



Making the internet hop faster.

A Reasonably Secure Workstation

Simon Newton, VP @ bunny.net



Agenda

1. **Intro**
2. **Requirements**
3. **Heads Firmware**
4. **Intel Management Engine**
5. **First Heads Intel Desktop with PCIe slots**
6. **Physical Security**
7. **The Qubes OS desktop**
8. **Get Involved & bunny.net Internship**




Intro

Simon Newton:

- VP at bunny.net, previously with Akamai for over a decade.
- 30 years working in tech, writing code since age 5
- Keen interest in opsec, persec, privacy and freedom
- Love hardware reverse engineering & RF stuff
- QubesOS community team, Heads contributor
- *Many* boxes of wires and peripherals

bunny.net

- EU based, privacy respecting, GDPR compliant CDN, Storage and Cloud service provider
- Ranked #1 fastest CDN in the world (cdnperf.com)
- Powering over a million sites
- 81+ Tbps capacity network
- 114 POPs globally



WORLD				CONTINENTS	COUNTRIES	PERF
CDN NAME	QUERY SPEED	0	20			
1 bunny.net	26.74 ms					
2 AWS CloudFront CDN	26.97 ms					
3 CacheFly	27.45 ms					
4 Edgio	27.45 ms					
5 Cloudflare CDN	28.19 ms					
6 Edgecast CDN	29.68 ms					
7 Fastly CDN	29.97 ms					
8 CDNetworks	30.52 ms					
9 Google Cloud CDN	31.58 ms					
10 CDN77	32.26 ms					
11 Akamai CDN	34.32 ms					

Making a hardened workstation

Requirements

1. We want to be able to audit and verify from the first instruction in CPU
2. We want to use entirely open source code – no closed vendor blobs
3. We want to be sure that our firmware integrity is maintained
4. We want to be sure our operating system integrity is maintained
5. We want to ensure our day to day use of our workstation enhances privacy while blocking adversaries – without impeding our ability to operate effectively
6. We want to be able to run multiple operating systems seamlessly
7. We want our system to be efficient and useful for our research too



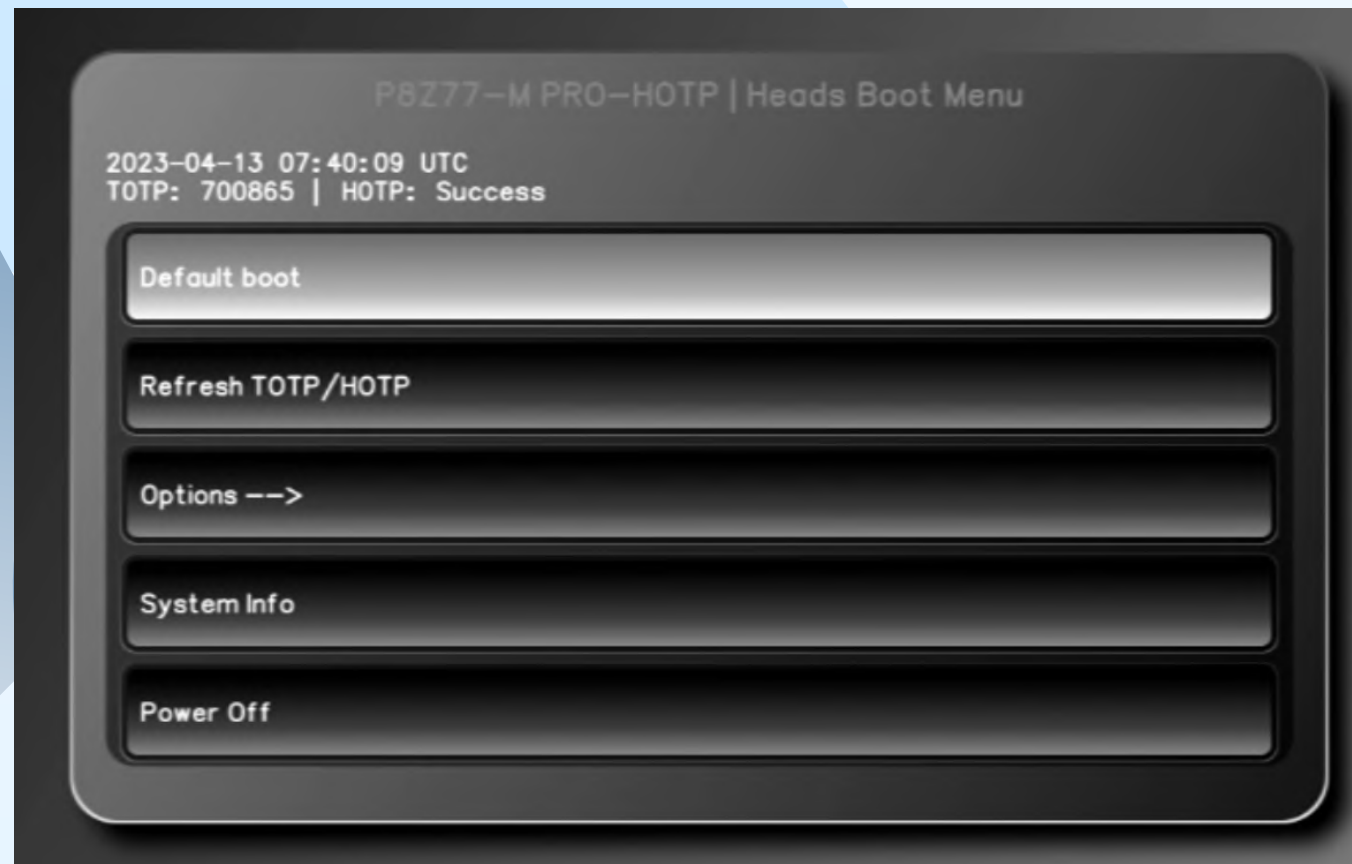


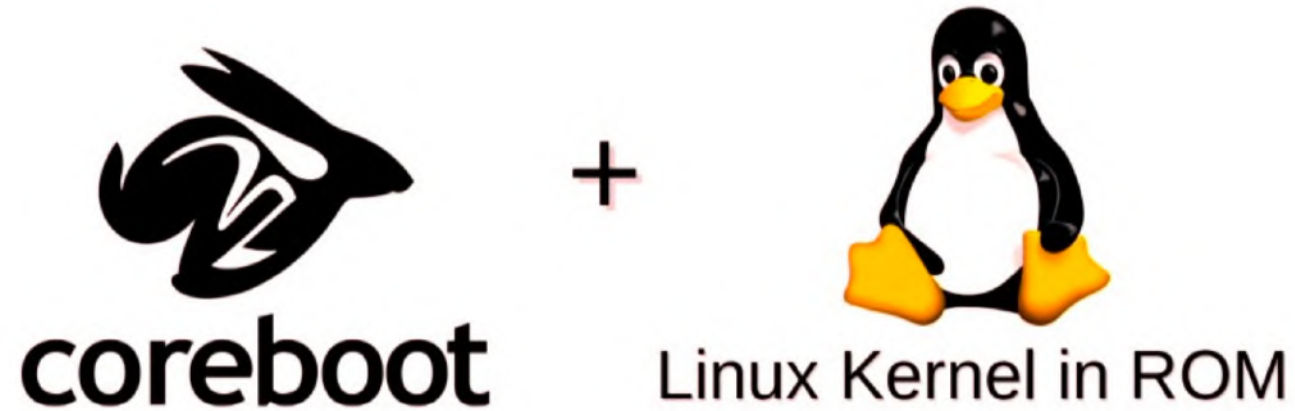
Heads Firmware

How to cover requirements 1 through 4



- Replaces vendor 'black box' firmware
- Built on Coreboot and Linux
- Used on Chromebooks, Puri.sm, Insurgo & Nitrokey laptops
- Open source, transparent, auditable
- Reproducible builds
- Handles all hardware init and PCRs
- Attests to its own integrity at boot time
- Uses NitroKey or Librem key for HOTP
- Uses Authenticator app for TOTP
- Prevents "Evil Maid" attacks
- Prevents firmware rootkits & backdoors





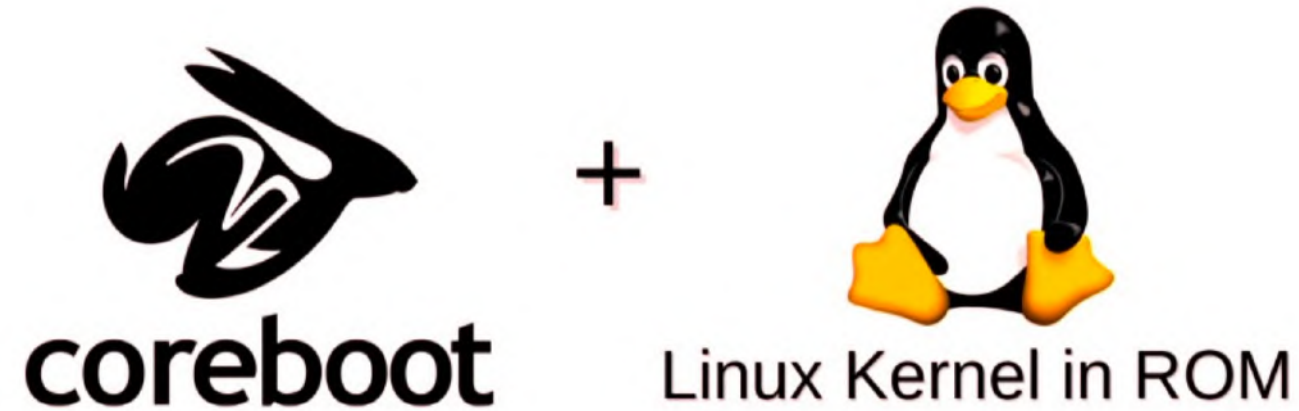
Heads Firmware

Requirements 1 & 2

- Uses Coreboot (hardware init) & Linux (coreboot payload) to replace vendor BIOS/UEFI
- Moves the root of trust to ROM, prevents modifications to the bootup code
- Runs the first instruction on CPU, measuring firmware, config & system state into the TPM
- Leverages TPM Platform Configuration Registers (PCRs)

Vendor Blob situation

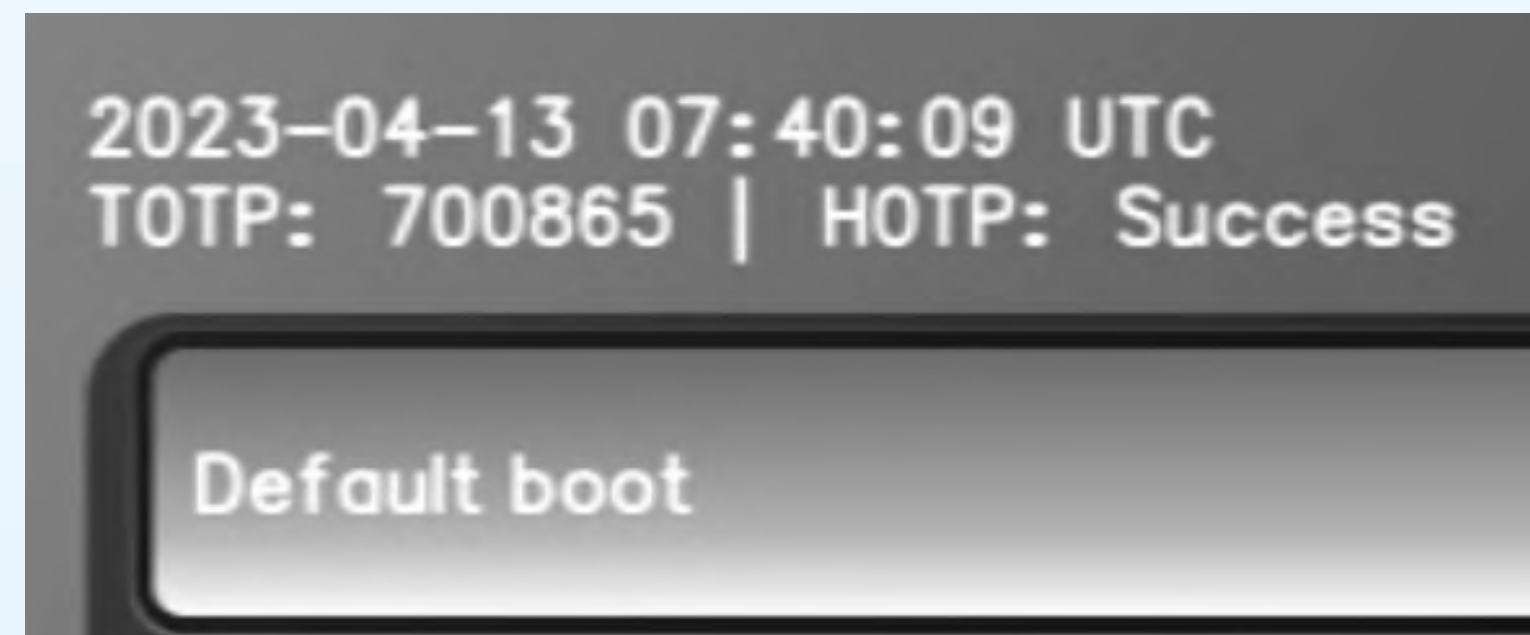
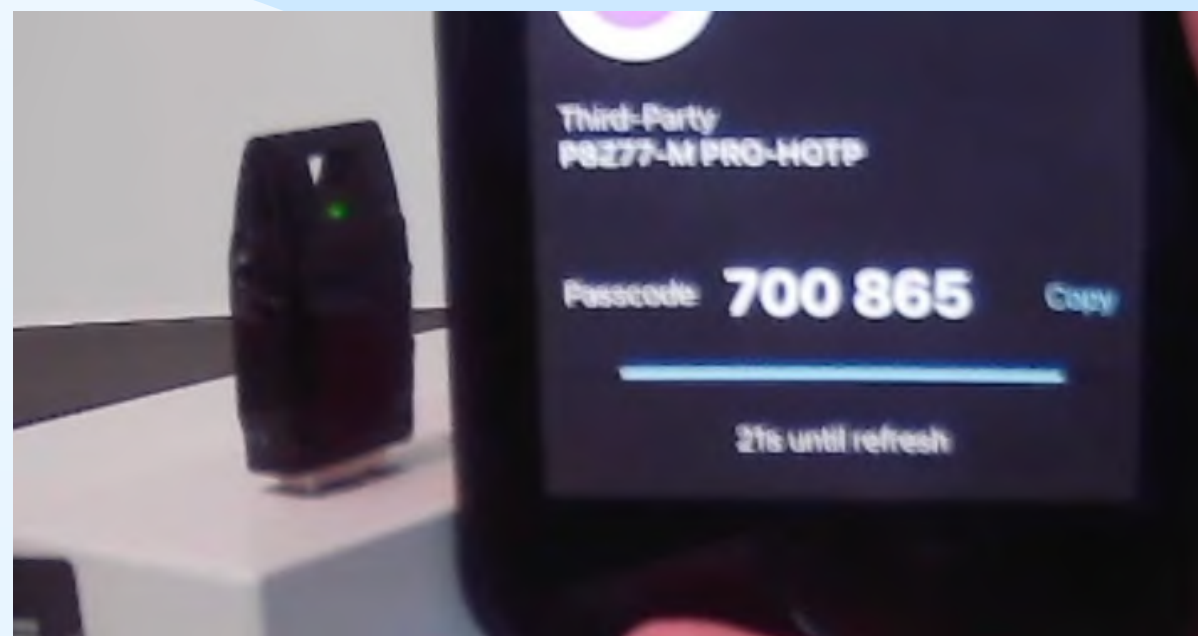
- On x86 Intel Ivy/Sandy Bridge:
 - All native hardware init, no blobs required
 - Neuter, soft disable and shrink Intel Management Engine blob
- On x86: AMD – Fam15h: All native hardware init, no blobs required
- Power9 open architecture – Talos II: All native hardware init, no blobs required
- More info: <https://github.com/osresearch/heads/issues/692>



Heads Firmware

Requirement 3 – Firmware Integrity

- Attestation that nothing has changed our firmware (backdoor, rootkit persistence, etc)
- Uses TPM PCR values to measure the boot process and attest firmware integrity
- Uses HOTP to give a visual green/red firmware attestation with an external token
- Uses TOTP to validate with your favorite TOTP authenticator
- Any change halts boot and reports a failure





Heads Firmware

Requirement 4 – OS integrity

- GPG token device is used to sign the /boot/ hashes of the system with private GPG key
- The GPG public key is stored in the heads, attested firmware
- When booting the linux kernel (kexec) heads uses embedded GPG PubKey to perform verification of /boot integrity:
 - Kernel hashes
 - Initrd hashes
 - System map hashes
 - Grub config hashes
- Any change halts boot and reports failure

```
+++ Found verified kexec boot params
gpg: Signature made Thu Apr 13 07:32:28 2023 UTC
gpg:                using RSA key 5CA7BC55A8A01B623D75BE0203E96BFF
gpg: Good signature from "OEM Key (OEM-generated key) <oen-2023041
+++ Found verified kexec boot params
New value of PCR[4]: 11c4ecaf31383e76686cc64f0c1dd88bd918ea3d
+++ Scanning for unsigned boot options
+++ Checking verified boot hash file
+++ Verified boot hashes
170917203: 000e4522135100000062
/tnp/counter-170917203: OK
+++ Checking verified default boot hash file
+++ Verified default boot hashes
+++ Executing default boot for Qubes, with Xen hypervisor:
Loading the new kernel:
kexec -l /boot/xen-4.14.5.gz --command-line "placeholder console=t
z-5.15.94-1.qubes.fc32.x86_64 placeholder root=/dev/mapper/qubes d
```


Heads Firmware

Drawbacks to heads

- Take a little longer to boot on some boards while measuring
- Limited vendor and board support (help porting always welcome!)
- Mostly older generation CPUs (though some work ongoing for newer systems)
- Some supported machines have complex disassembly & programming requirements to install
- My recent P8Z77-M Pro port makes installing so much easier thanks to a DIP8 removable flash rom

<https://github.com/osresearch/heads/pull/1358>

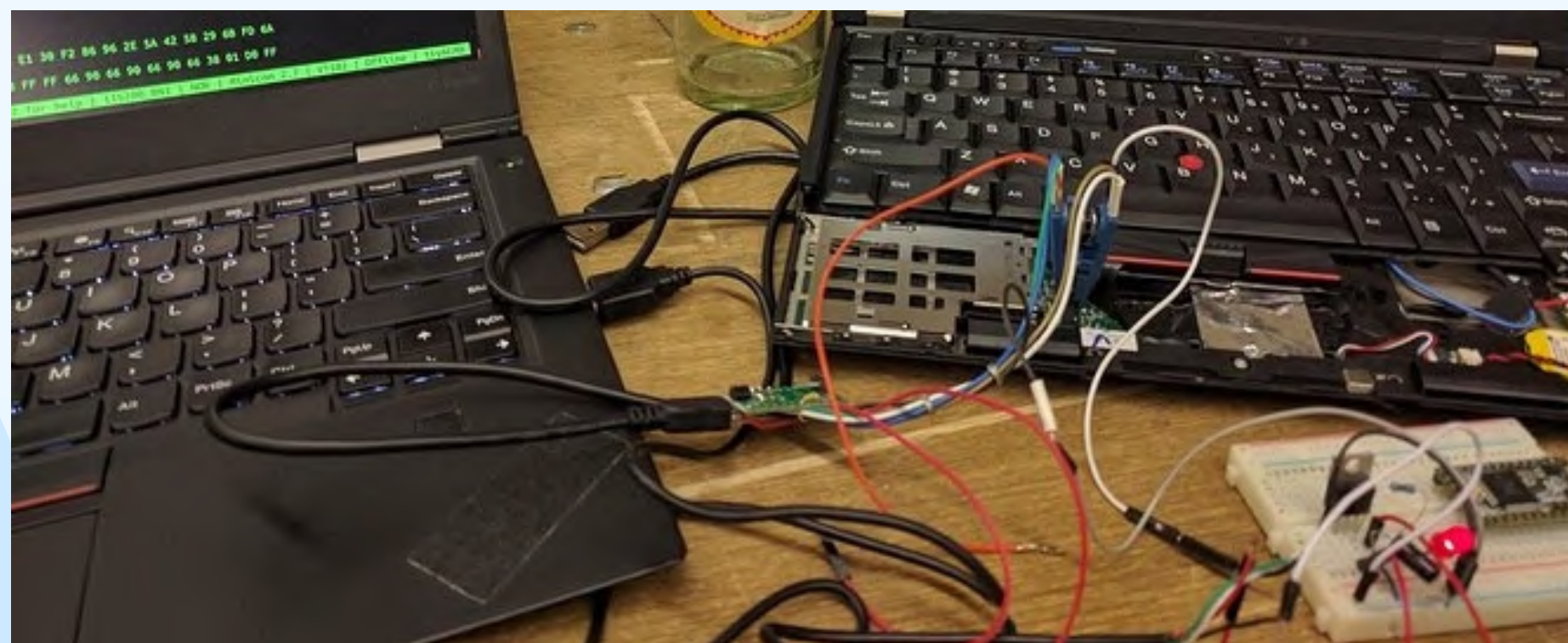
Upcoming developments?

- NixOS pure build environment
- Haswell family native raminit
- Write protection changes

Deeper drive

- Check out tlaurions (Insurgo) Fosdem 2023 presentation and slides;

https://fosdem.org/2023/schedule/event/heads_status_update/attachments/paper/5659/export/events/attachments/heads_status_update/paper/5659/Heads_status_update.pdf



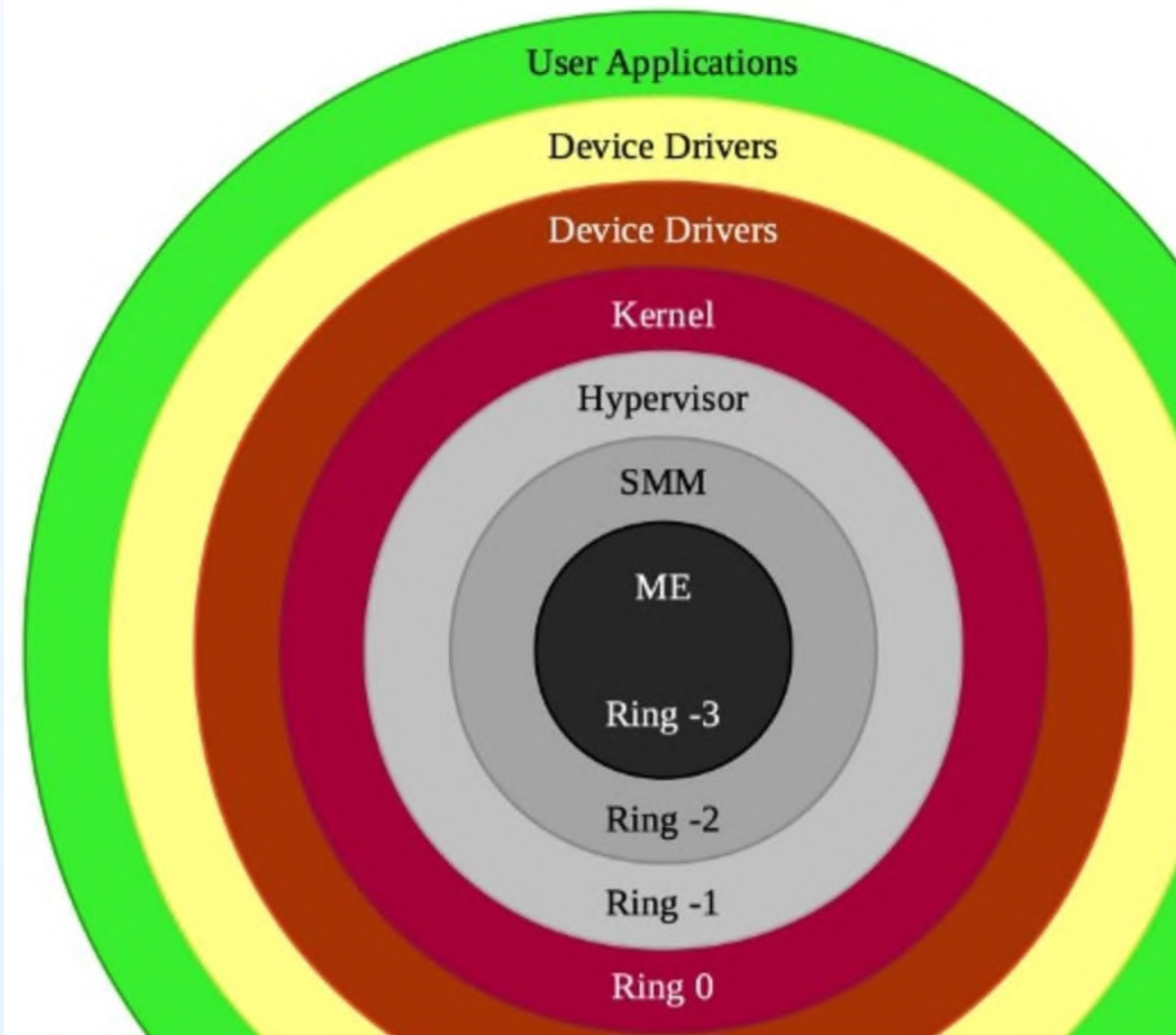
How does it actually look?





- It is a system inside your Intel x86 system, with its fingers in all of your pies.
- A microcontroller with its own processor, memory, and I/O and running closed firmware
- Has direct memory access, can inspect your CPU and can talk to network hardware.
- Etched into the CPU die, so not removable
- Absolute level of pwn over an x86 system
- Some described ME as a kind of "Ring -3"
- Networking ability – remote access
- Adversary with ME access owns your machine

Intel Management Engine



Intel Management Engine

Neuter

- The ME firmware is contained in the vendor ROM
- Without ME being operative on intel x86, a "reset" timer starts which forces your machine to reboot after 30 minutes of uptime
- No one knows just who or what can use it – there is plenty of speculation online.
- Warning: Deep "rabbit hole" with lots of opinion
- What we do know has come from leaked documentation and reverse engineering
- Good news ... ways to neuter it have been found!



IME before neuter



IME after neuter



+



Linux Kernel in ROM

Intel Management Engine

First shoutout goes to the NSA!

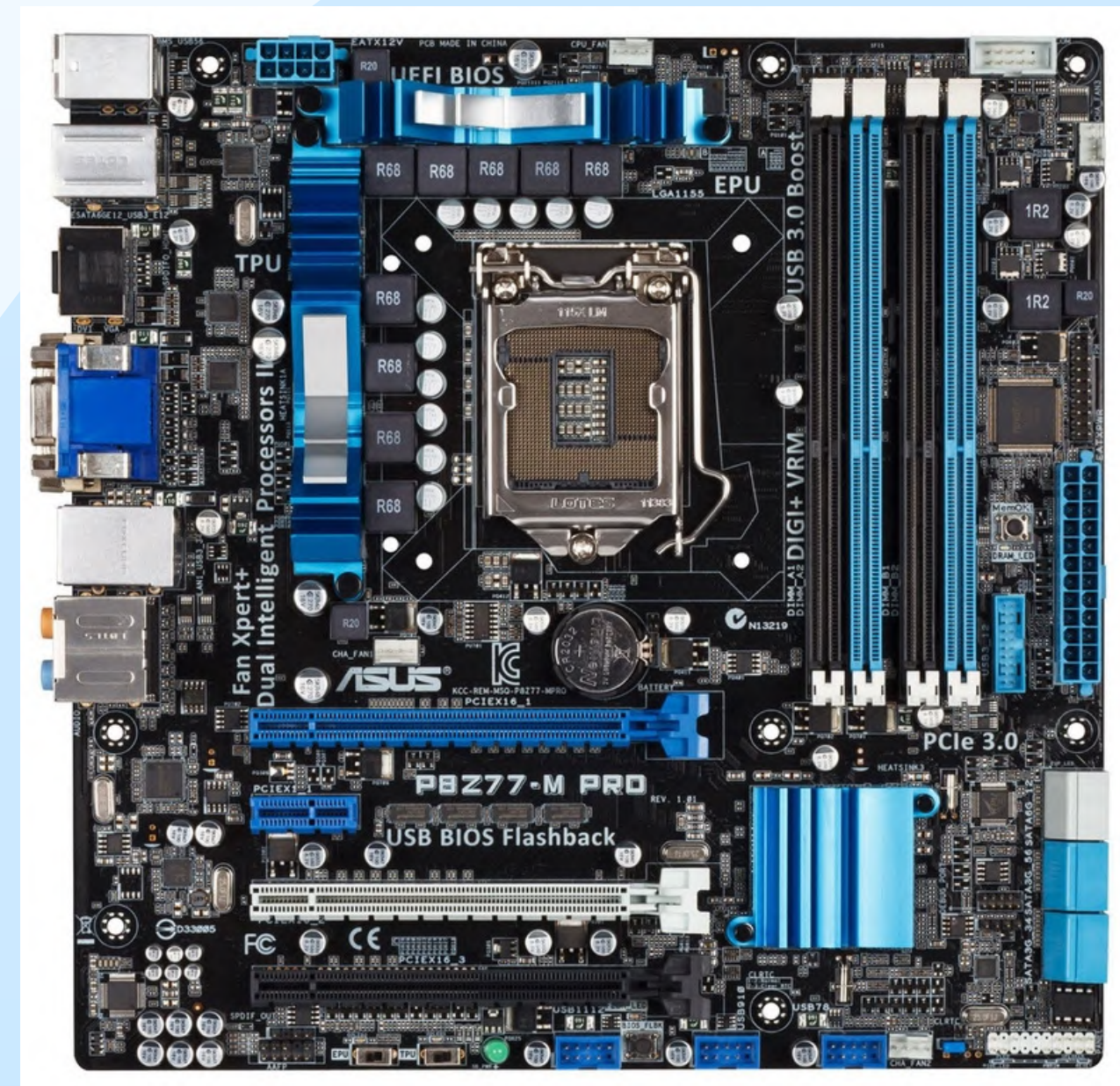
- It was discovered that the NSA required intel to add a 'kill switch' to ME for X86 systems used in their "High Assurance Platform" – We know where to find it, and heads firmware enables this. Thanks NSA!
- Also discovered on earlier CPUs: when ME is looped in a startup/BUP state, the reboot timer stops
- Nicola Corna wrote the me_cleaner utility which removes all partitions from the ME blob, (except the code needed to start booting the ME system). Heads uses this utility. https://github.com/corna/me_cleaner
- ME now stays in the BUP state, non functional, but this stops the reboot timer
- The unknown ME firmware can now be shrunk from 4MB to about 98KB
- Heads takes up the freed up space. Any attempt to change the ME size will change heads, resulting in changed TPM measurements and the HOTP and TOTP failing
- I introduced the optional use of VSCC deletion to Heads firmware recently, aiming to prevent ME ROM writeback

New to Heads – mATX Intel x86 Desktop

- Recently I ported the P8Z77-M Pro motherboard
- This is first 'traditional' non-mini Intel Desktop x86 Platform in the Heads firmware project
- Requires no complex disassembly process or SOIC clips to flash the ROM chip
- Enough power to run plenty of tools in QubesOS
- The DIP8 socketed ROM is removable,

Spec

- Ivybridge CPU (not all have VT-d QubesOS requirement)
- Up to 32GB RAM
- PCI-e expansion ports (for number crunching GPUs?)
- 6x SATA Ports 2x eSATA, loads of USB 2 and 3, VGA, HDMI
- VT-x & VT-d (QubesOS requirement)
- PS/2 Keyboard port (QubesOS recommendation)
- Discreet TPM module header (Heads requirement)





Physical Security

Tamper Protection

Physical access

With firmware now in a known good state.....

Threat: Could adversary tamper inside case?

Threat: Could adversary tamper with the SSD?

Threat: Could adversary tamper with the TPM?

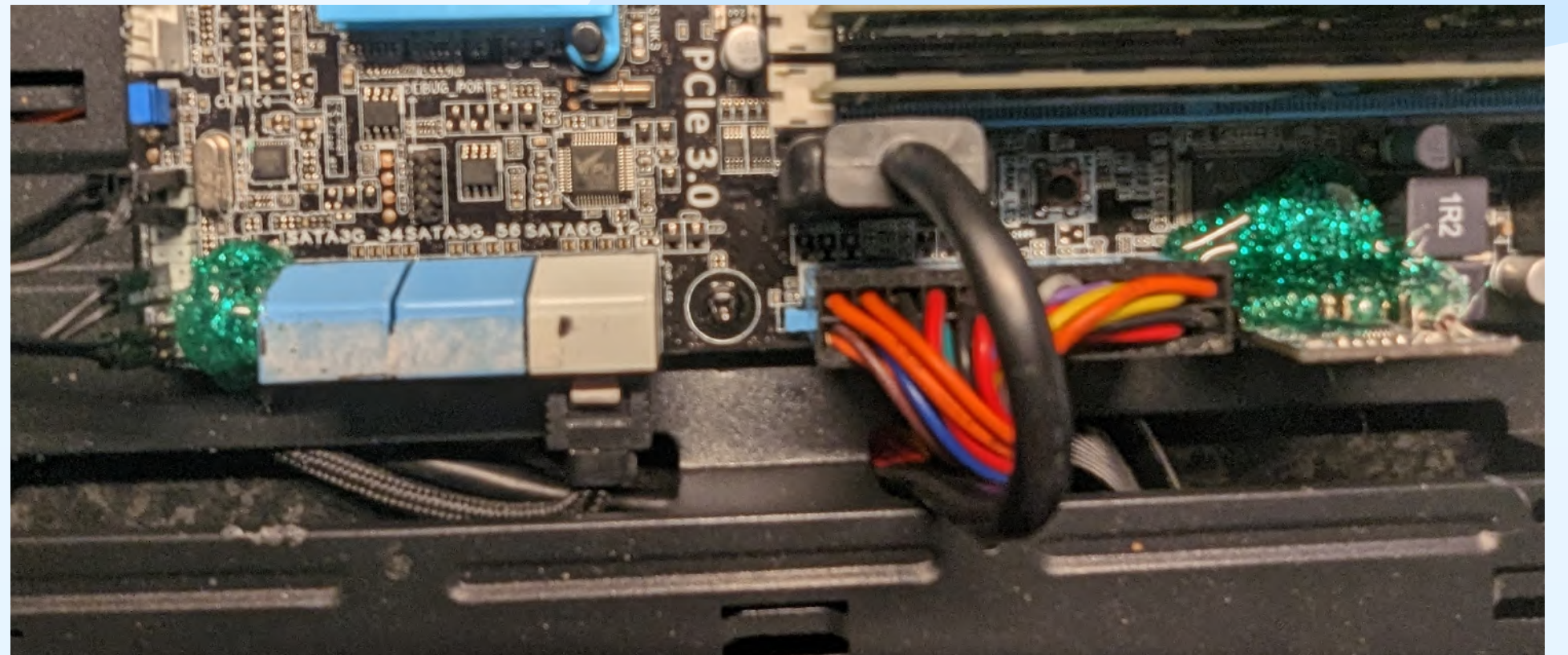
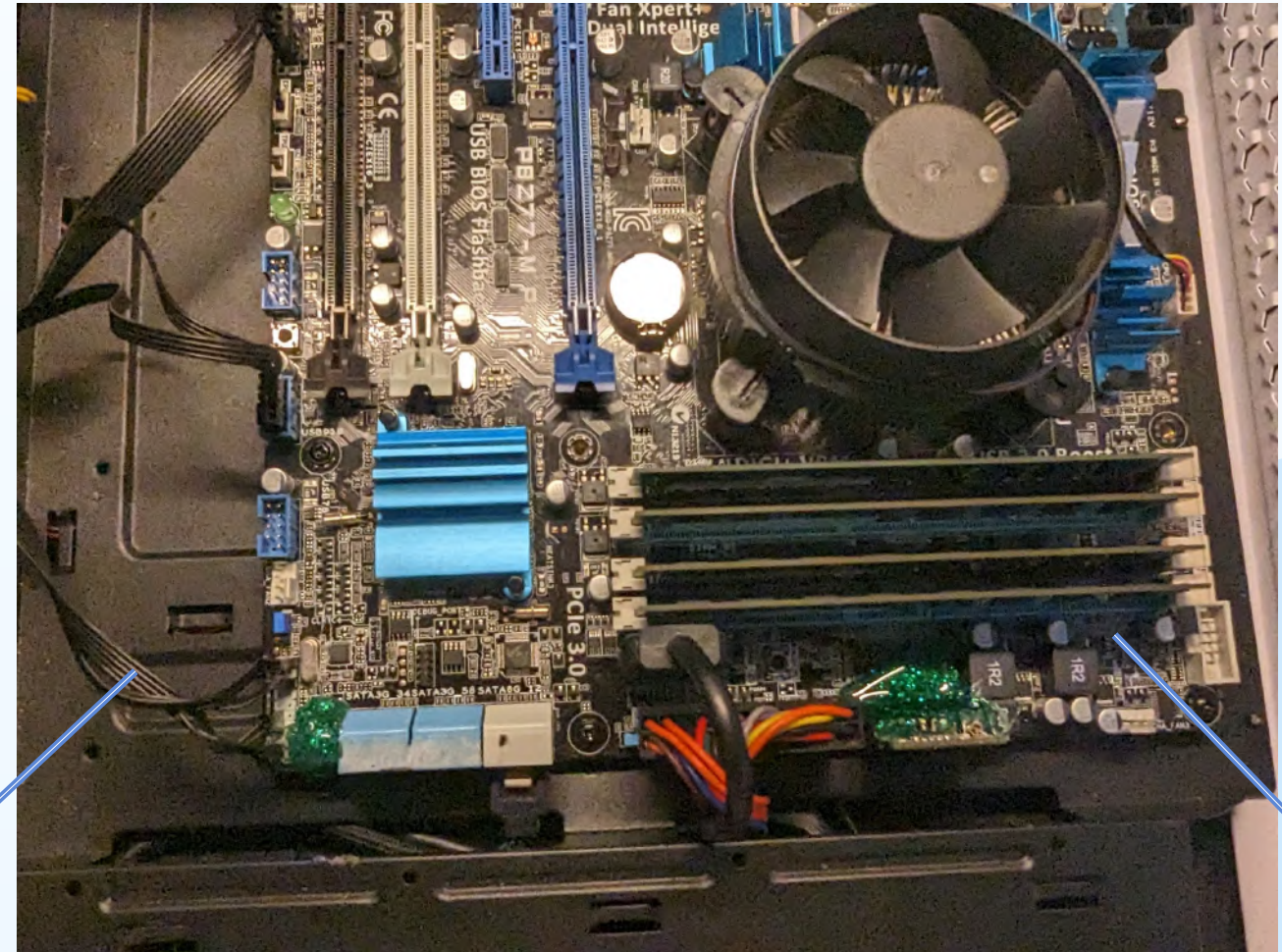
Threat: Could adversary tamper with the ROM?

.....**We need visual tamper protection!**



TPM & ROM: Glitter Glue

A great one time, unique seal
(plus a sparkly motherboard)



Numbered Tamper Seals

Front



Rear



SSD



Now what?

We have a tamper proof machine

It is a known good state

What OS do we install on it?

.....Qubes OS!



Qubes OS

Requirements 4, 5, 6 & 7

- Practical, usable security to vulnerable and actively-targeted individuals, such as journalists, activists, whistleblowers, and researchers.
- Designed with the understanding that people make mistakes, and it allows you to protect yourself from your own mistakes.
- It's a place where you can click on links, open attachments, plug in devices, and install software free from worry.
- It's a place where *you* have control over your software
- Full disk encryption using luksCrypt



Qubes OS

- Open source, security-oriented operating system
- Xen-based virtualization, creation & management of isolated compartments called qubes, seamlessly integrated together in one Desktop
- Unified desktop environment with unforgeable colored window borders
- Qubes have:
 - **Purposes:** with a predefined set of one or many isolated applications, for personal or professional projects, to manage the network stack, the firewall, or to fulfill other user-defined purposes.
 - **Natures:** full-fledged or stripped-down virtual machines based on popular operating systems, such as Fedora, Debian, and Windows.
 - **Levels of trust:** from complete to non-existent.

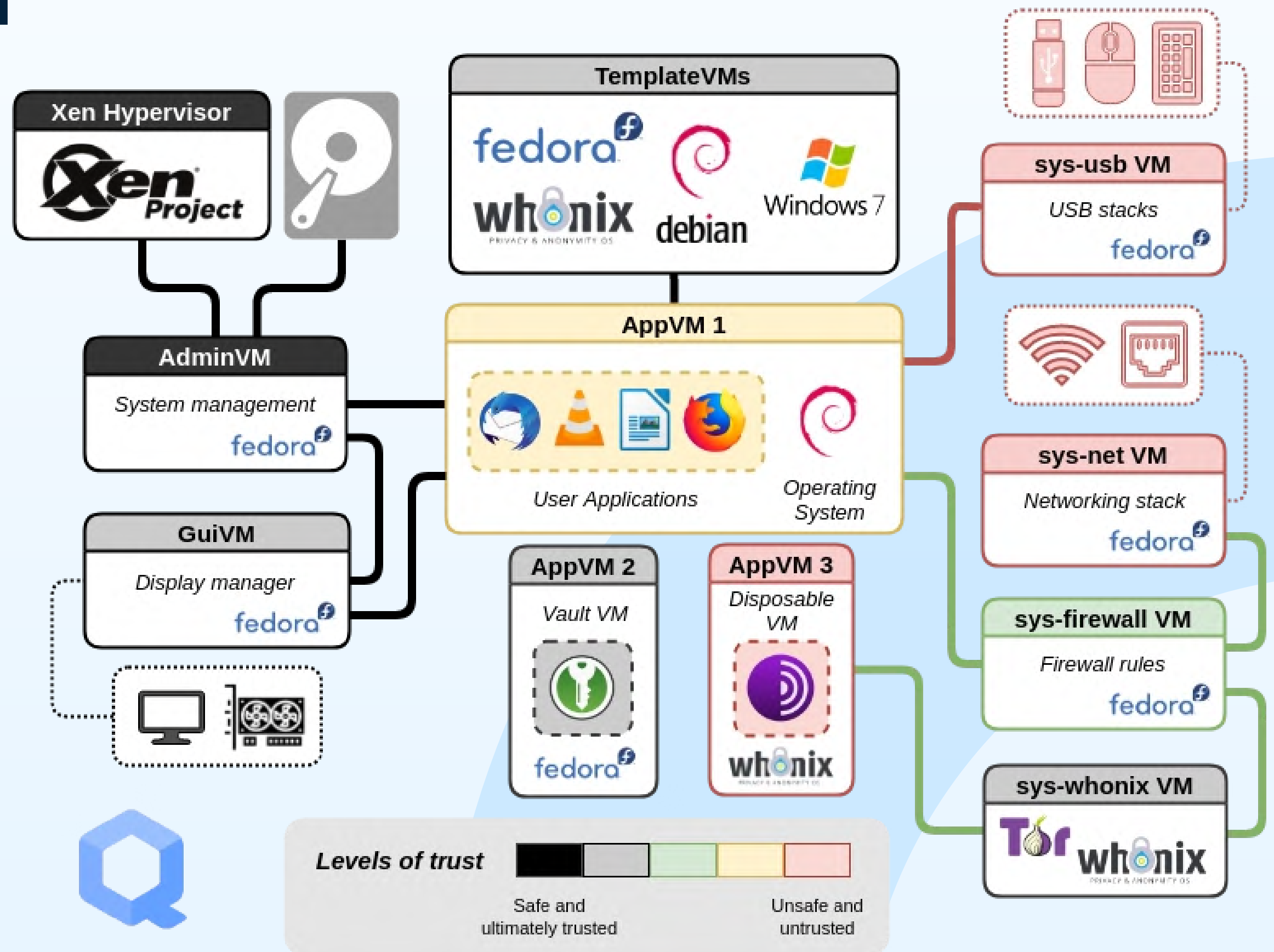


Qubes OS

1. **Strong isolation:** It helps to isolate software as if they were installed in separate machines. It uses PV or HVM virtualization techniques for isolation.
2. **Multiple Operating Systems:** It allows the user to use multiple operating systems at the same time, such as Fedora, Debian, Kali, Whonix, Windows & more.
3. **Templates:** Qubes spawn from air gapped OS templates. Changes to the OS in a PV Qube do not affect the template from its 'known good' state.
4. **Controller Isolation:** It helps the user to secure device handling through isolation of network cards and USB controllers.
5. **Configurable networking:** Bridge Qubes, isolate Qubes, route via VPN, TOR, your ISP
6. **U2F proxy:** It helps to use two-factor authentication by using Qubes U2F proxy.
7. **Disposable virtual machines:** Disposable VMs are lightweight, created quickly and get disappear when closed. Anything that happens in them disappears at shutdown.

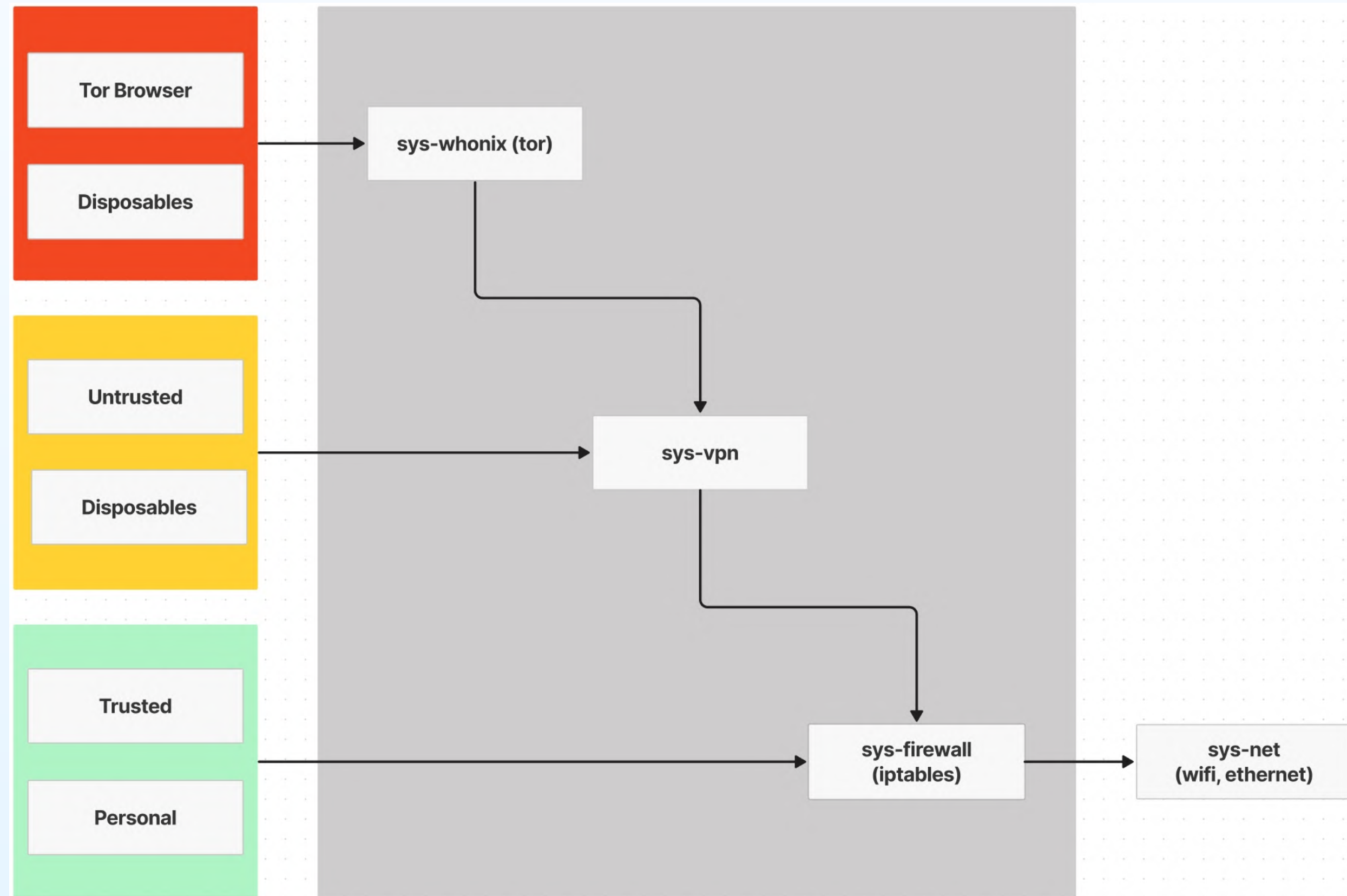
Qubes OS System

- Looks Complex
- Actually simple to grasp
- Some next level protection



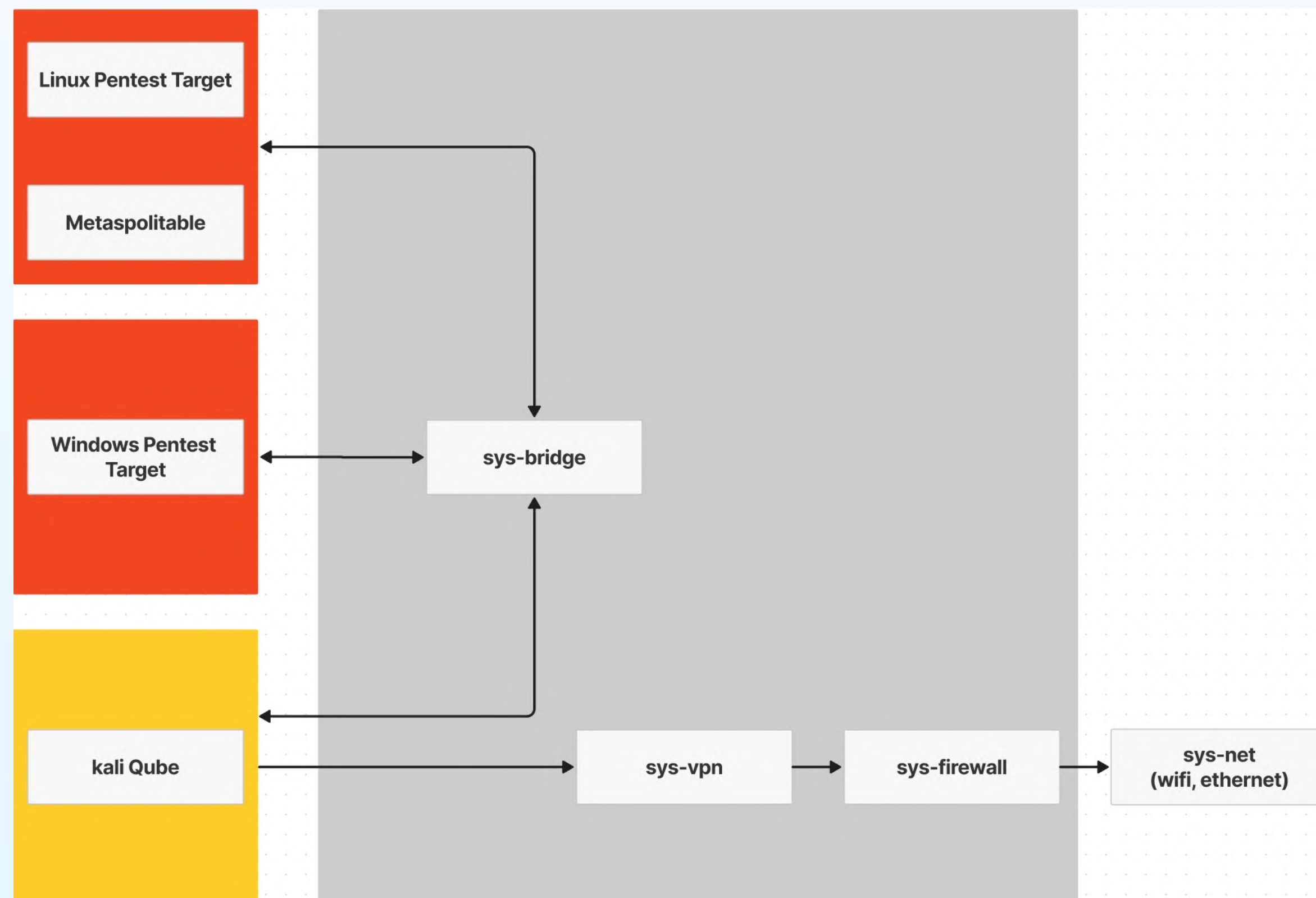
Qubes OS Networking – Example

- Define topology by selecting which sys qube to use for networking
- networking qubes can be chained
- sys-net has PCI passthrough of actual network device via IOMMU
- sys-firewall is iptables NAT
- sys qubes can be configured and made disposable/immutable



Qubes OS Networking – Example

- Pentest bridge example
- Kali still has networking for updating sploit DBs etc (optional)
- Lots of possibilities
- Test targets could easily be given outbound connectivity via bridge (eg, test XSS, user interaction requirements/link clicking/mitm etc)

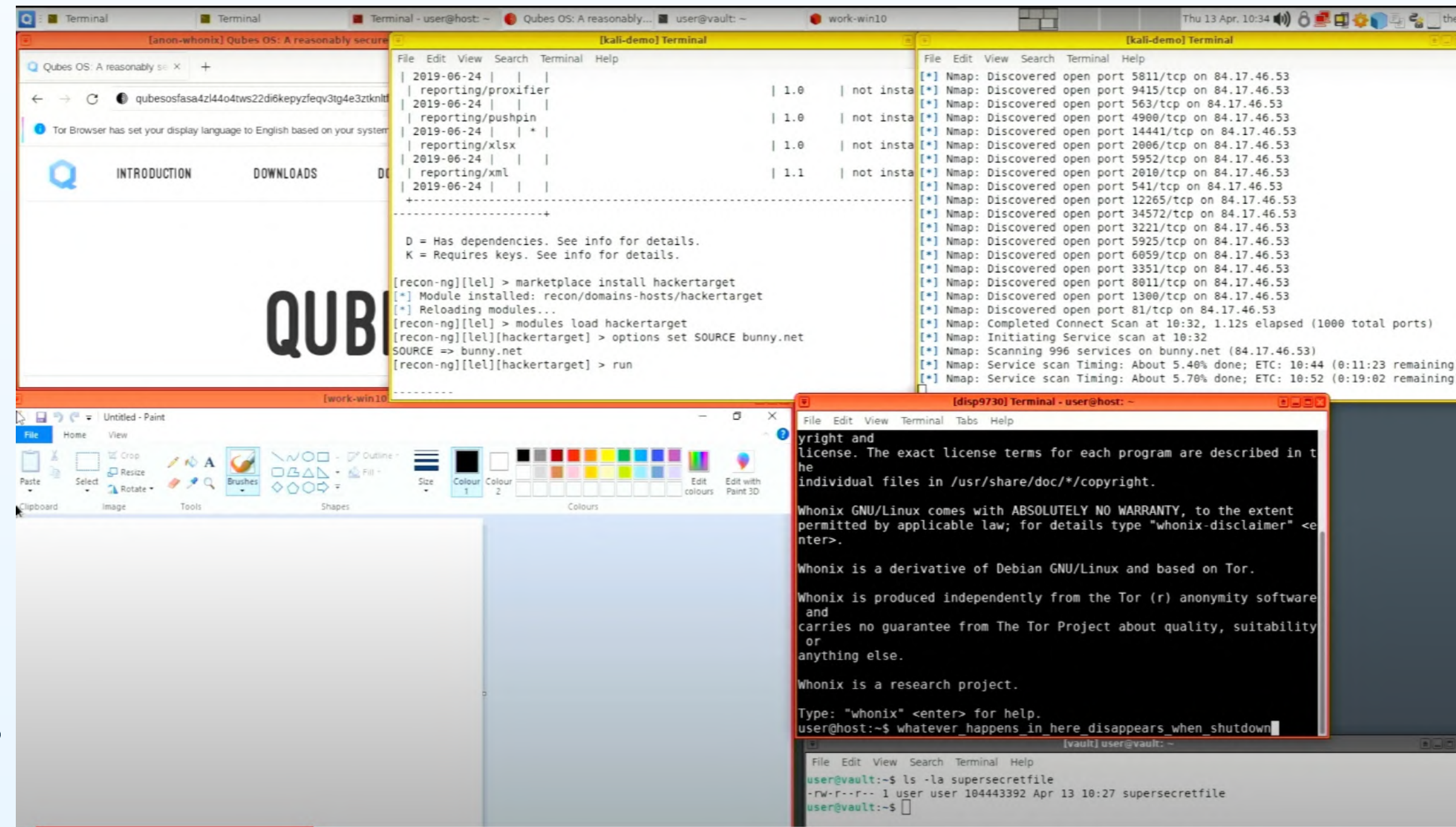


This is one Desktop with;

- recon-ng in Kali Qube
- metasploit – nmap in Kali Qube
- Whonix browser Qube
- Secure Vault Qube
- Disposable VM Debian Qube
- MS paint in Windows 10 Qube
- All with different trust levels
- All with different networking permissions & routing

A few ideas.....

- Remote pentest between Qubes
- Payload inspection
- Run Metaspolitable: <https://forum.qubes-os.org/t/creating-a-metasploitable-vm-lab-in-qubes-4-1>
- Org sec audits: Deeplows great guide : <https://forum.qubes-os.org/t/qubes-for-organizational-security-auditing-talk-notes/>
- Seclab in a box: <https://forum.qubes-os.org/t/recommended-ways-to-setup-security-labs-pentesting-lab-malware-lab/>




Qubes OS – Endorsements



"Qubes OS gives us greater confidence in the security of systems being used to remotely access our servers, mainly because powerful physical and logical privilege separation between workspaces allows our engineers to select appropriate degrees of isolation for different processes."


— Let's Encrypt, non-profit, world's largest certificate authority



"Fun fact: We use @QubesOS on a majority of our workstations since a while back, and boy do we like it! #qubes #security #privacy #vpn" 

— Mullvad, privacy-focused VPN provider



"If you're serious about security, @QubesOS is the best OS available today. It's what I use, and free. Nobody does VM isolation better." 

— Edward Snowden, whistleblower and privacy advocate



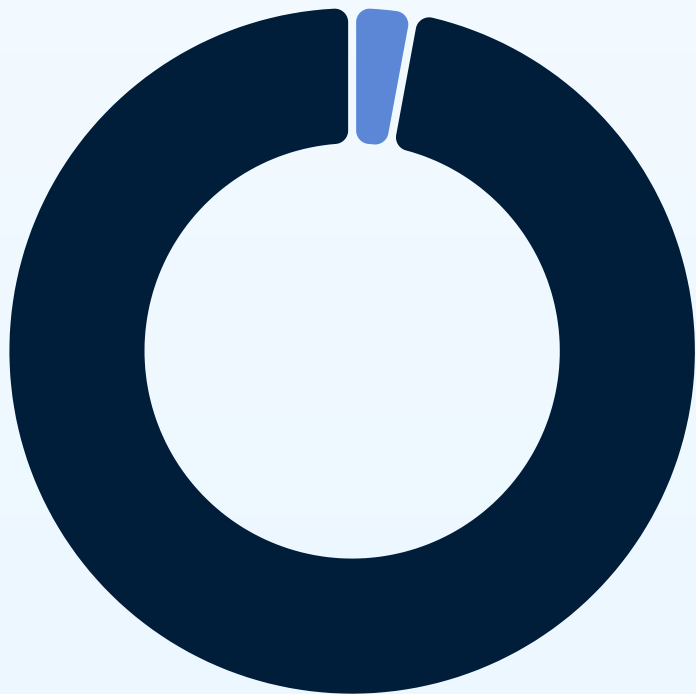
"When I use Qubes I feel like a god. Software thinks that it's in control, that it can do what it wants? It can't. I'm in control."

— Micah Lee, Director of Information Security at The Intercept, advisor to DDoSecrets

Qubes OS Adoption



Researchers who should use QubesOS



Researchers who already use QubesOS



Get Involved

Ready to help?

Apply for our first **CloudSec Intern program** @ bunny.net

Gain real industry experience on a hyperscale cloud platform – come grab a coffee chat today

Check out <https://bunny.net/careers/>

For those already in the industry and looking for your next opportunity

QubesOS Project – <https://www.qubes-os.org/>

Heads Firmware & Community – <https://osresearch.net/>

QubesOS Community <https://forum.qubes-os.org/>

Coreboot & Community – <https://coreboot.org/>

