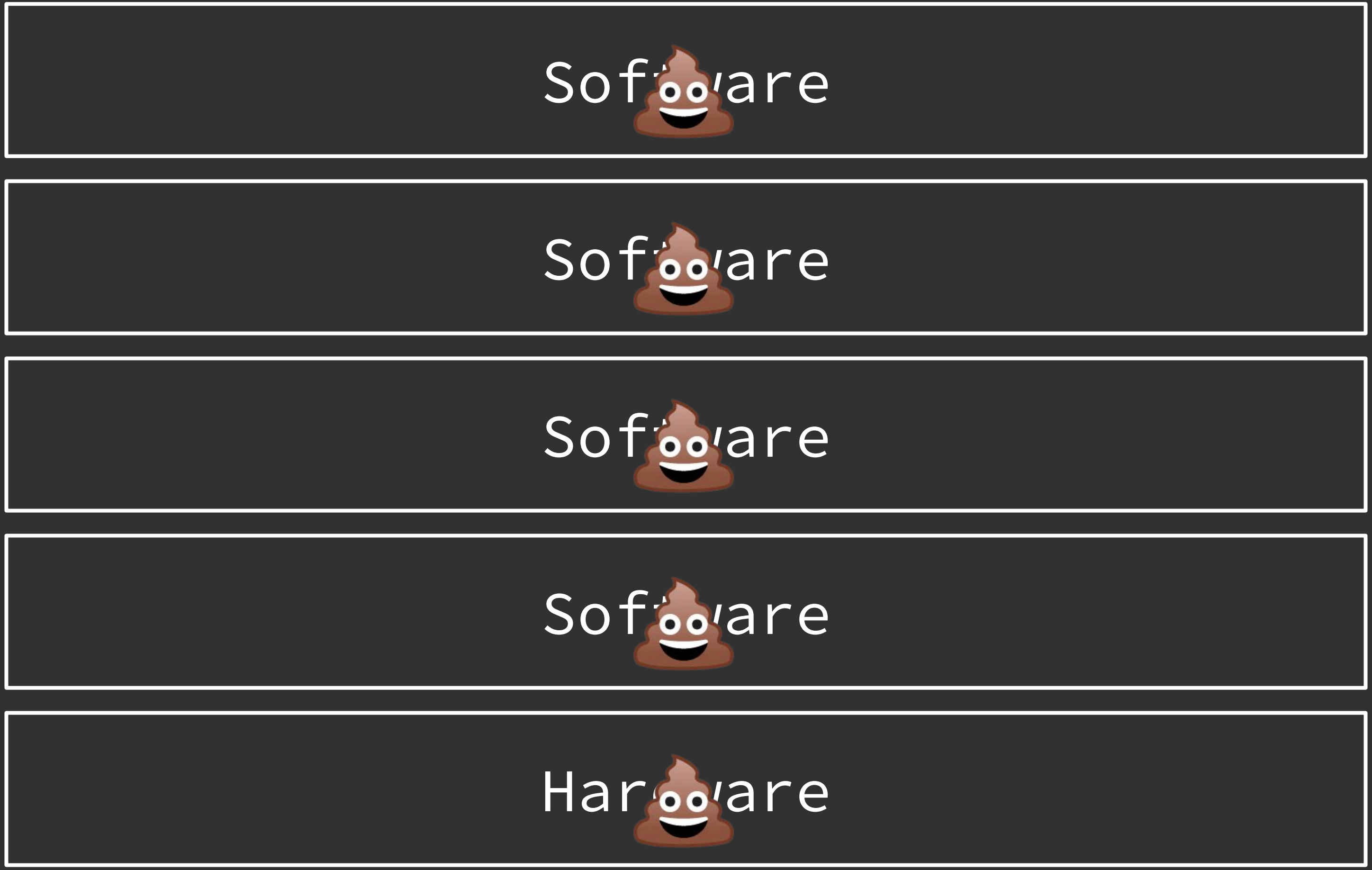
Other providers of bare-metal cloud computing might also be vulnerable to BMC hack.

DAN GOODIN - 2/26/2019, 7:00 AM

How did no one think about the BMC when building softlayer?

I’ve personally seen these miscommunications happen in the container ecosystem as well...

Miscommunications at various layers of the stack lead to bugs in the intersecting layers, based off incorrect assumptions.



Soft💩are

Soft💩are

Soft💩are

Soft💩are

Hard💩are

How do we fix these things?

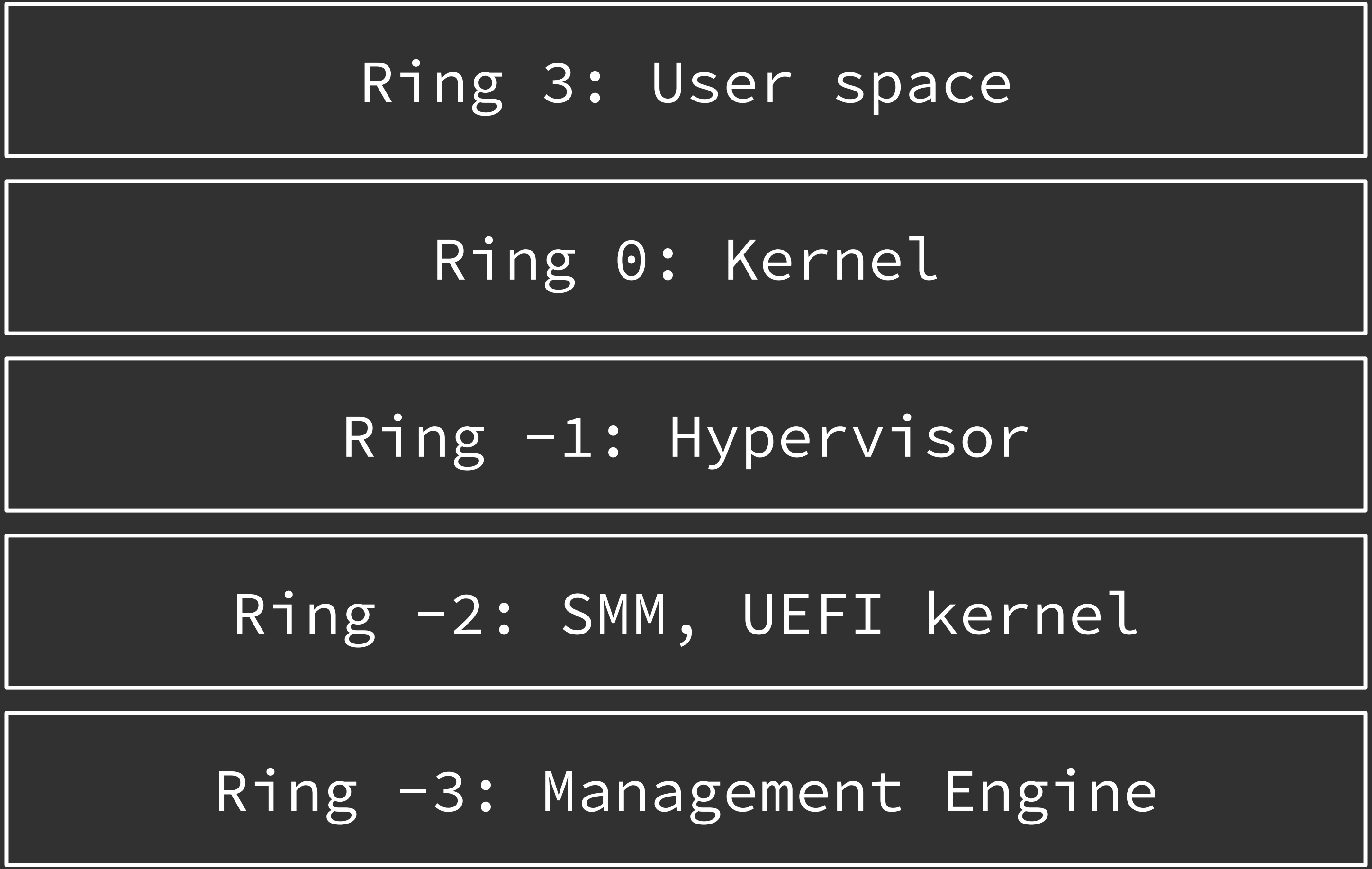
1. Security
2. Usability
3. Visibility

Open Source Firmware

NERF:
Non-Extensible
Reduced Firmware

NERF Goals

- Make firmware less capable of doing harm
- Make its actions more visible
- Remove all runtime components
  - With ME we can’t remove all but we can take away the web server and IP stack
- Remove UEFI IP stack and other drivers
- Remove ME/UEFI self-reflash capability
- Let linux manage flash updates



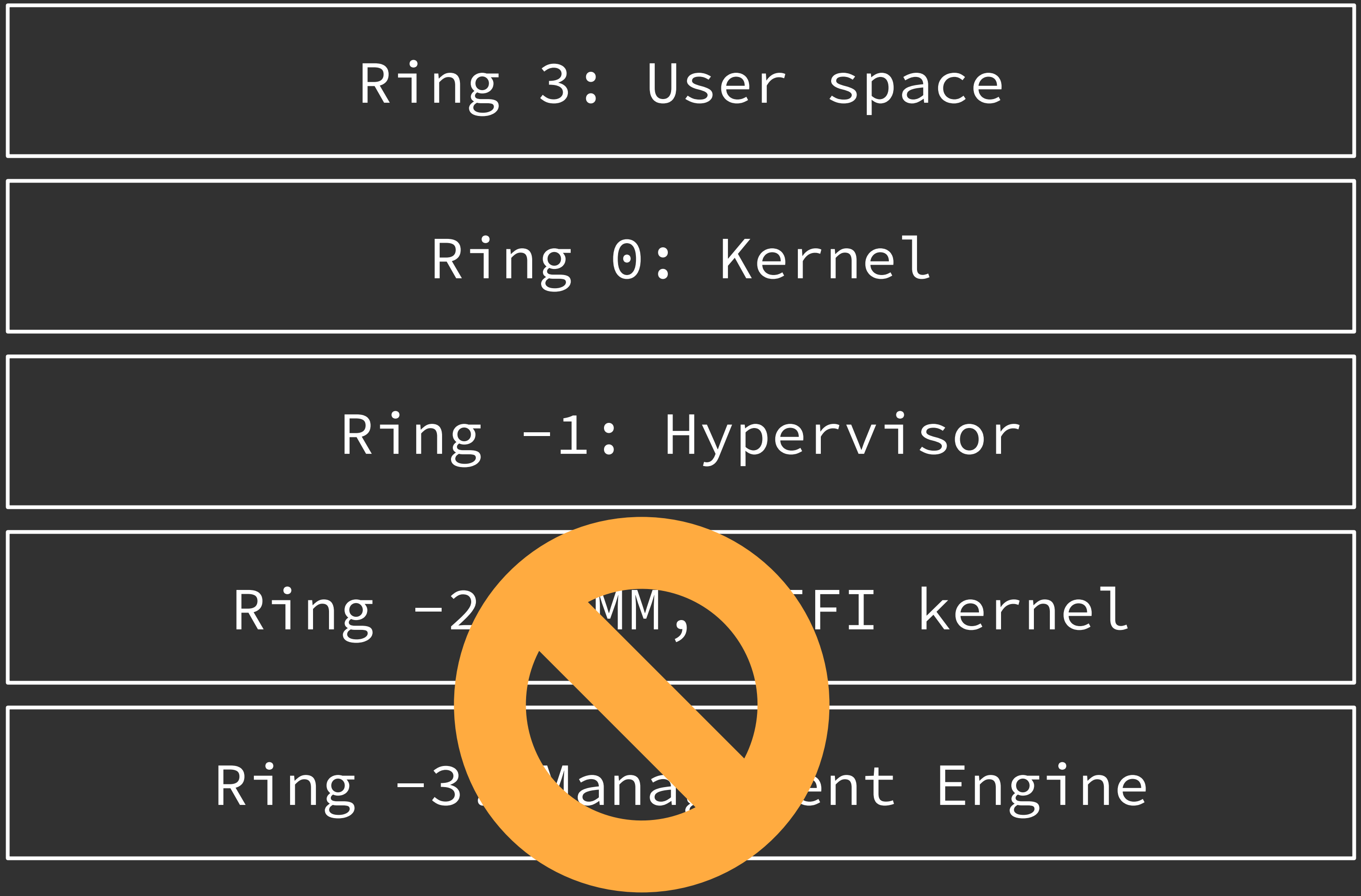
Ring 3: User space

Ring 0: Kernel

Ring -1: Hypervisor

Ring -2: SMM, UEFI kernel

Ring -3: Management Engine



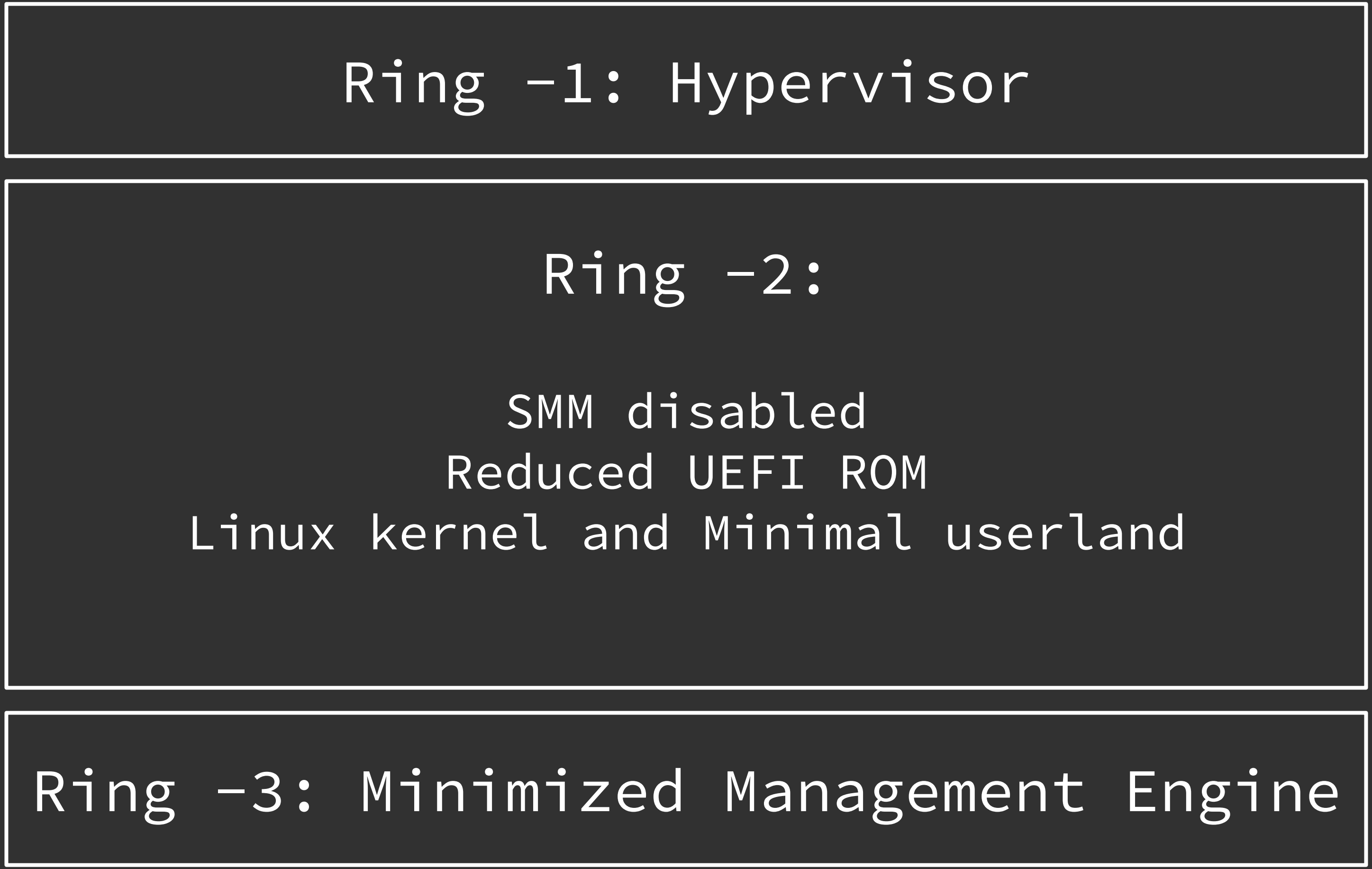
Ring 3: User space

Ring 0: Kernel

Ring -1: Hypervisor

Ring -2: SMM, UEFI kernel

Ring -3: Management Engine

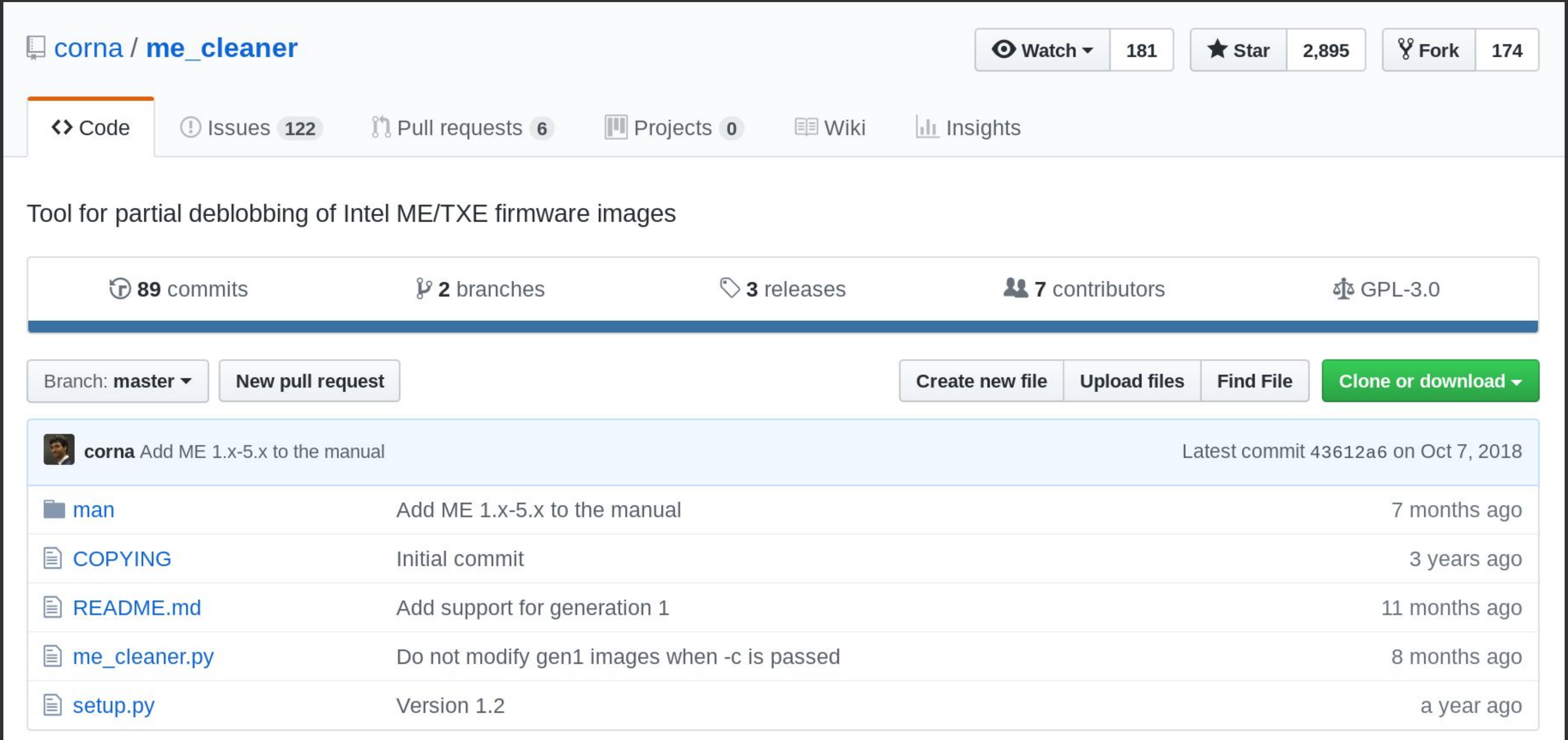


Ring -1: Hypervisor

Ring -2:

SMM disabled
Reduced UEFI ROM
Linux kernel and Minimal userland

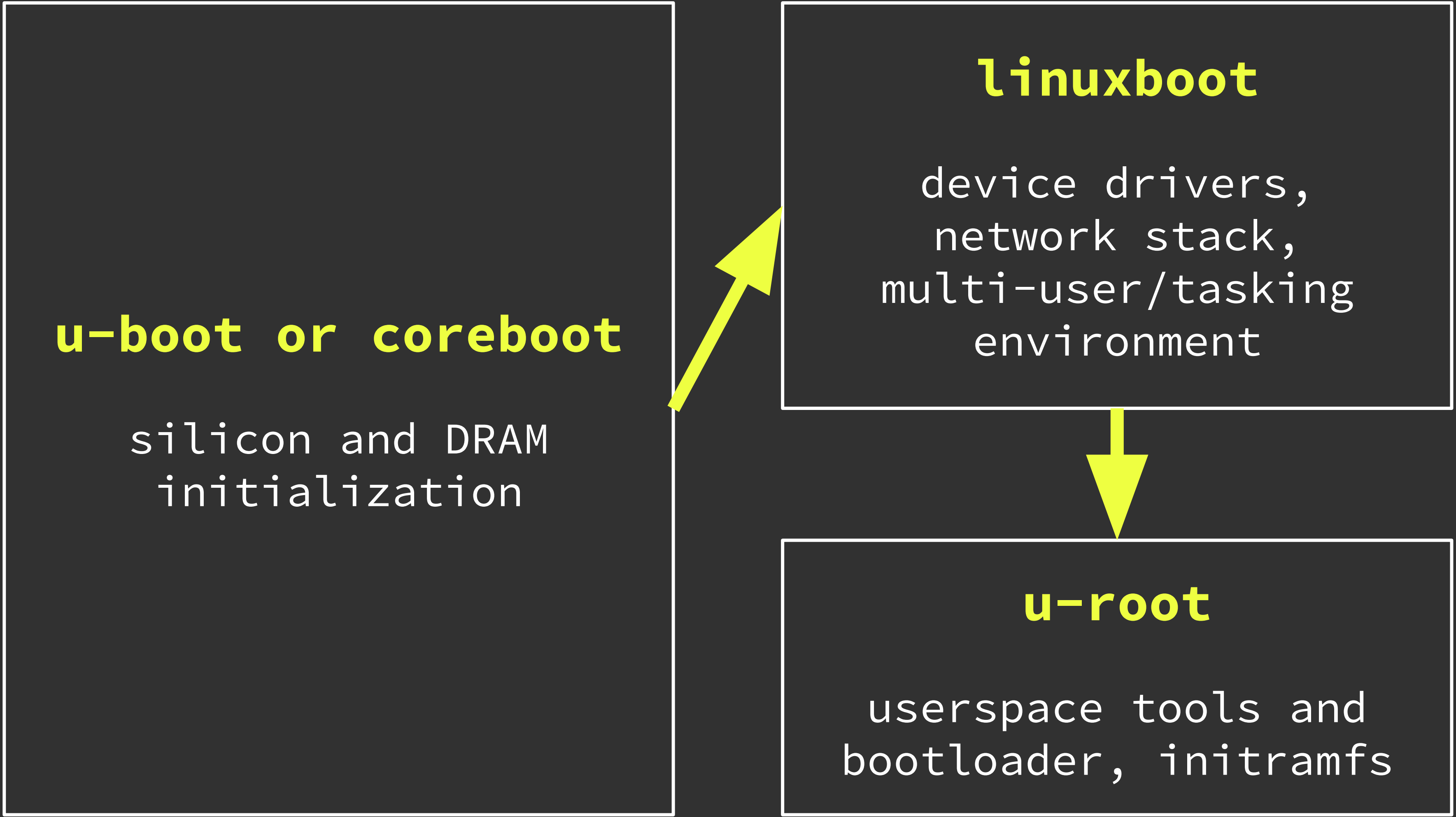
Ring -3: Minimized Management Engine



Tool for partial deblobbing of Intel ME/TXE firmware images

corna Add ME 1.x-5.x to the manual — Latest commit `43612a6` on Oct 7, 2018

man	Add ME 1.x-5.x to the manual	7 months ago
COPYING	Initial commit	3 years ago
README.md	Add support for generation 1	11 months ago
me_cleaner.py	Do not modify gen1 images when -c is passed	8 months ago
setup.py	Version 1.2	a year ago



u-boot or coreboot

silicon and DRAM initialization

linuxboot

device drivers, network stack, multi-user/tasking environment

u-root

userspace tools and bootloader, initramfs

Why linux?

- Single kernel works for several boards
- Already quite vetted and has a lot of eyes on it since it is used quite extensively
- Single, open source kernel versus the 2½ other kernels that were all different and most closed off
- Improves boot reliability by replacing lightly-tested firmware drivers with hardened Linux drivers.

Other wins

- Firmware devs can build in tools they already know
- When they need to write logic for signature verification, disk decryption, etc it’s in a language that is modern, easily auditable, maintainable, and readable
- Memory safety wins as well since the language can be higher level

Makes boot time
20x faster.

Through open source, visibility, minimalism, and open communication we can push computing to a better, more secure place from the hardware up.

We can’t keep building on top of 💩. We really need to care about the base we build on.

Huge thanks to the firmware community for all their work on this!

Ron Minnich

Trammel Hudson

Chris Koch

Rick Altherr

Zaolin

Thanks for having me!