Red Hat

KVM FORUM

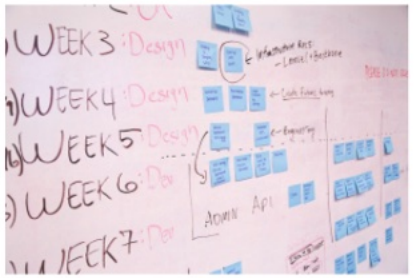
Chains of trust in Confidential Computing

Knowing and verifying what you run

Christophe de Dinechin
Senior Principal Software Engineer
cddd@redhat.com

Agenda

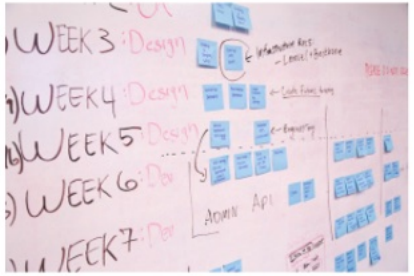
Key topics we are going to cover today



- Overview of Confidential Computing
- What is Attestation?

Agenda

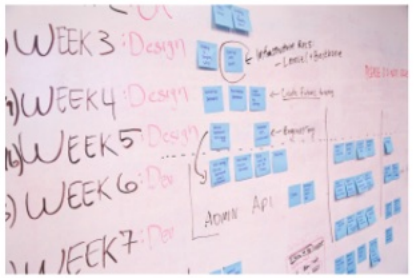
Key topics we are going to cover today



- Overview of Confidential Computing
- What is Attestation?
- Use cases for Confidential Computing

Agenda

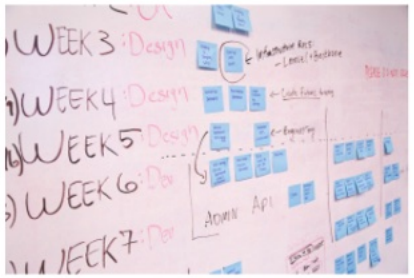
Key topics we are going to cover today



- Overview of Confidential Computing
- What is Attestation?
- Use cases for Confidential Computing
- From root of trust to actual trust
- Platform-specific details

Agenda

Key topics we are going to cover today



- Overview of Confidential Computing
- What is Attestation?
- Use cases for Confidential Computing
- From root of trust to actual trust
- Platform-specific details
- Supporting technologies

- See blog for more details

Red Hat

KVM FORUM

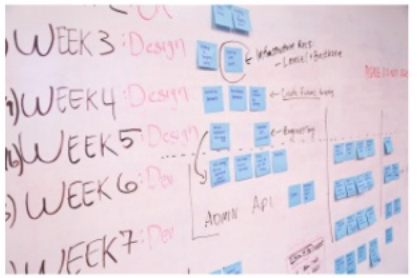
Chains of trust in Confidential Computing

Knowing and verifying what you run

Christophe de Dinechin
Senior Principal Software Engineer
cddd@redhat.com

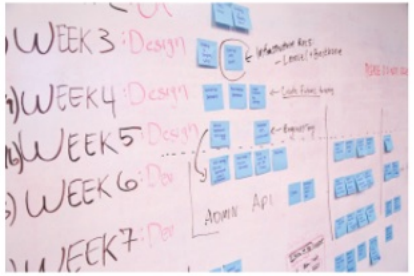
Agenda

Key topics we are going to cover today



Agenda

Key topics we are going to cover today



- Overview of Confidential Computing

Red Hat

KVM FORUM

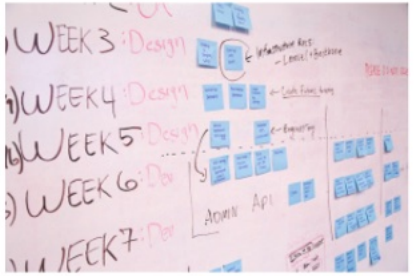
Chains of trust in Confidential Computing

Knowing and verifying what you run

Christophe de Dinechin
Senior Principal Software Engineer
cddd@redhat.com

Agenda

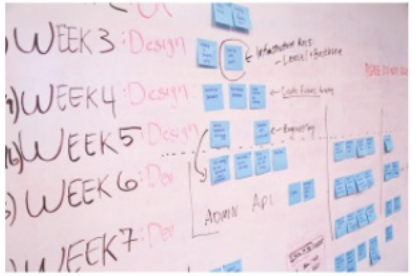
Key topics we are going to cover today



- Overview of Confidential Computing

Agenda

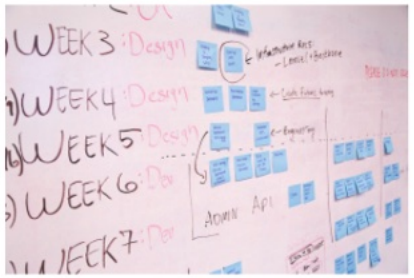
Key topics we are going to cover today



- Overview of Confidential Computing
- What is Attestation?

Agenda

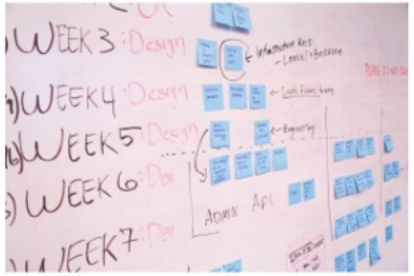
Key topics we are going to cover today



- Overview of Confidential Computing
- What is Attestation?
- Use cases for Confidential Computing

Agenda

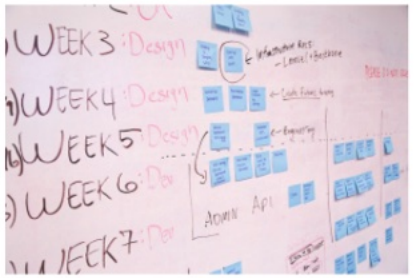
Key topics we are going to cover today



- Overview of Confidential Computing
- What is Attestation?
- Use cases for Confidential Computing
- From root of trust to actual trust

Agenda

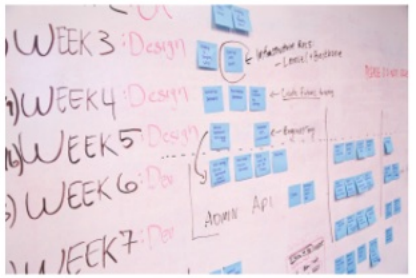
Key topics we are going to cover today



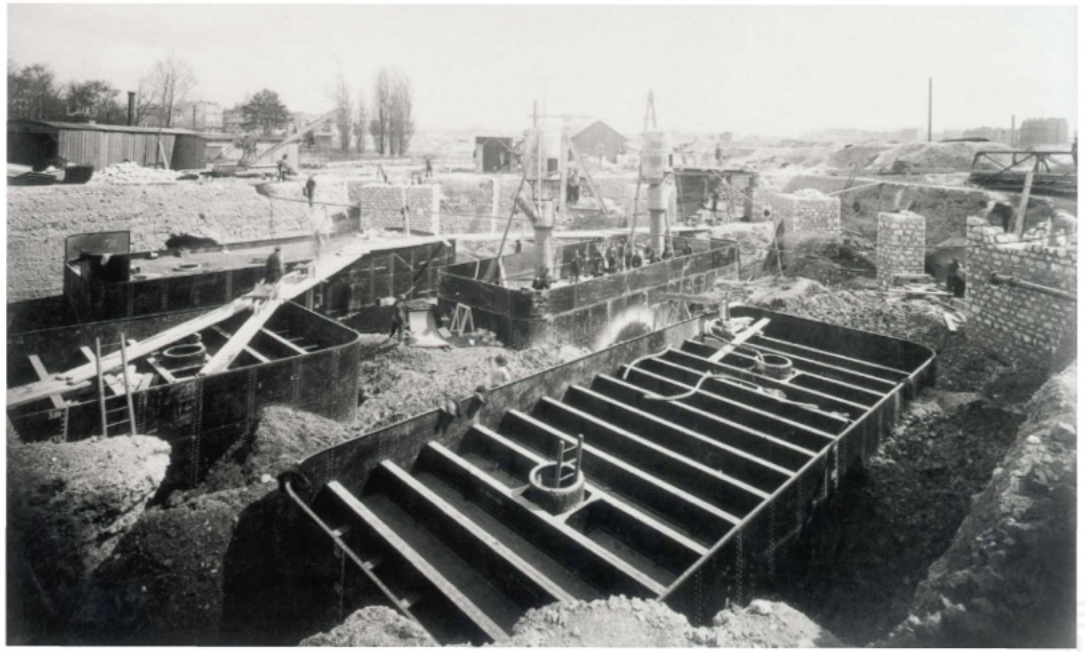
- Overview of Confidential Computing
- What is Attestation?
- Use cases for Confidential Computing
- From root of trust to actual trust
- Platform-specific details

Agenda

Key topics we are going to cover today

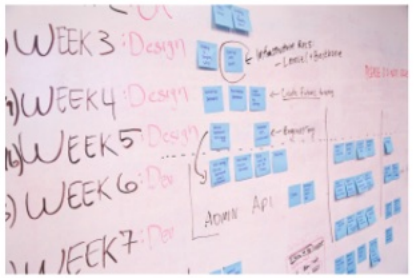


- Overview of Confidential Computing
- What is Attestation?
- Use cases for Confidential Computing
- From root of trust to actual trust
- Platform-specific details
- Supporting technologies



Agenda

Key topics we are going to cover today



- Overview of Confidential Computing
- What is Attestation?
- Use cases for Confidential Computing
- From root of trust to actual trust
- Platform-specific details
- Supporting technologies

- See blog for more details

KVM FORUM

Confidential Computing

Protecting data in-use

Red Hat

KVM FORUM

Confidential Computing

Protecting data in-use

I compromised the confidentiality of their proprietary software to advance my agenda of becoming the best at breaking through the lock.

Kevin Mitnick

Red Hat

Software now runs on hardware you do not own, like a cloud provider

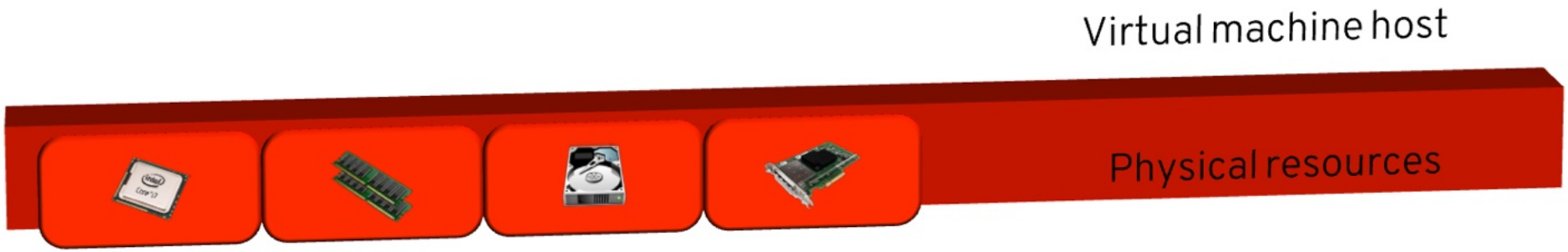
Problem Statement

Why should infrastructure see your data?

Hardware resources are owned by the host

Problem Statement

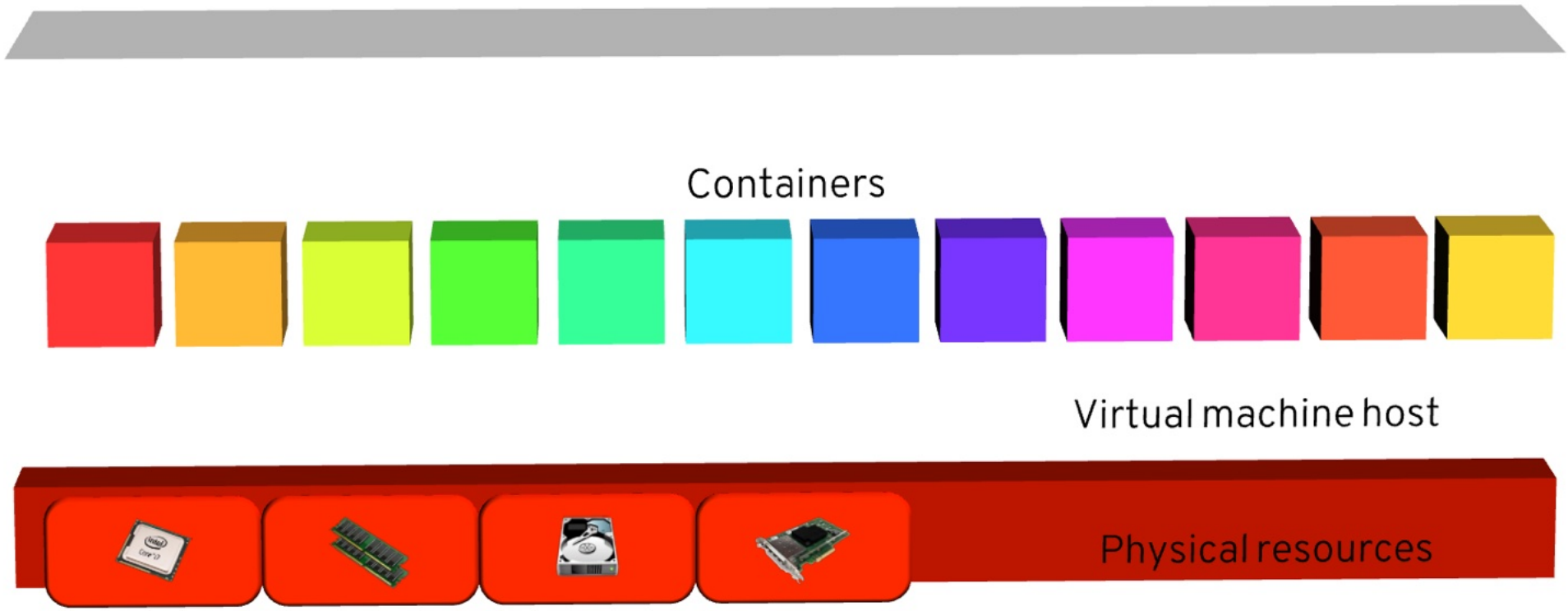
Why should infrastructure see your data?



Containers carve out resources from the host

Problem Statement

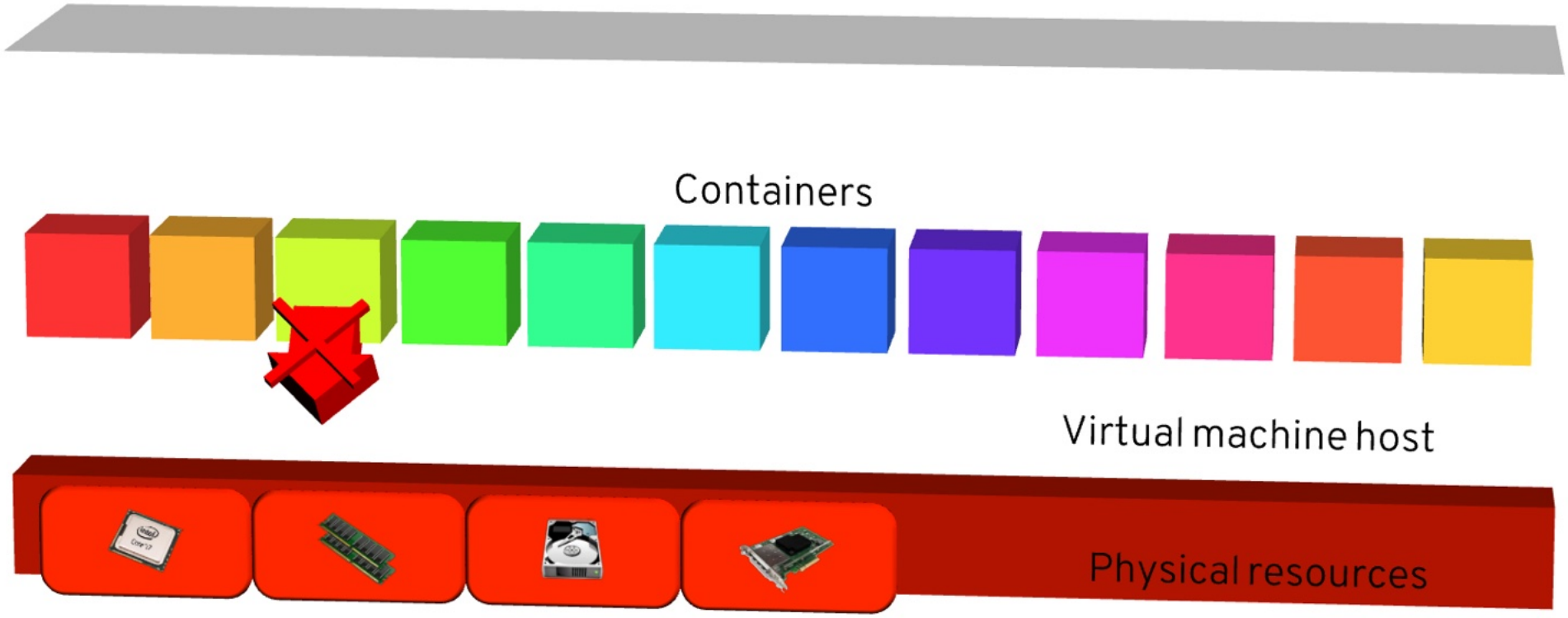
Why should infrastructure see your data?



Classical sandboxing only protects the host from the containers running on it

Problem Statement

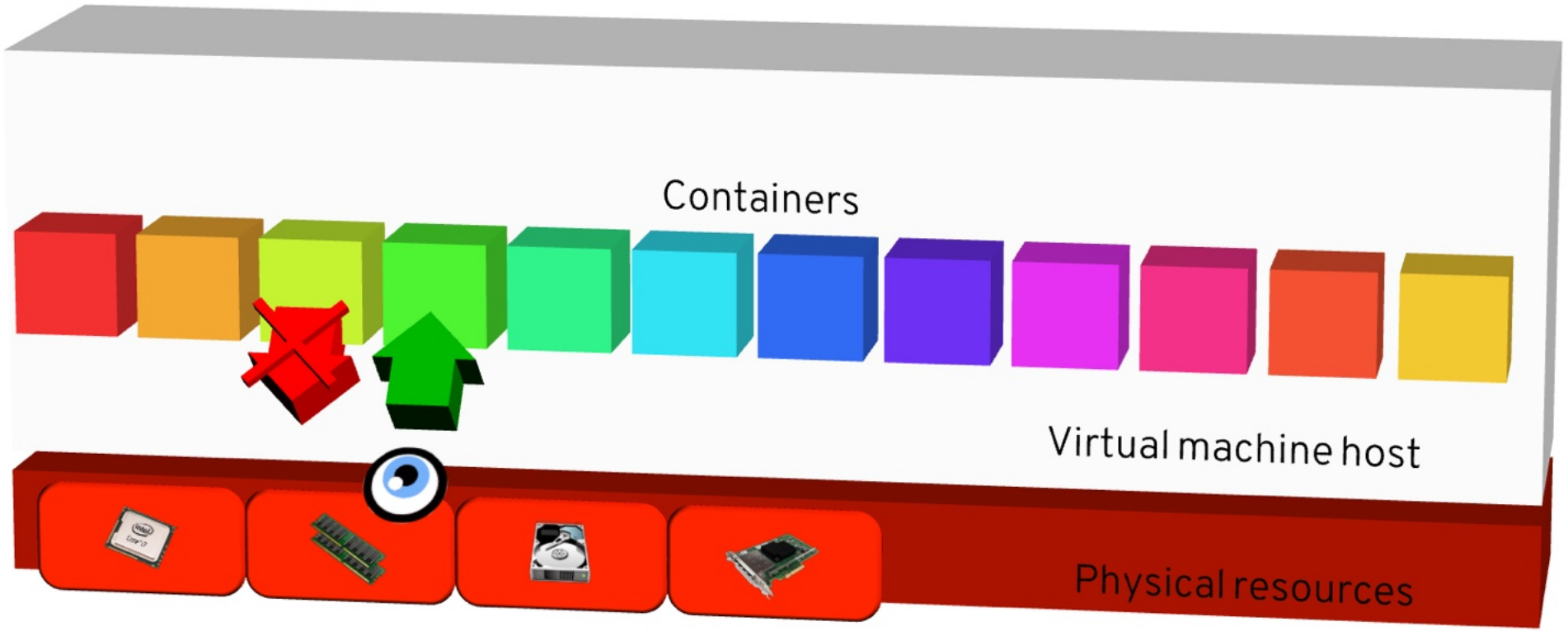
Why should infrastructure see your data?



The host can freely peek inside the container, for example read its memory

Problem Statement

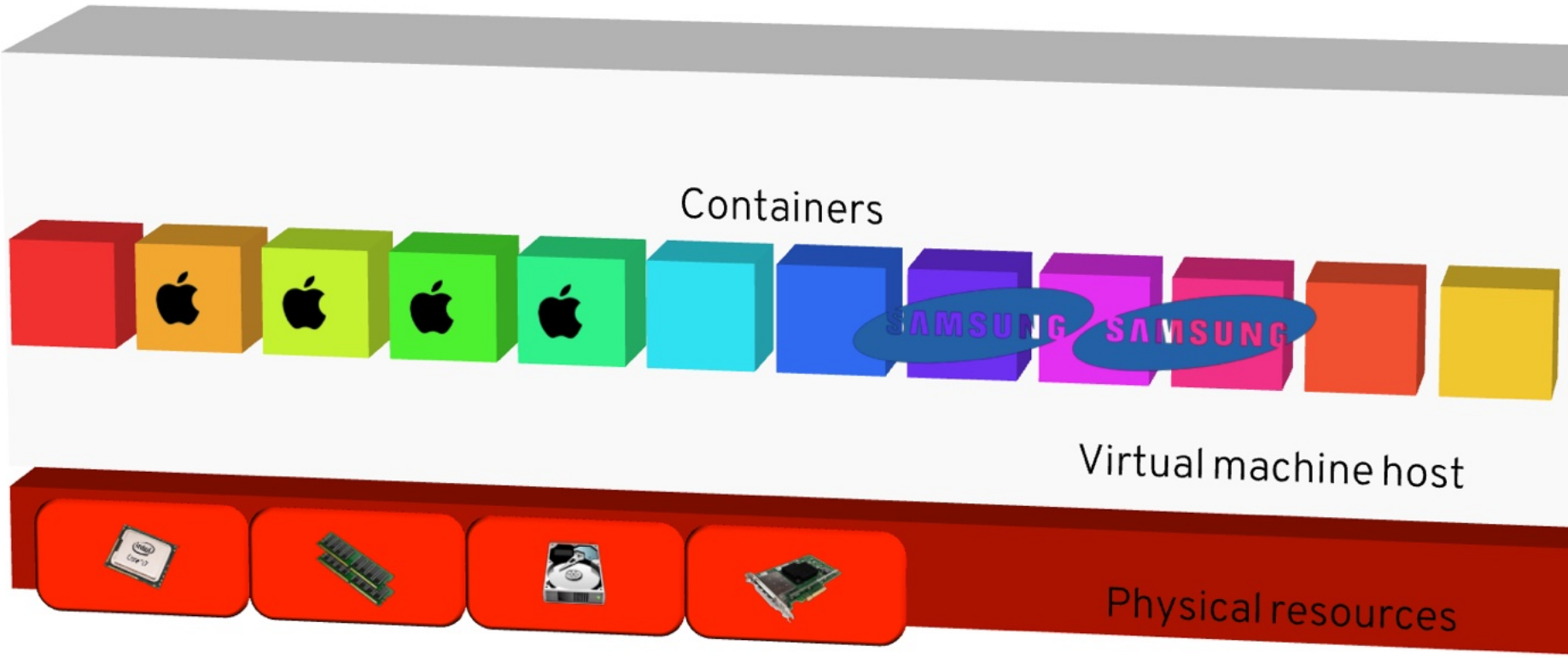
Why should infrastructure see your data?



For that reason, multiple tenants do not want to share hardware when processing sensitive data

Problem Statement

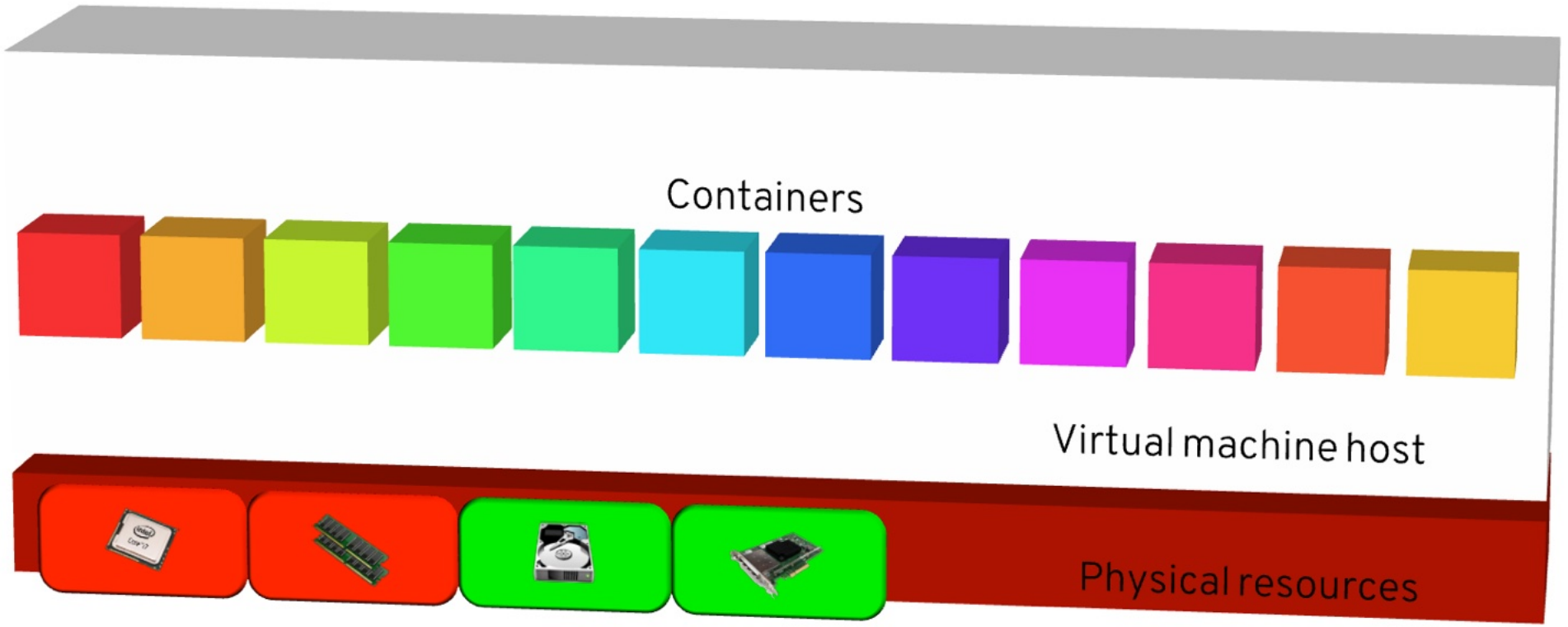
Why should infrastructure see your data?



Data on disk or in network is already encrypted today, so the VM host cannot read it nor tamper with it

Problem Statement

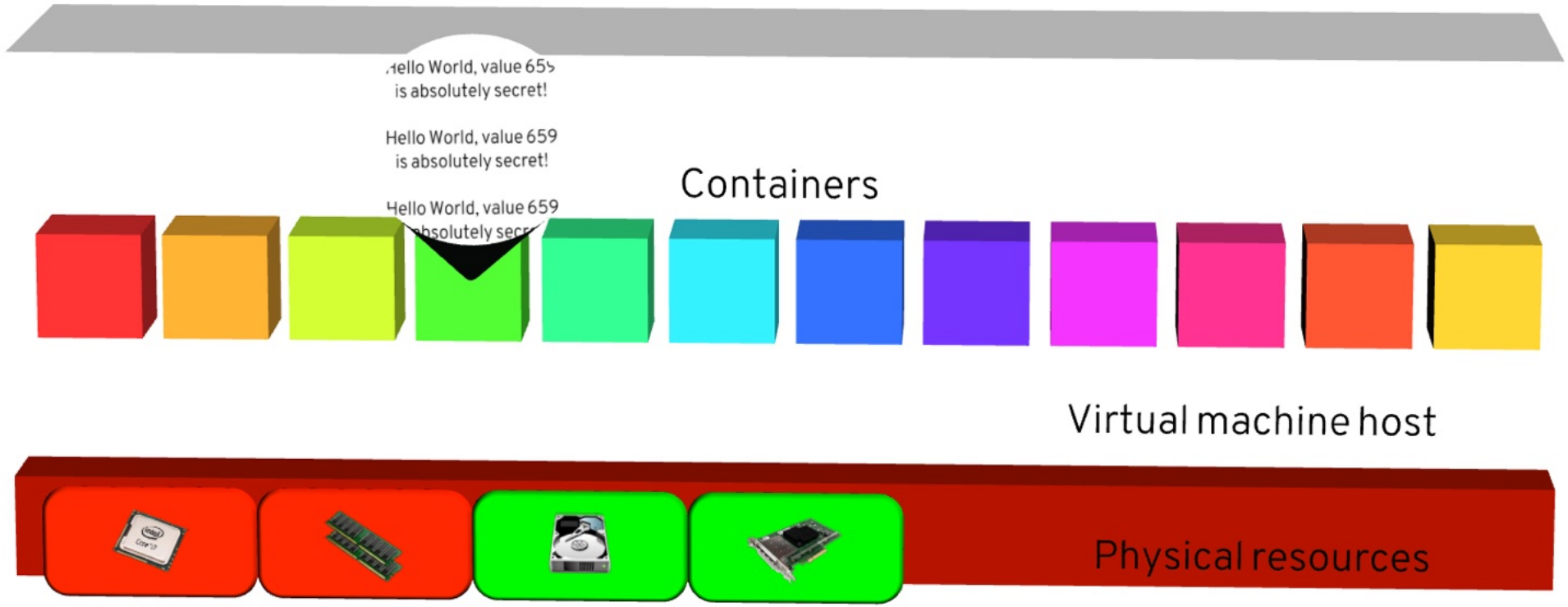
Why should infrastructure see your data?



In non-CC architectures, data in memory is not encrypted, so it can be accessed by the host

Problem Statement

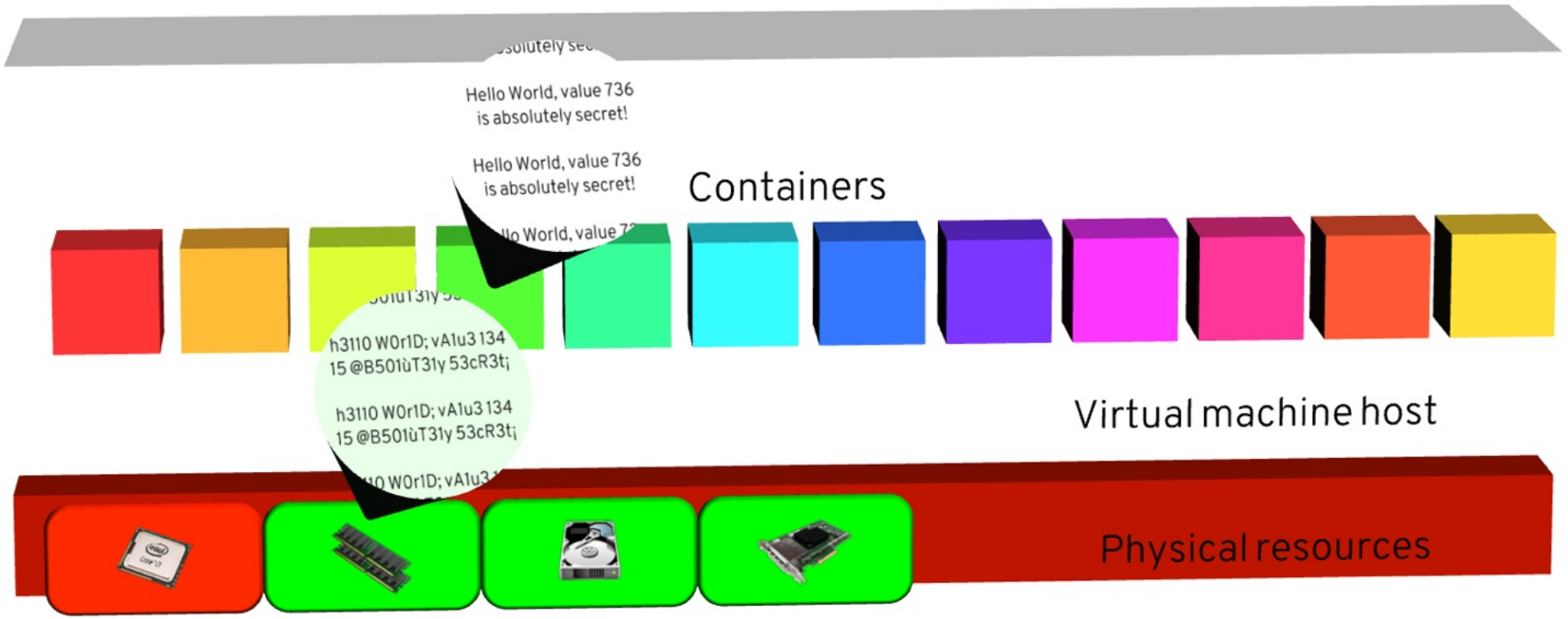
Why should infrastructure see your data?



Encrypted memory stores live data as cypher text, so that it becomes garbled when read by the host

Problem Statement

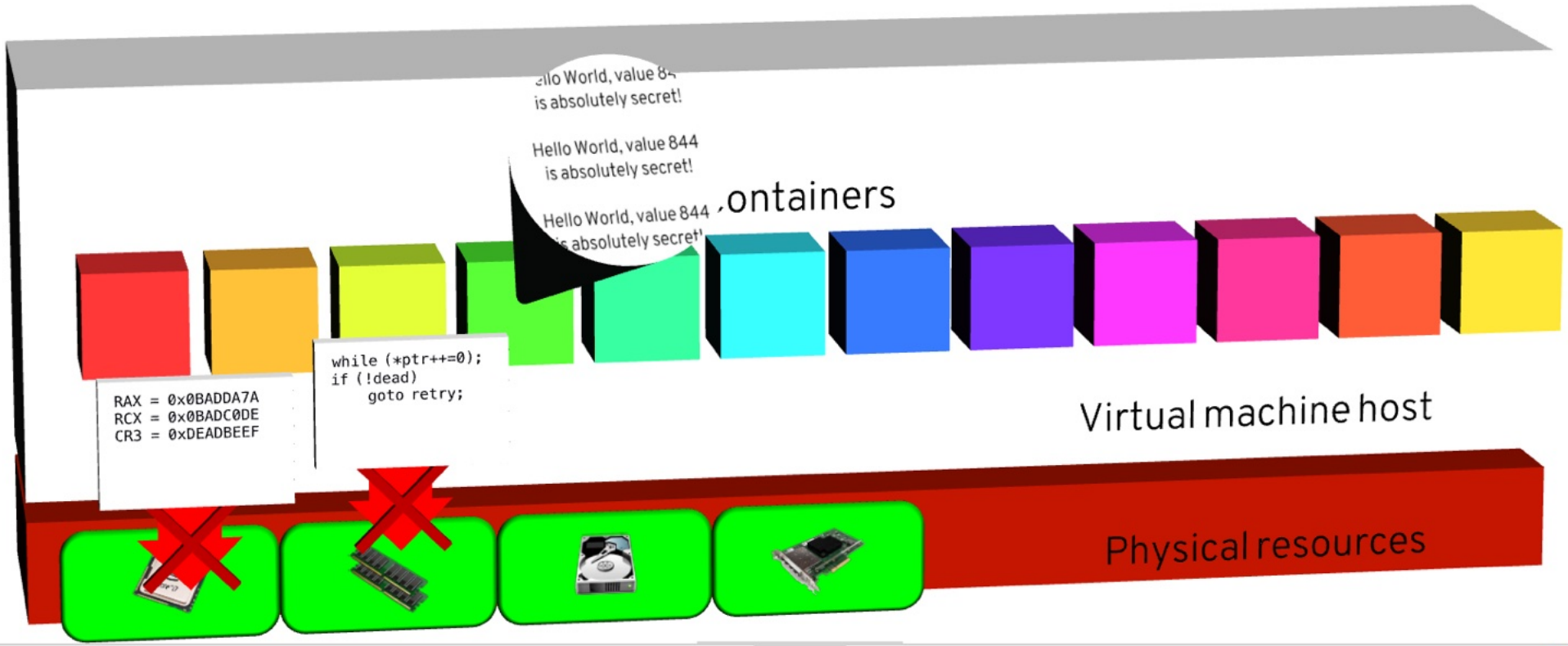
Why should infrastructure see your data?



Integrity ensures the host cannot corrupt nor poison CPU state or RAM contents

Problem Statement

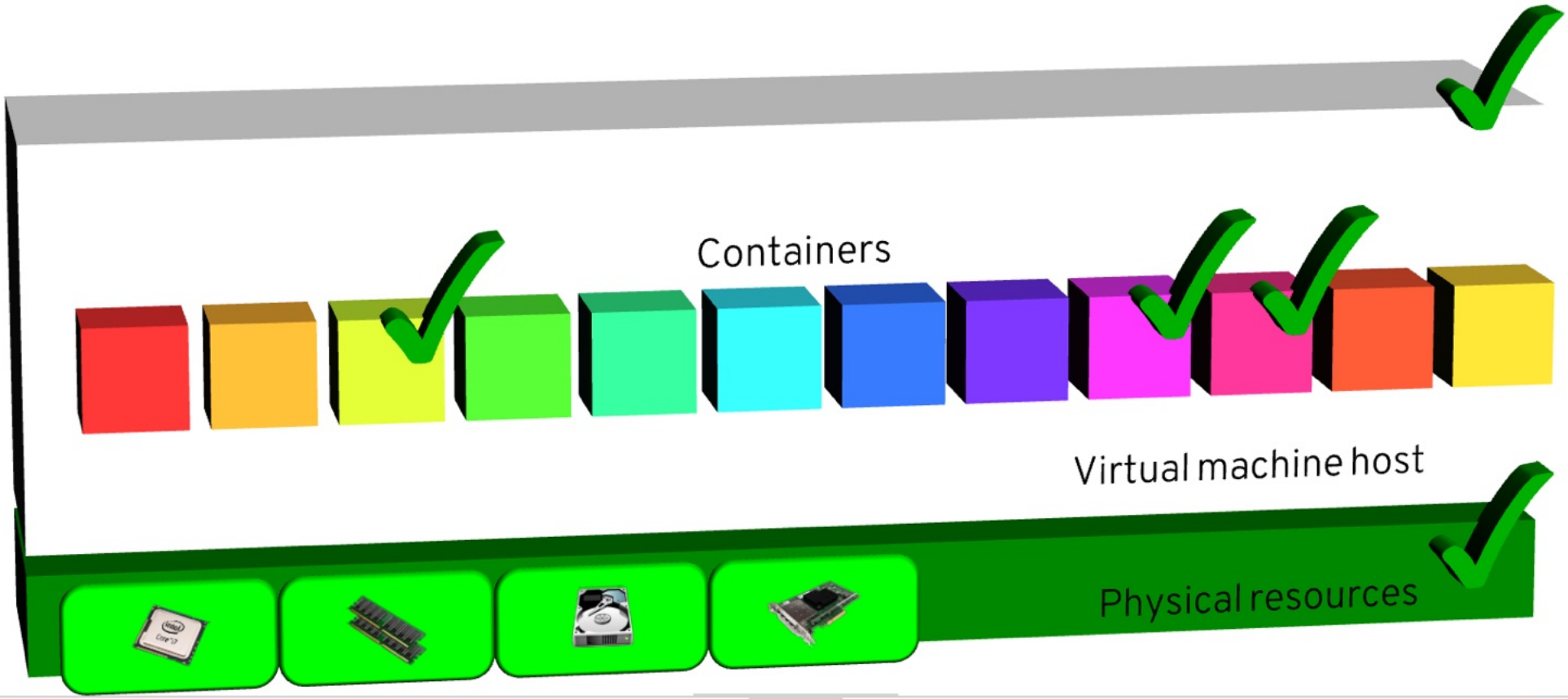
Why should infrastructure see your data?



Attestation proves where you are running and what you are running

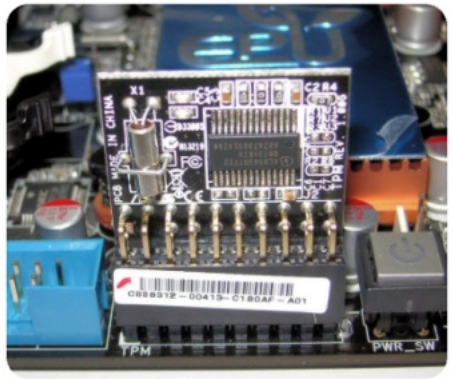
Problem Statement

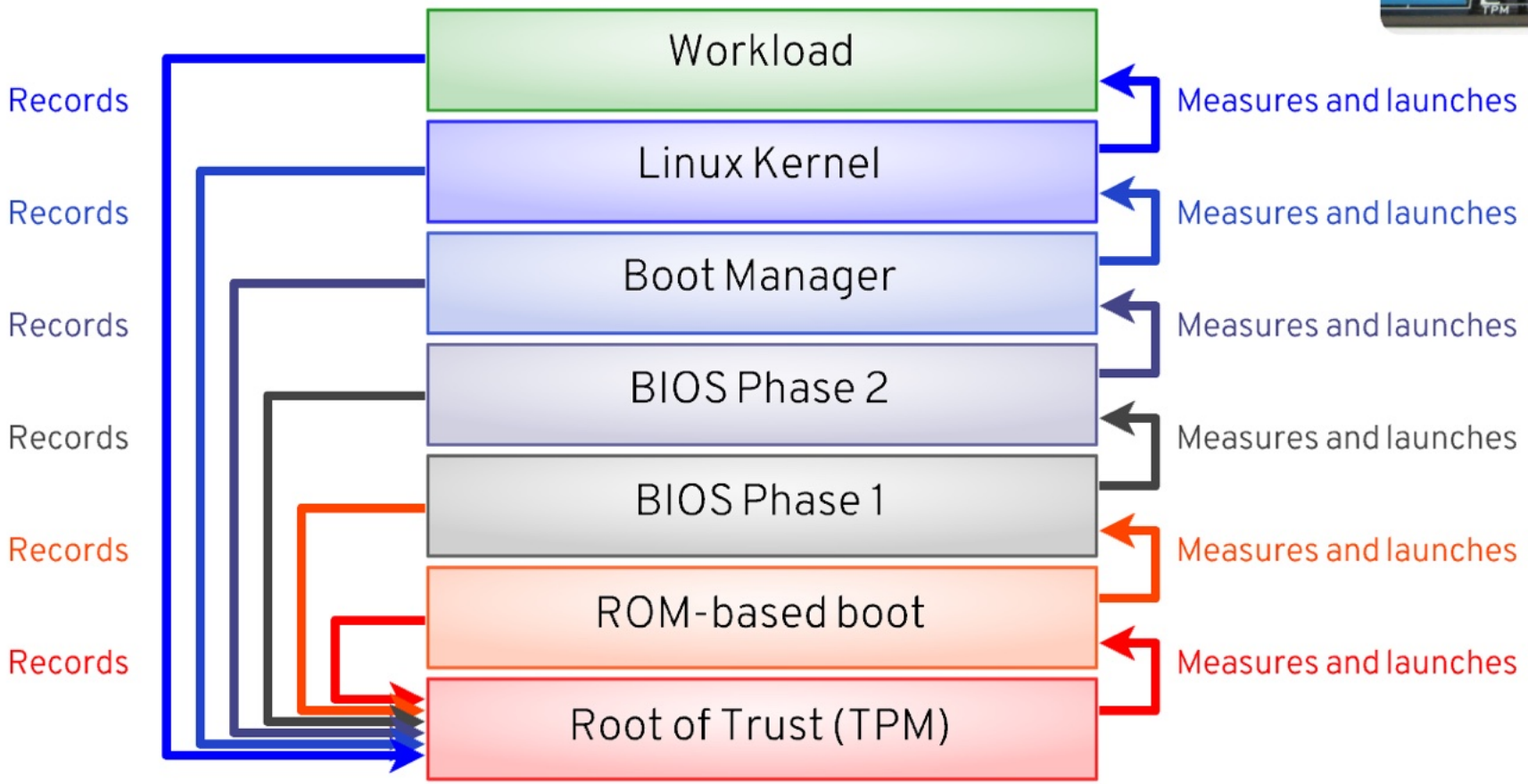
Why should infrastructure see your data?



Root of Trust

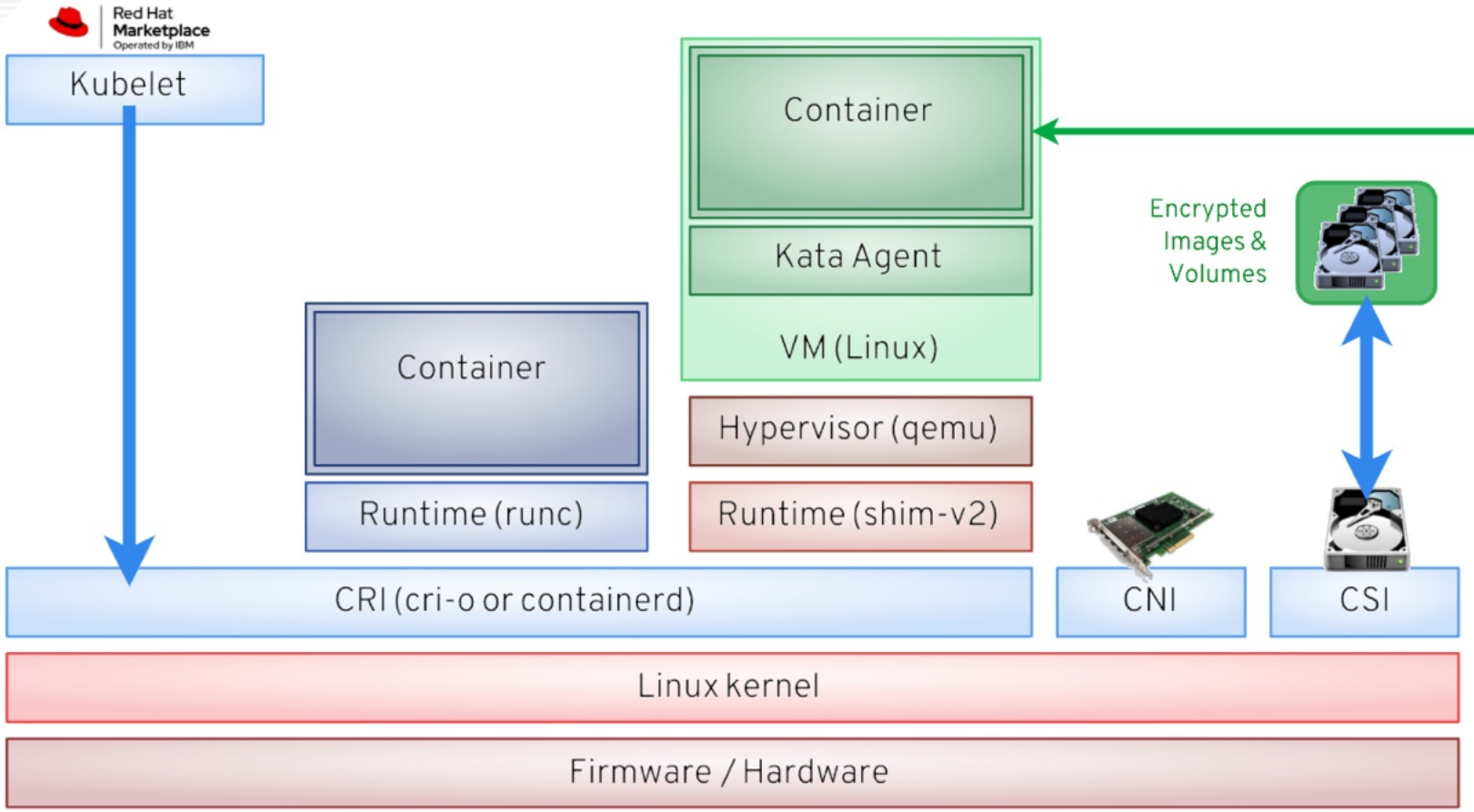
First there was the hardware





Trust domains

Example: confidential containers



Trusted platform:
Offers confidentiality guarantees enforced by hardware cryptography

Host:
Manages and offers resources used to run containers (CPU, memory, I/O, etc)

Tenant:
Confidential area, inaccessible to the host even when running on it

Guarantees

What does confidential computing really provide?

- Confidential computing is about... confidentiality

Guarantees

What does confidential computing really provide?

- Confidential computing is about... confidentiality
- Protect data in use from leaks or tampering

Guarantees

What does confidential computing really provide?

- Confidential computing is about... confidentiality
- Protect data in use from leaks or tampering

- Does not protect against crashes

Guarantees

What does confidential computing really provide?

- Confidential computing is about... confidentiality
- Protect data in use from leaks or tampering

- Does not protect against crashes
- Does not protect disk, network data, ...

Guarantees

What does confidential computing really provide?

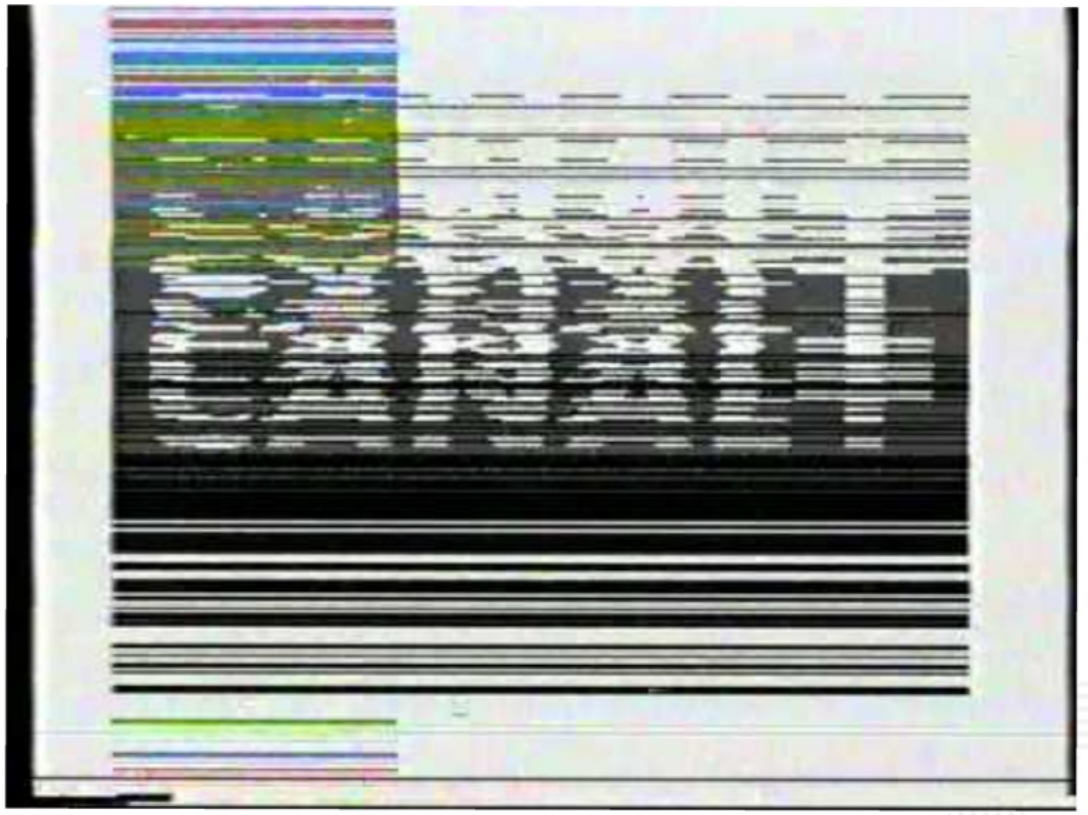
- Confidential computing is about... confidentiality
- Protect data in use from leaks or tampering

- Does not protect against crashes
- Does not protect disk, network data, ...
- Does not offer any guarantee of service

Guarantees

What does confidential computing really provide?

- Confidential computing is about... confidentiality
- Protect data in use from leaks or tampering



- Does not protect against crashes
- Does not protect disk, network data, ...
- Does not offer any guarantee of service
- Hardware-based, real-time cryptography

Guarantees

What does confidential computing really provide?

- Confidential computing is about... confidentiality
- Protect data in use from leaks or tampering

- Does not protect against crashes
- Does not protect disk, network data, ...
- Does not offer any guarantee of service
- Hardware-based, real-time cryptography
- Is highly implementation-dependent

✓ = Mitigated ★ = Optionally Mitigated ⊘ = Not Mitigated

Potential Threats	SEV	SEV-ES	SEV-SNP
Confidentiality			
VM Memory *Example attack: Hypervisor reads private VM memory*	✓	✓	✓
VM Register State *Example attack: Read VM register state after VMEXIT*	⊘	✓	✓
DMA Protection *Example attack: Device attempts to read VM memory*	✓	✓	✓
Integrity			
Replay Protection *Example attack: Replace VM memory with an old copy*	⊘	⊘	✓
Data Corruption *Example attack: Replace VM memory with junk data*	⊘	⊘	✓
Memory Aliasing *Example attack: Map two guest pages to same DRAM page*	⊘	⊘	✓
Memory Re-Mapping *Example attack: Switch DRAM page mapped to a guest page*	⊘	⊘	✓
Availability			
Denial of Service on Hypervisor *Example attack: Malicious guest refuses to yield/exit*	✓	✓	✓
Denial of Service on Guest *Example attack: Malicious hypervisor refuses to run guest*	⊘	⊘	⊘
Physical Access Attacks			
Offline DRAM analysis *Example attack: Cold boot*	✓	✓	✓
Active DRAM corruption *Example attack: Manipulate DDR bus while VM is running*	⊘	⊘	⊘
Misc.			
TCB Rollback *Example attack: Revert AMD-SP firmware to old version*	⊘	⊘	✓
Malicious Interrupt/Exception Injection *Example attack: Inject interrupt while RFLAGS.IF=0*	⊘	⊘	★
Indirect Branch Predictor Poisoning *Example attack: Poison BTB from hypervisor*	⊘	⊘	★
Secure Hardware Debug Registers *Example attack: Change breakpoints during debug*	⊘	⊘	★
Trusted CPUID Information *Example attack: Hypervisors lies about platform capabilities*	⊘	⊘	★
Architectural Side Channels *Example attack: PRIME+PROBE to track VM accesses*	⊘	⊘	⊘
Page-level Side Channels *Example attack: Track VM access patterns through page tables*	⊘	⊘	⊘
Performance Counter Tracking *Example attack: Fingerprint VM apps by performance data*	⊘	⊘	⊘

Guarantees

What does confidential computing really provide?

- Confidential computing is about... confidentiality
- Protect data in use from leaks or tampering

- Does not protect against crashes
- Does not protect disk, network data, ...
- Does not offer any guarantee of service
- Hardware-based, real-time cryptography
- Is highly implementation-dependent

- TL;DR: There is no automatic security

KVM FORUM

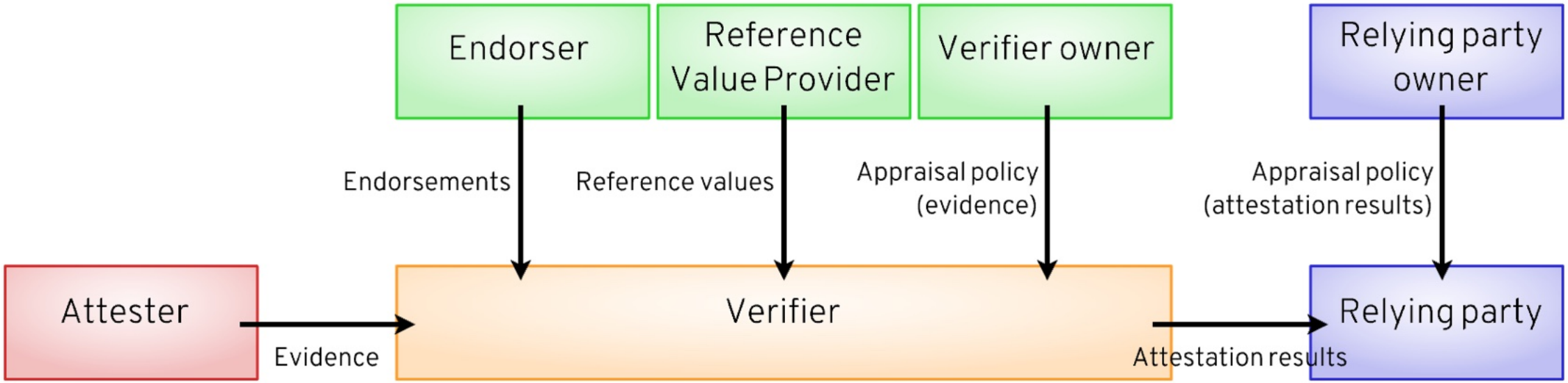
What is Attestation?

Proving that you run
what you want to run
where you want to run it

Red Hat

A little bit of terminology

The RATS model (from IETF)



Attestation: Basic concepts

Offering proofs about the configuration of a system

- In general, attestation proves a property of a system

Attestation: Basic concepts

Offering proofs about the configuration of a system

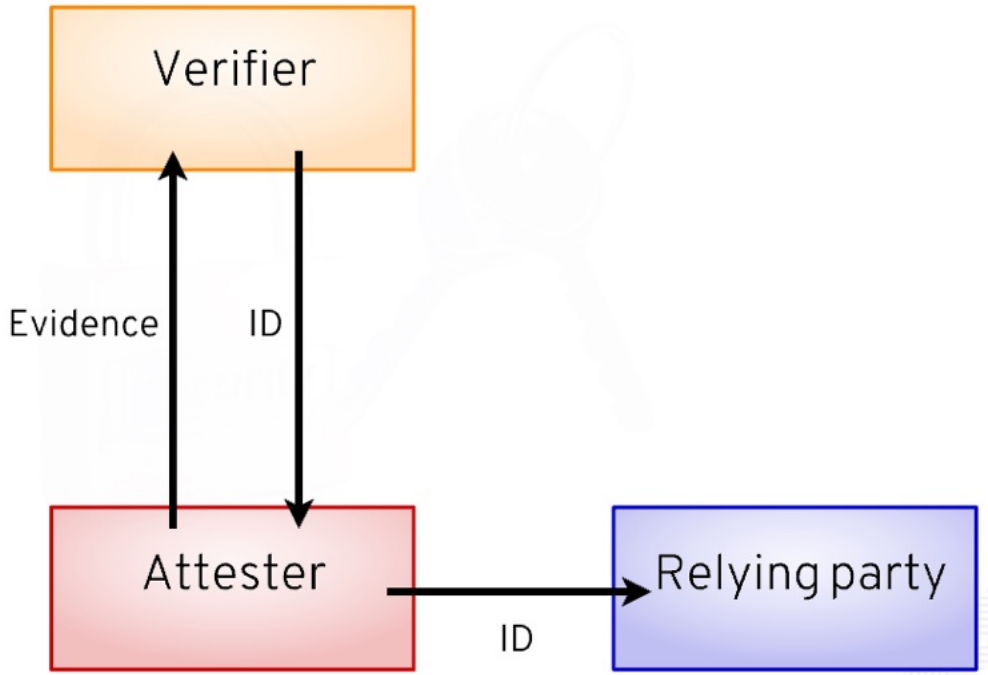
- In general, attestation proves a property of a system
- Remote attestation decouples evidence from verification

Attestation: Basic concepts

Offering proofs about the configuration of a system

- In general, attestation proves a property of a system
- Remote attestation decouples evidence from verification

- Passport check model: present evidence

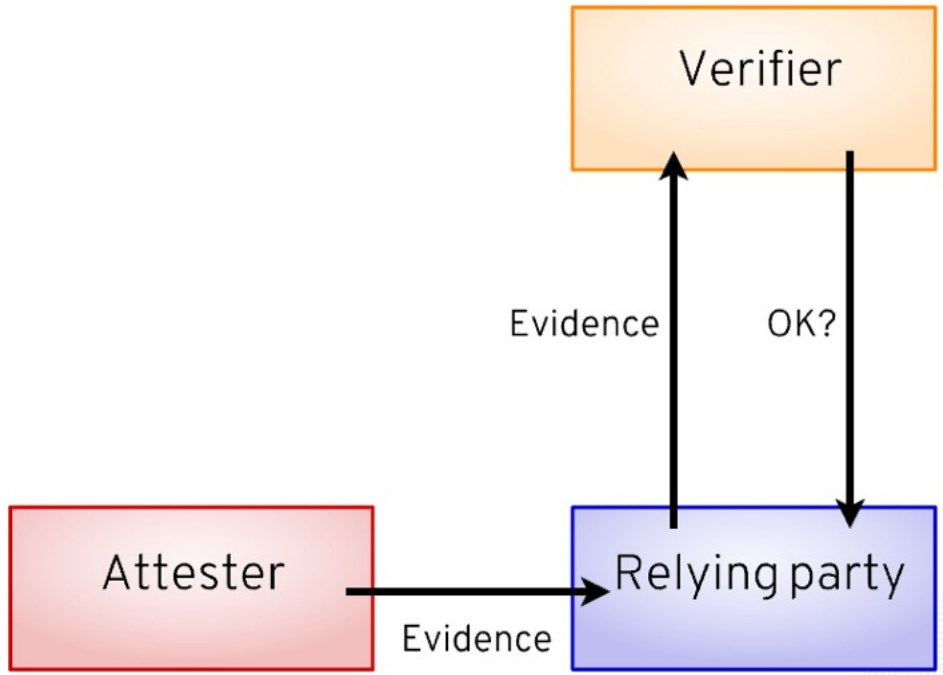


Attestation: Basic concepts

Offering proofs about the configuration of a system

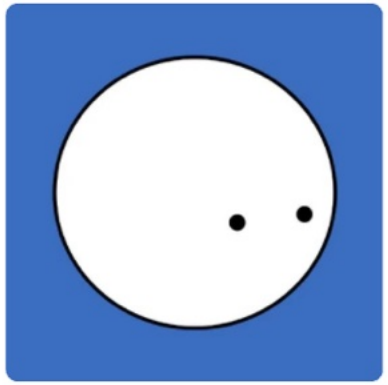
- In general, attestation proves a property of a system
- Remote attestation decouples evidence from verification

- Passport check model: present evidence
- Background check mode: validate evidence



REMITS pipeline

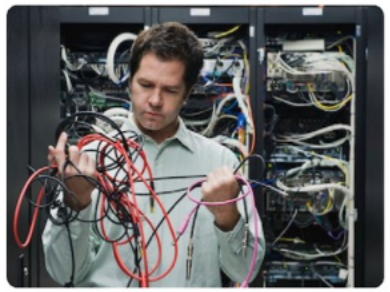
A simplified (simplistic?) model of trust chains



R E M I T S

REMITS pipeline examples

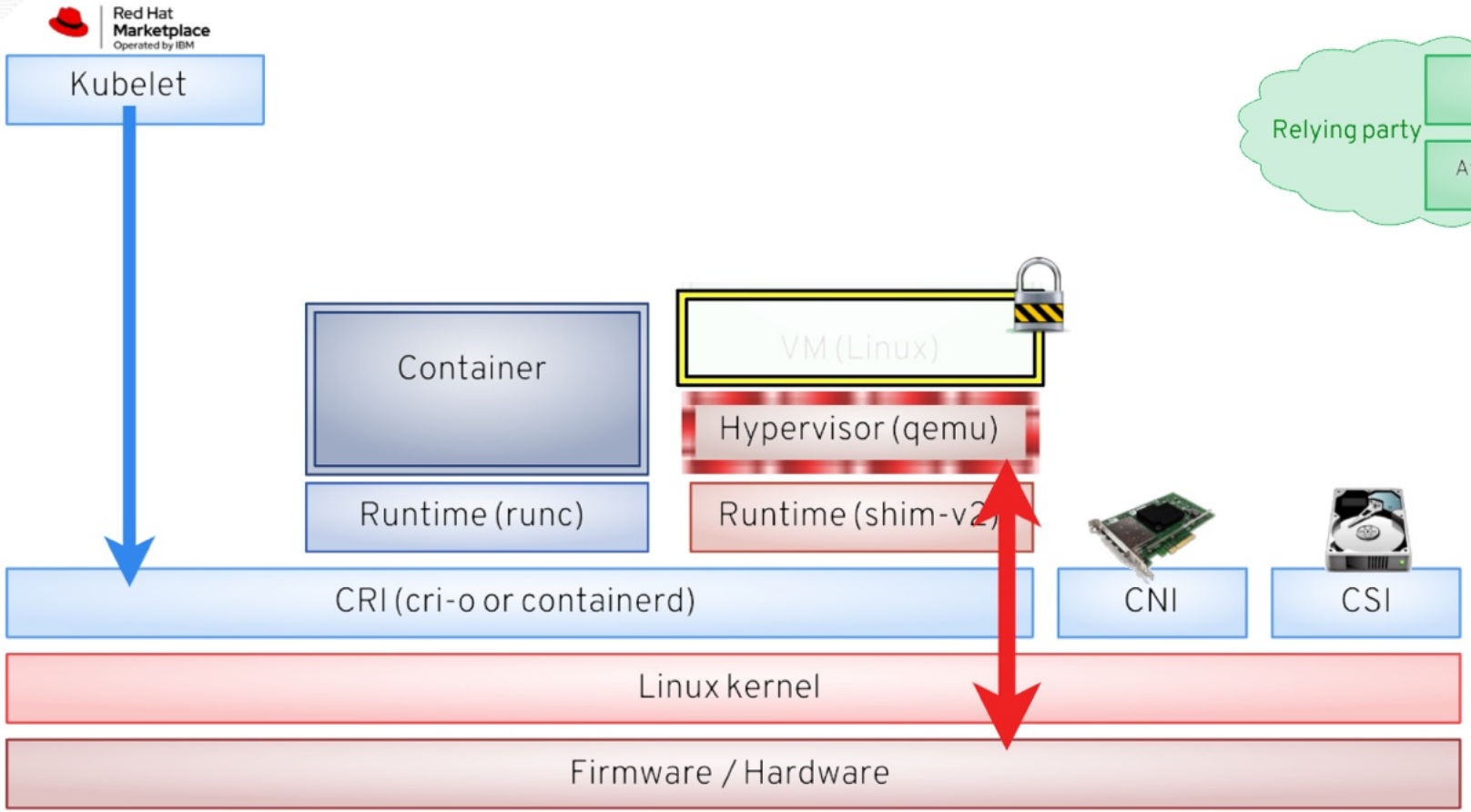
Some simple applications of the REMITS pipeline



Secure Boot	TPM	Manufacturer	Firmware, bootloader	Signed attestation	Attestation policies	Cloud API secret
Selling a property	Notary	Signed records	Deed or affidavit	Property description	I got your money	Handing the keys
Historical money	Gold or silver	Government	Market value	Number of dollars	Handing over cash	Getting food or goods

Attestation

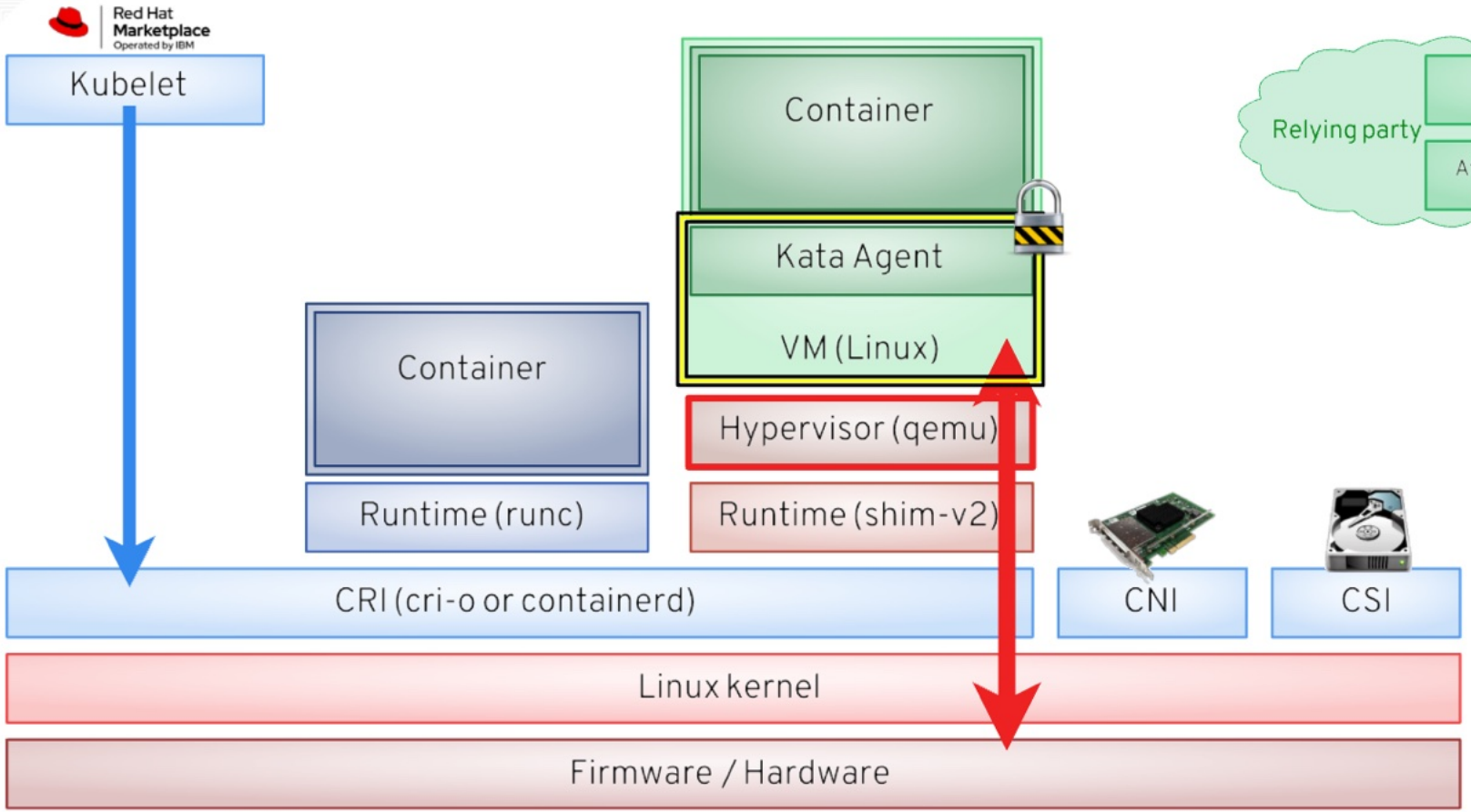
Measuring what we run using cryptography



Preattestation:
Measure the payload before allowing it to start (original SEV)

Attestation

Measuring what we run using cryptography

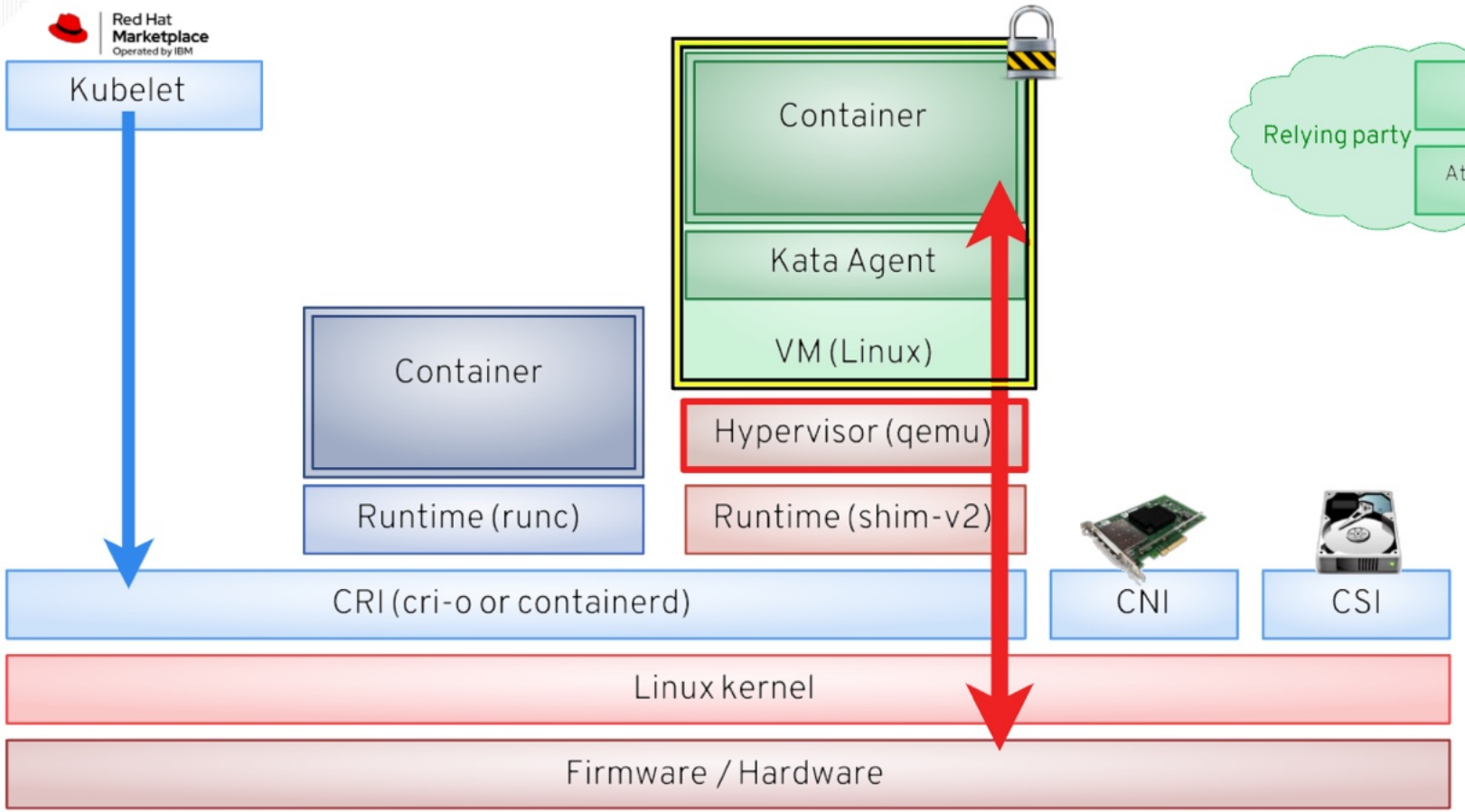


Preattestation:
Measure the payload before allowing it to start (original SEV)

Postattestation:
Code in the payload can confirm its identity using the measurement in order to get secrets

Attestation

Measuring what we run using cryptography



Preattestation:
Measure the payload before allowing it to start (original SEV)

Postattestation:
Code in the payload can confirm its identity using the measurement in order to get secrets

Workload attestation
The workload itself is attested, e.g. gets secrets from attestation

KVM FORUM

Use Cases

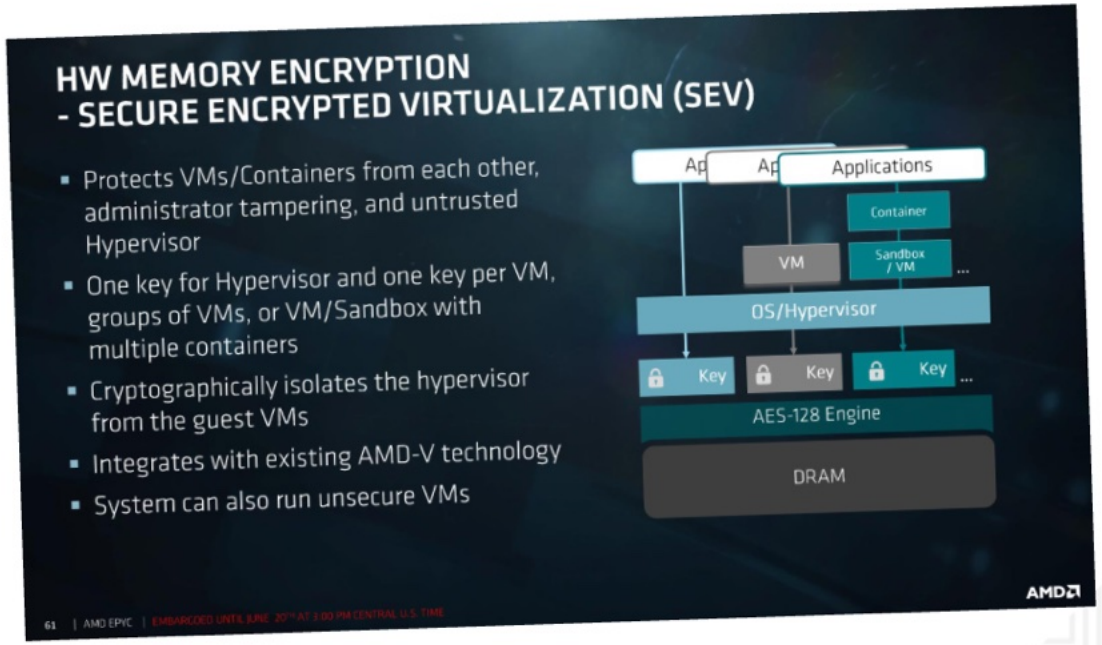
Various ways to deploy confidential computing

Red Hat

Use Cases

From virtual machine to whole clusters

- Base: Confidential Virtual Machines



Use Cases

From virtual machine to whole clusters

- Base: Confidential Virtual Machines
- Functions: Confidential Workloads (krunvm)

krunvm

`krunvm` is a CLI-based utility for creating microVMs from OCI images, using libkrun and buildah.

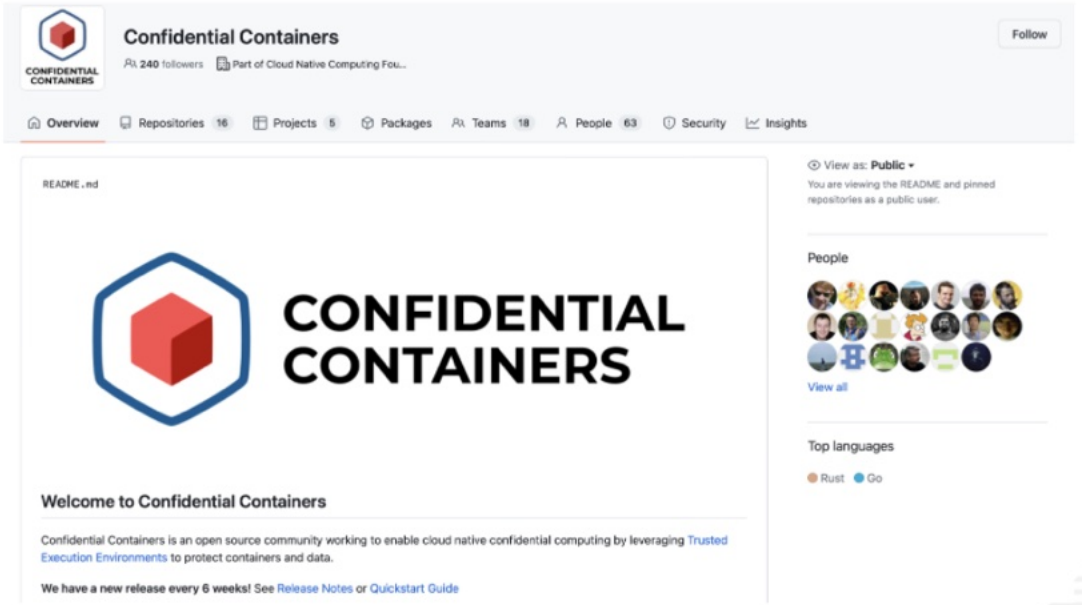
Features

- Minimal footprint
- Fast boot time
- Zero disk image maintenance
- Zero network configuration
- Support for mapping host volumes into the guest
- Support for exposing guest ports to the host

Use Cases

From virtual machine to whole clusters

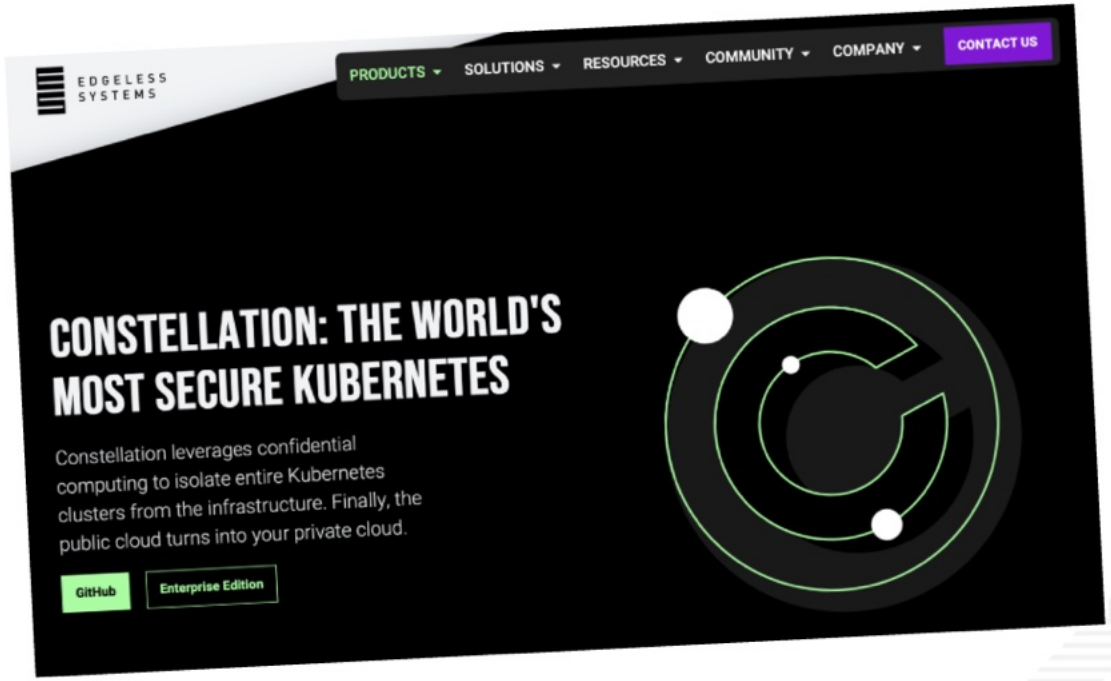
- Base: Confidential Virtual Machines
- Functions: Confidential Workloads (krunvm)
- Orchestrated: Confidential Containers



Use Cases

From virtual machine to whole clusters

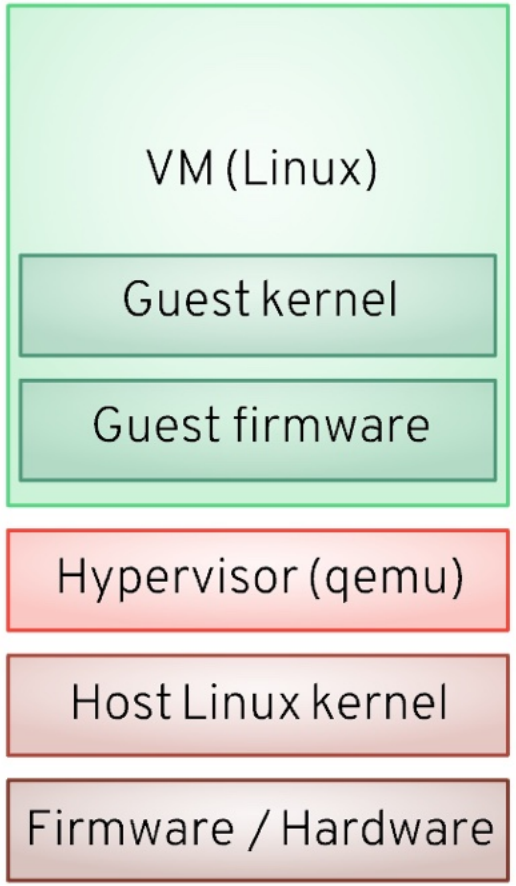
- Base: Confidential Virtual Machines
- Functions: Confidential Workloads (krunvm)
- Orchestrated: Confidential Containers
- Enchilada: Confidential Clusters



Confidential Virtual Machines

The basic technology behind it all

- New hardware / firmware ABI with new features
- Host kernel no longer trusted, exposes new devices
- Hypervisor no longer trusted, exposes new features
- VM becomes a confidential enclave
- Guest firmware and boot sequence is measured
- Guest kernel can be measured



Confidential Workloads

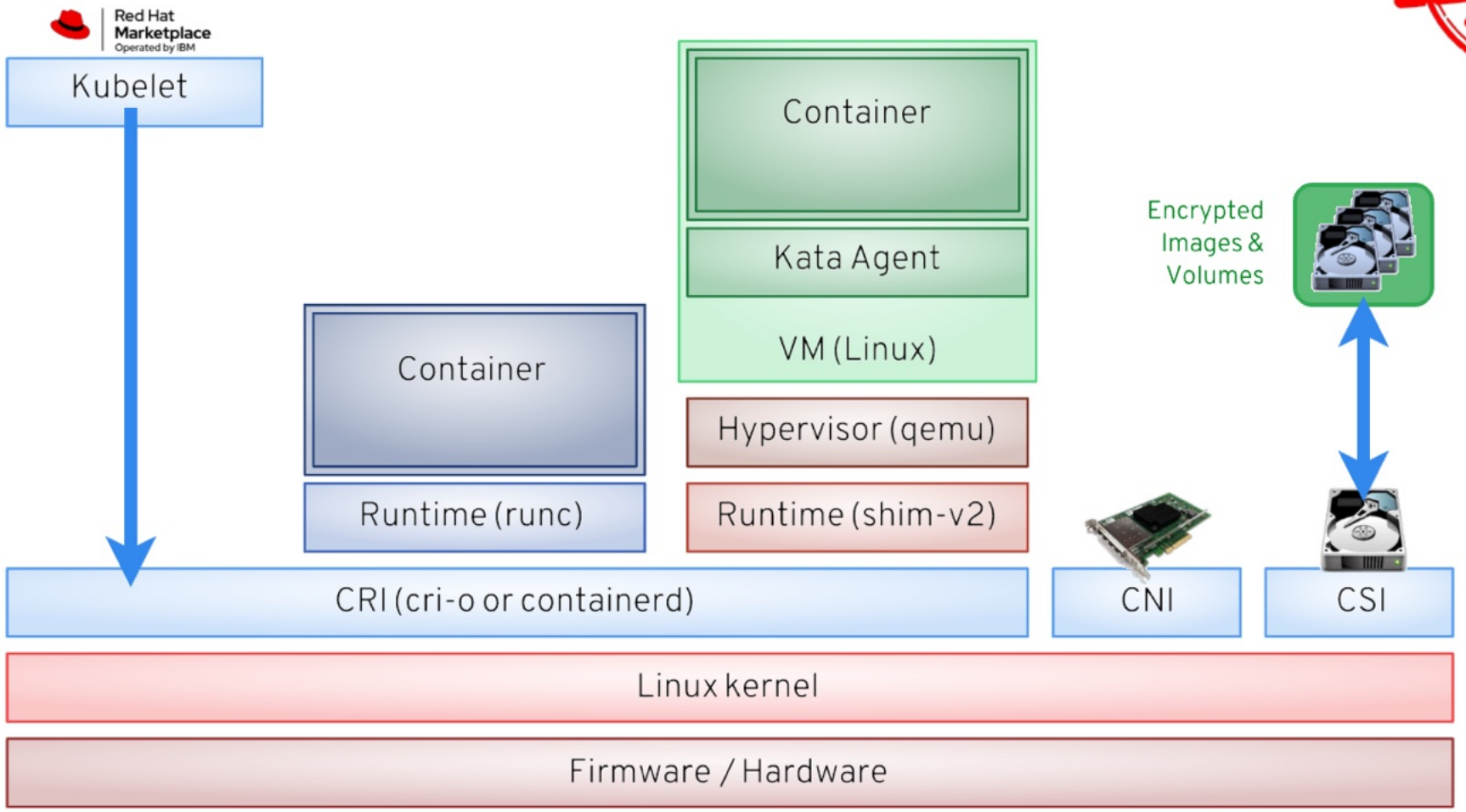
Lightweight, quick, container-like

- VM as a library (libkrun)
- Direct integration with Podman
- Got very early support for SEV
- First working attestation

```
~> uname -a                                                                    14:08:08ptitbras
Darwin ptitbras 22.5.0 Darwin Kernel Version 22.5.0: Mon Apr 24 20:52:24 PDT 2023; root:xnu-8796.121.2~5/RELEASE_ARM64_T6000 ar
m64
~> ls                                                                          14:08:20ptitbras
Applications Downloads    Hatari       Maildir      Pictures     Work         k8s
Desktop      Dropbox      Library      Movies       Public       bin
Documents    Google Drive Machine      Music        Sites        go
~> krunvm start ubuntu                                                         14:08:25ptitbras
root@ubuntu:/ddd# ls
 Applications   Documents   Dropbox          Hatari    Machine   Movies   Pictures   Sites   bin   k8s
 Desktop        Downloads   'Google Drive'   Library   Maildir   Music    Public     Work    go
root@ubuntu:/ddd# uname -a
Linux ubuntu 6.2.9 #1 SMP Mon Apr  3 04:28:59 PM CEST 2023 aarch64 aarch64 aarch64 GNU/Linux
root@ubuntu:/ddd#
```

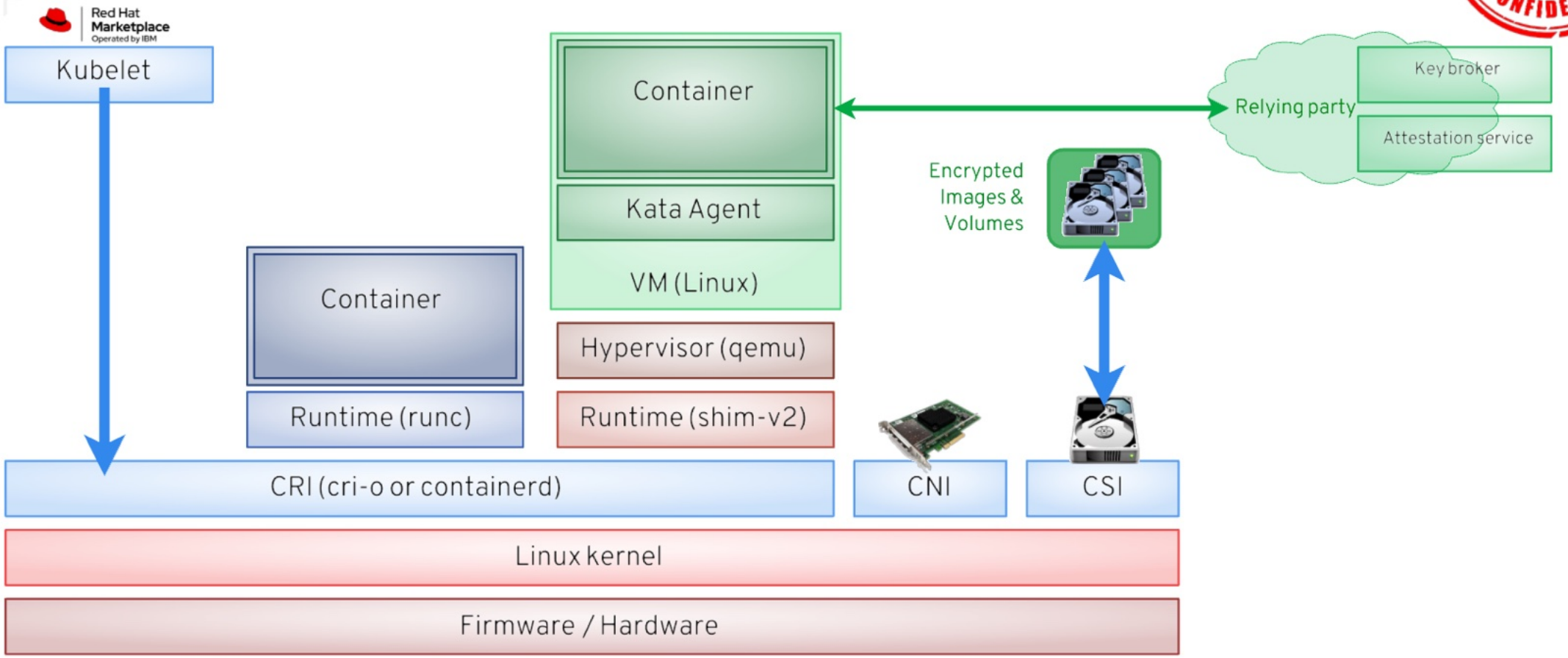
Confidential Containers with Kata

Using confidential VMs as a Kubernetes runtime



Confidential Containers with Kata

Using confidential VMs as a Kubernetes runtime



Confidential Clusters

Running an entire cluster inside confidential enclaves

- Make the whole cluster confidential
- Works at the cloud provider level
- Generates confidential nodes
- Attested TLS (ATLS)
- JoinService: Attest nodes
- VerificationService: User-facing

```
~# constellation iam create azure --region=westus --resourceGroup=c3d-constellation-resources --servicePrincipal=c3d-constellat
ion-service-principal --generate-config


The following IAM configuration will be created:

Region:                 westus
Resource Group:         c3d-constellation-resources
Service Principal:      c3d-constellation-service-principal

The configuration file constellation-conf.yaml will be automatically generated and populated with the IAM values.
Do you want to create the configuration? [y/n]: y
Creating

Your IAM configuration was created and filled into constellation-conf.yaml successfully.
~#                                                                                                        SEV Demo
```

KVM FORUM

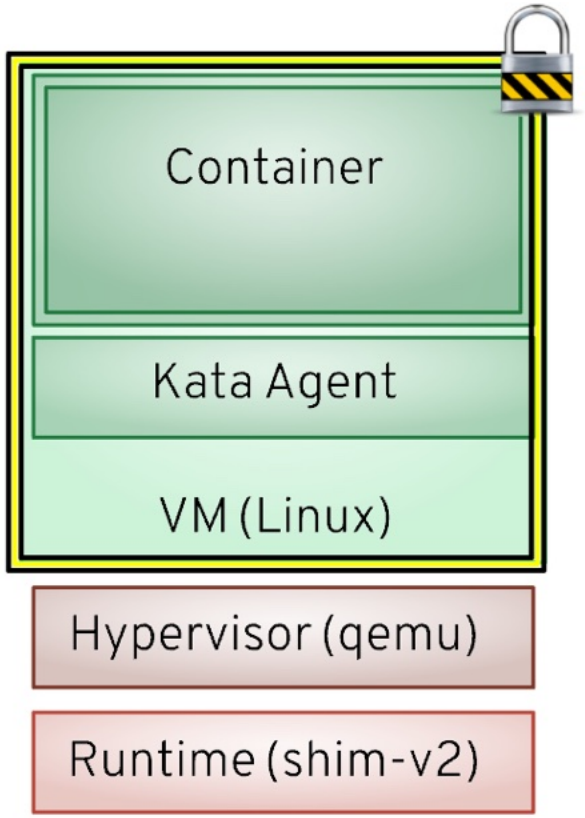
Building actual trust

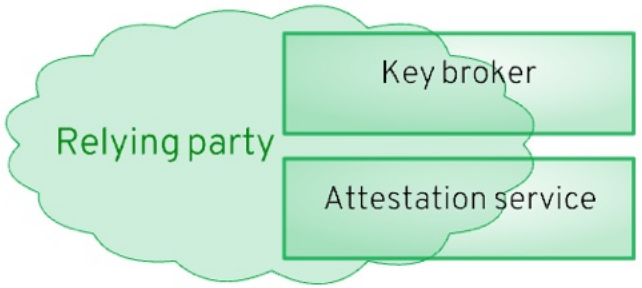
Keeping the trust alive along the way

Red Hat

How does attestation work?

Challenge / response to deliver secrets

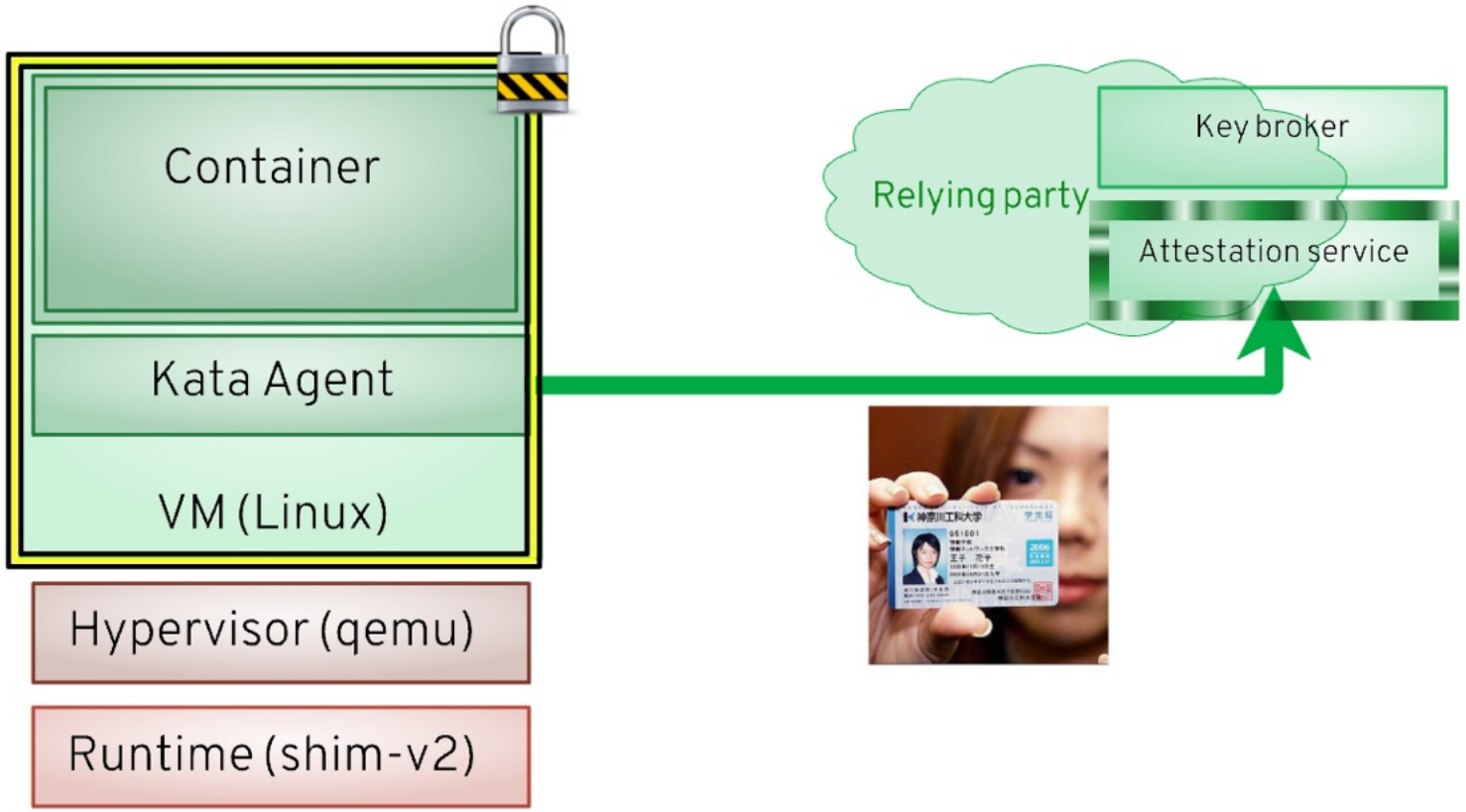




Cryptographic measurement:
Measurement of relevant memory performed by hardware / firmware

How does attestation work?

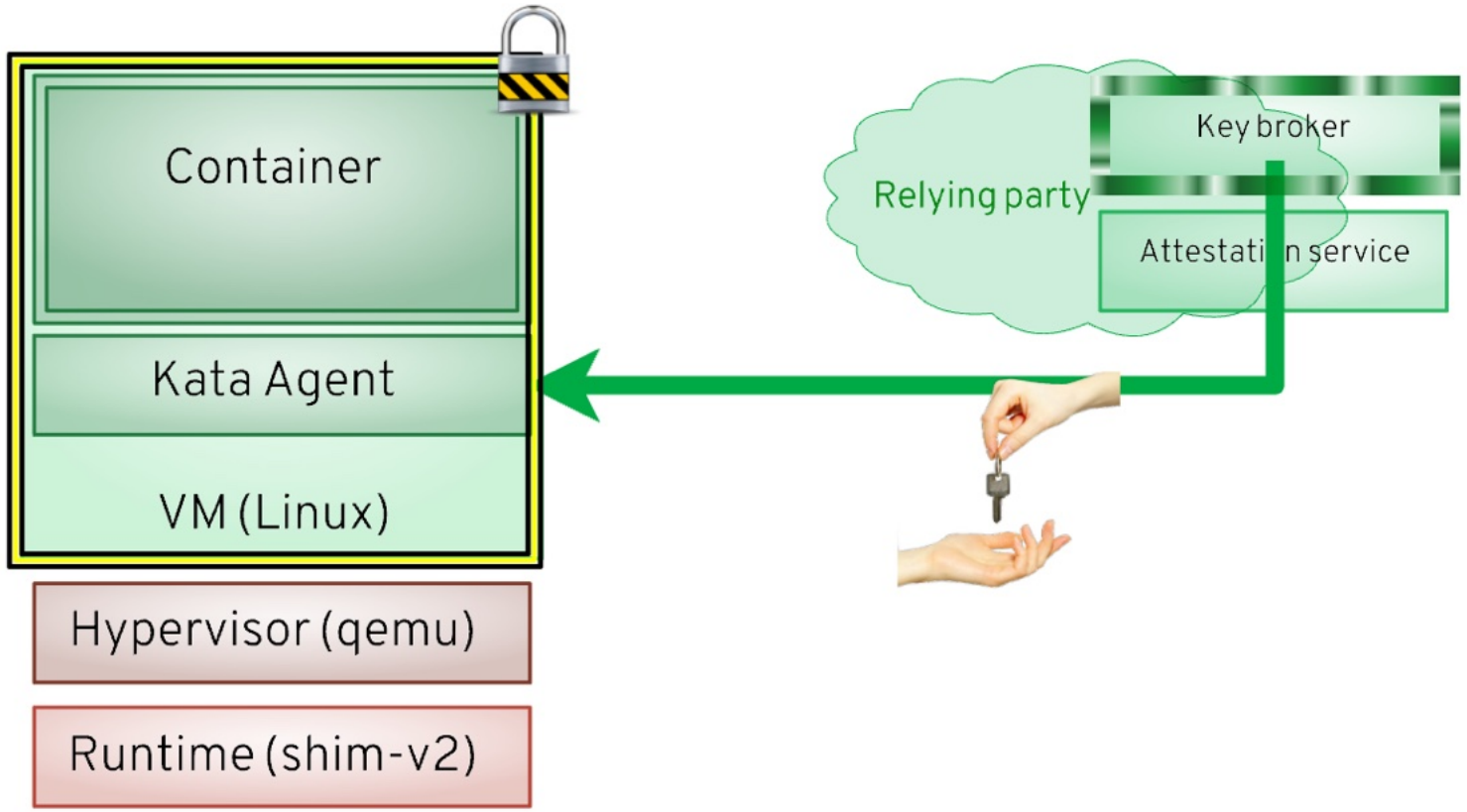
Challenge / response to deliver secrets



Cryptographic measurement:
Measurement of relevant memory performed by hardware / firmware
Cryptographic challenge:
Send proof of identity (salted)

How does attestation work?

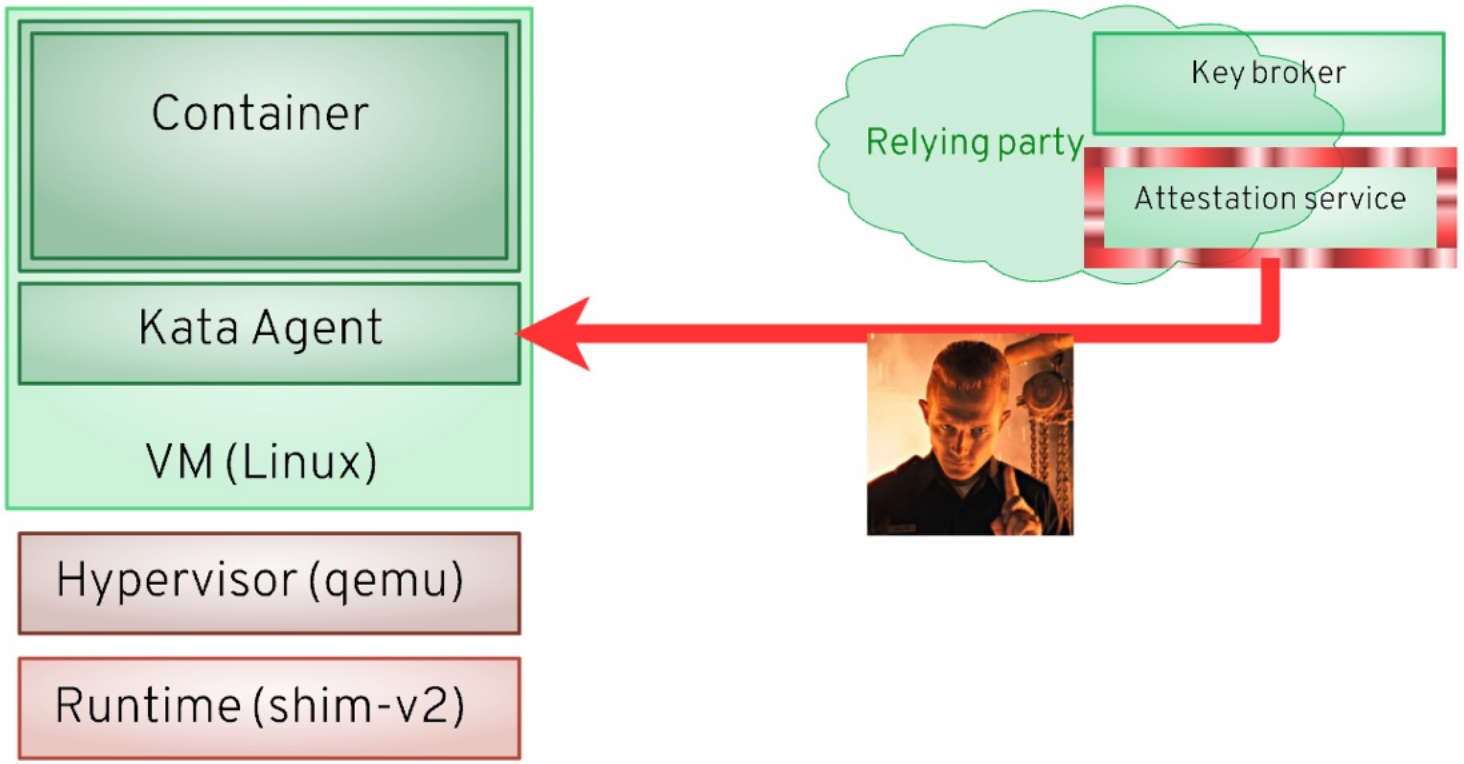
Challenge / response to deliver secrets



Cryptographic measurement:
Measurement of relevant memory performed by hardware / firmware
Cryptographic challenge:
Send proof of identity (salted)
Secret delivery
Ensure the workload cannot do harm if not attested

How does attestation work?

Challenge / response to deliver secrets

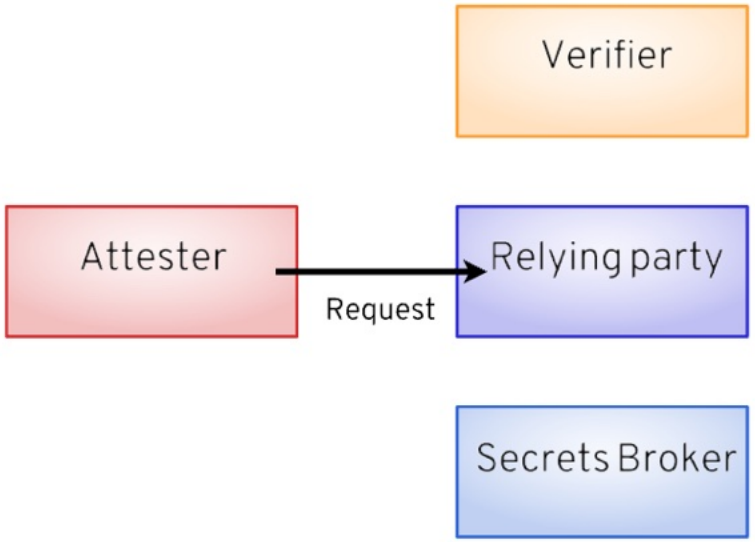


Cryptographic measurement:
Measurement of relevant memory performed by hardware / firmware
Cryptographic challenge:
Send proof of identity (salted)
Secret delivery
Ensure the workload cannot do harm if not attested
Remote attestation:
Can invalidate workloads e.g. if compromized

Attestation flow

Unlock workloads by giving secrets

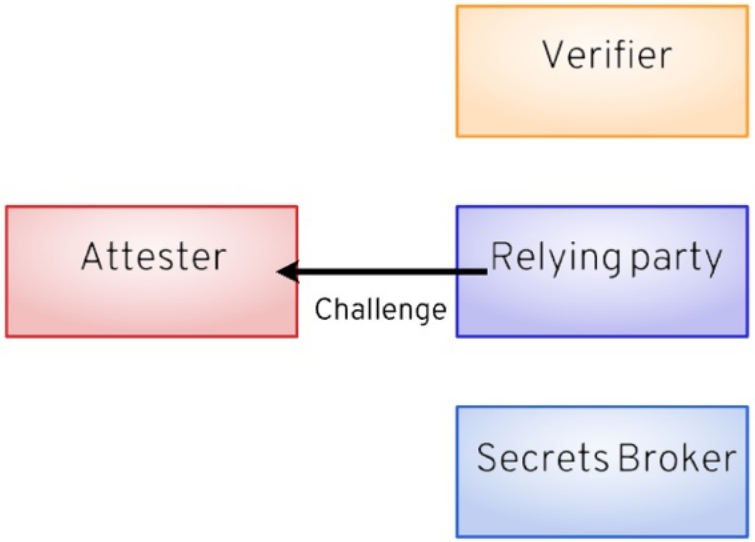
- Attester sends request



Attestation flow

Unlock workloads by giving secrets

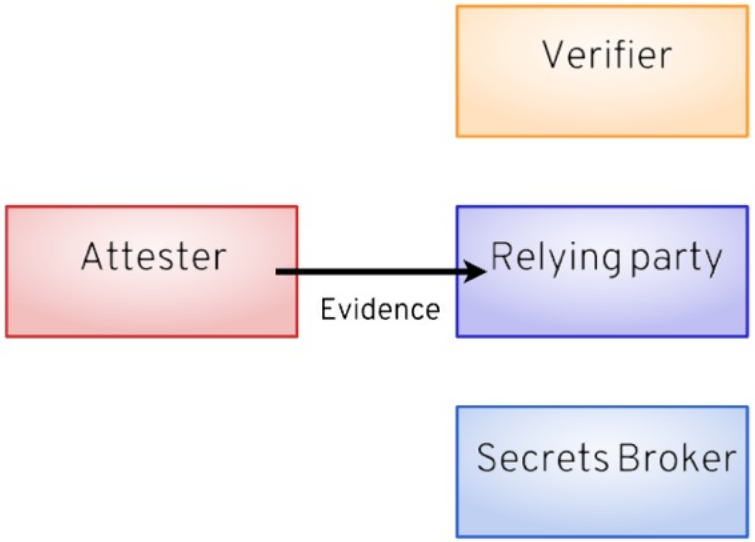
- Attester sends request
- Response is a cryptographic challenge



Attestation flow

Unlock workloads by giving secrets

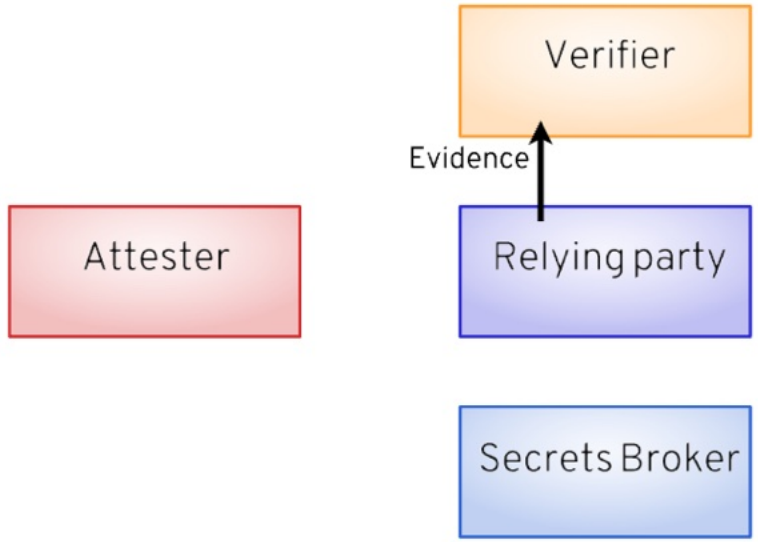
- Attester sends request
- Response is a cryptographic challenge
- Attester presents crypted evidence



Attestation flow

Unlock workloads by giving secrets

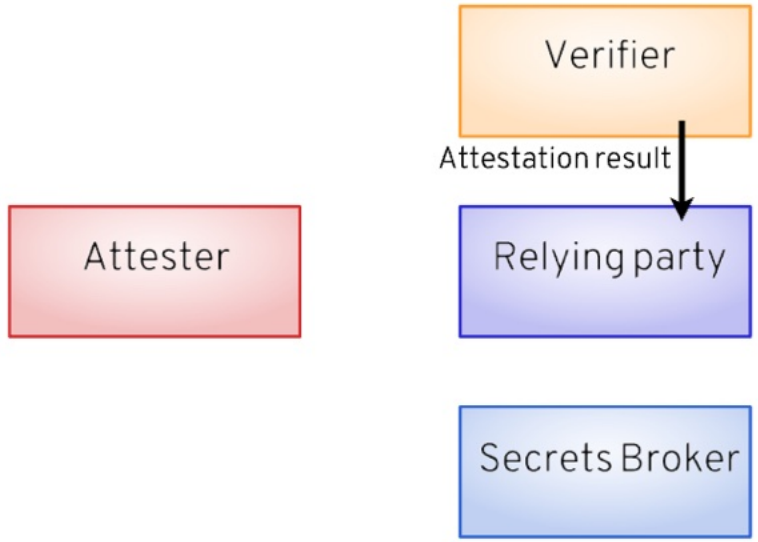
- Attester sends request
- Response is a cryptographic challenge
- Attester presents crypted evidence
- Evidence relayed to verifier



Attestation flow

Unlock workloads by giving secrets

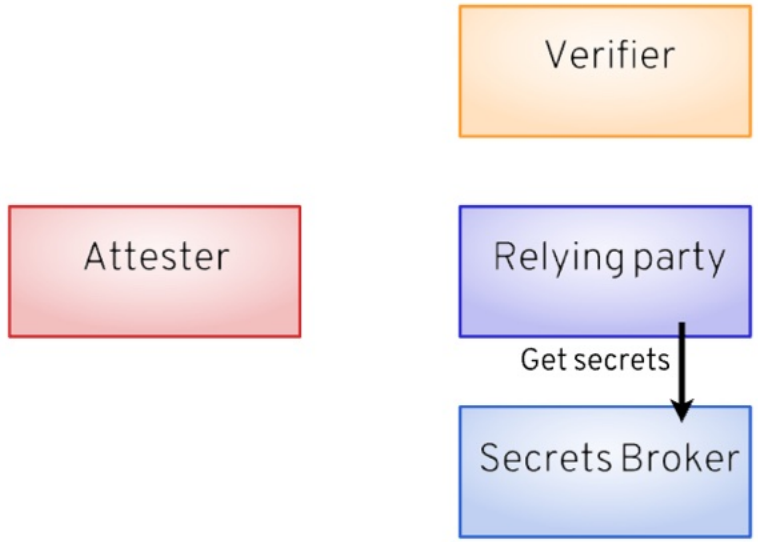
- Attester sends request
- Response is a cryptographic challenge
- Attester presents crypted evidence
- Evidence relayed to verifier
- Attestation result returned



Attestation flow

Unlock workloads by giving secrets

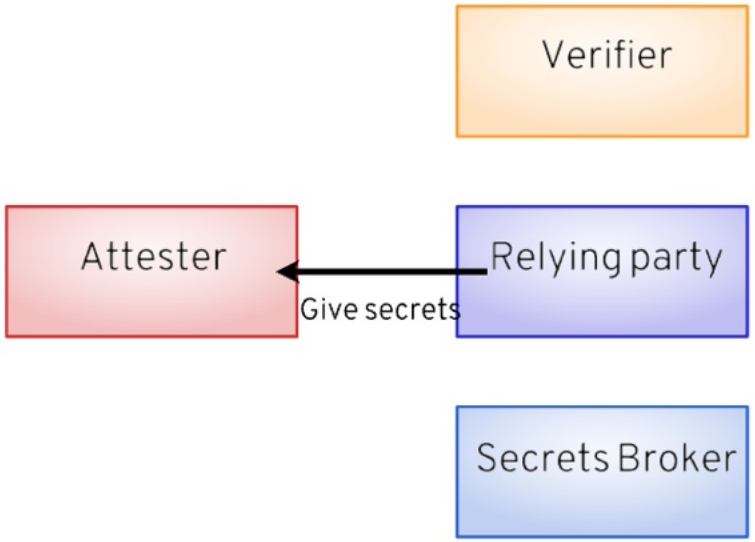
- Attester sends request
- Response is a cryptographic challenge
- Attester presents crypted evidence
- Evidence relayed to verifier
- Attestation result returned
- Secrets retrieved



Attestation flow

Unlock workloads by giving secrets

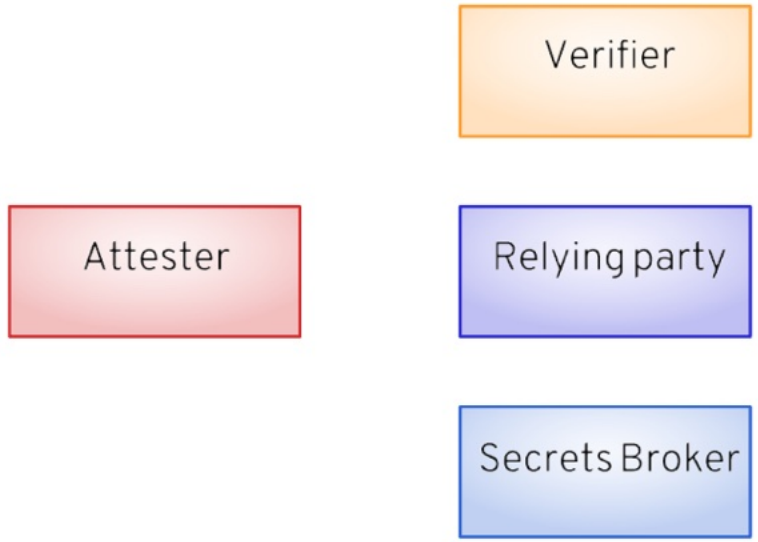
- Attester sends request
- Response is a cryptographic challenge
- Attester presents crypted evidence
- Evidence relayed to verifier
- Attestation result returned
- Secrets retrieved
- Response sent to attester



Attestation flow

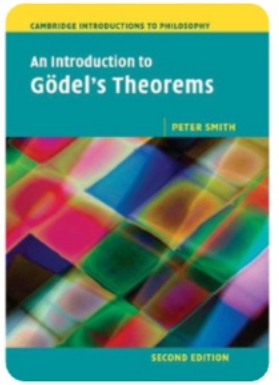
Unlock workloads by giving secrets

- Attester sends request
- Response is a cryptographic challenge
- Attester presents crypted evidence
- Evidence relayed to verifier
- Attestation result returned
- Secrets retrieved
- Response sent to attester



Who proves what to whom?

Different kinds of proof for different consumers



- System-facing: System software building a trusted execution environment
- User-facing: User checking if a system is trusted
- Workload-facing: Workload checking if runtime environment is trusted
- Peer-facing: Workloads checking of other workload is trusted
- Cluster-facing: nodes in a cluster check each other

KVM FORUM

Platform-specific details

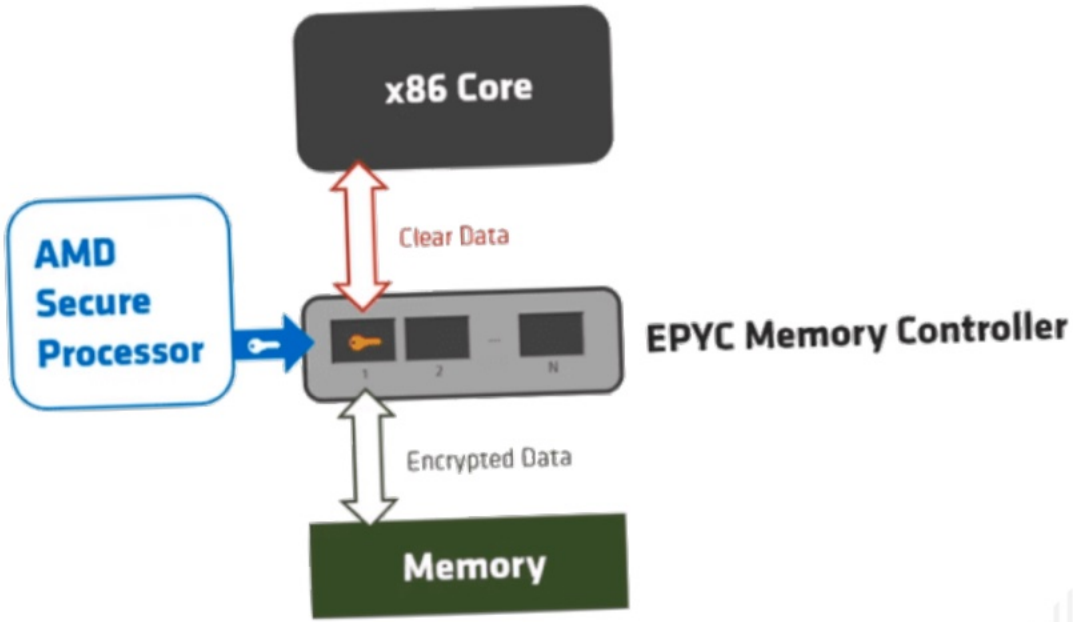
Beyond that point, there be zombies

Red Hat

Vendor landscape

Different vendors with different approaches?

- AMD: Secure Encrypted Virtualization (SEV)



Vendor landscape

Different vendors with different approaches?

- AMD: Secure Encrypted Virtualization (SEV)
  - SEV-ES adds Encrypted State (e.g. CPU register file)

Vendor landscape

Different vendors with different approaches?

- AMD: Secure Encrypted Virtualization (SEV)
  - SEV-ES adds Encrypted State (e.g. CPU register file)
  - SEV-SNP adds Secure Nested Pages (integrity protection)

✓ = Mitigated ★ = Optionally Mitigated ⊘ = Not Mitigated	SEV	SEV-ES	SEV-SNP
Potential Threats			
Confidentiality			
VM Memory *Example attack: Hypervisor reads private VM memory*	✓	✓	✓
VM Register State *Example attack: Read VM register state after VMEXIT*	⊘	✓	✓
DMA Protection *Example attack: Device attempts to read VM memory*	✓	✓	✓
Integrity			
Replay Protection *Example attack: Replace VM memory with an old copy*	⊘	⊘	✓
Data Corruption *Example attack: Replace VM memory with junk data*	⊘	⊘	✓
Memory Aliasing *Example attack: Map two guest pages to same DRAM page*	⊘	⊘	✓
Memory Re-Mapping *Example attack: Switch DRAM page mapped to a guest page*	⊘	⊘	✓
Availability			
Denial of Service on Hypervisor *Example attack: Malicious guest refuses to yield/exit*	✓	✓	✓
Denial of Service on Guest *Example attack: Malicious hypervisor refuses to run guest*	⊘	⊘	⊘
Physical Access Attacks			
Offline DRAM analysis *Example attack: Cold boot*	✓	✓	✓
Active DRAM corruption *Example attack: Manipulate DDR bus while VM is running*	⊘	⊘	⊘
Misc.			
TCB Rollback *Example attack: Revert AMD-SP firmware to old version*	⊘	⊘	✓
Malicious Interrupt/Exception Injection *Example attack: Inject interrupt while RFLAGS.IF=0*	⊘	⊘	★
Indirect Branch Predictor Poisoning *Example attack: Poison BTB from hypervisor*	⊘	⊘	★
Secure Hardware Debug Registers *Example attack: Change breakpoints during debug*	⊘	⊘	★
Trusted CPUID Information *Example attack: Hypervisors lies about platform capabilities*	⊘	⊘	★
Architectural Side Channels *Example attack: PRIME+PROBE to track VM accesses*	⊘	⊘	⊘
Page-level Side Channels *Example attack: Track VM access patterns through page tables*	⊘	⊘	⊘
Performance Counter Tracking *Example attack: Fingerprint VM apps by performance data*	⊘	⊘	⊘

Vendor landscape

Different vendors with different approaches?

- AMD: Secure Encrypted Virtualization (SEV)
  - SEV-ES adds Encrypted State (e.g. CPU register file)
  - SEV-SNP adds Secure Nested Pages (integrity protection)
- Intel: Trusted Domain Extensions (TDX)

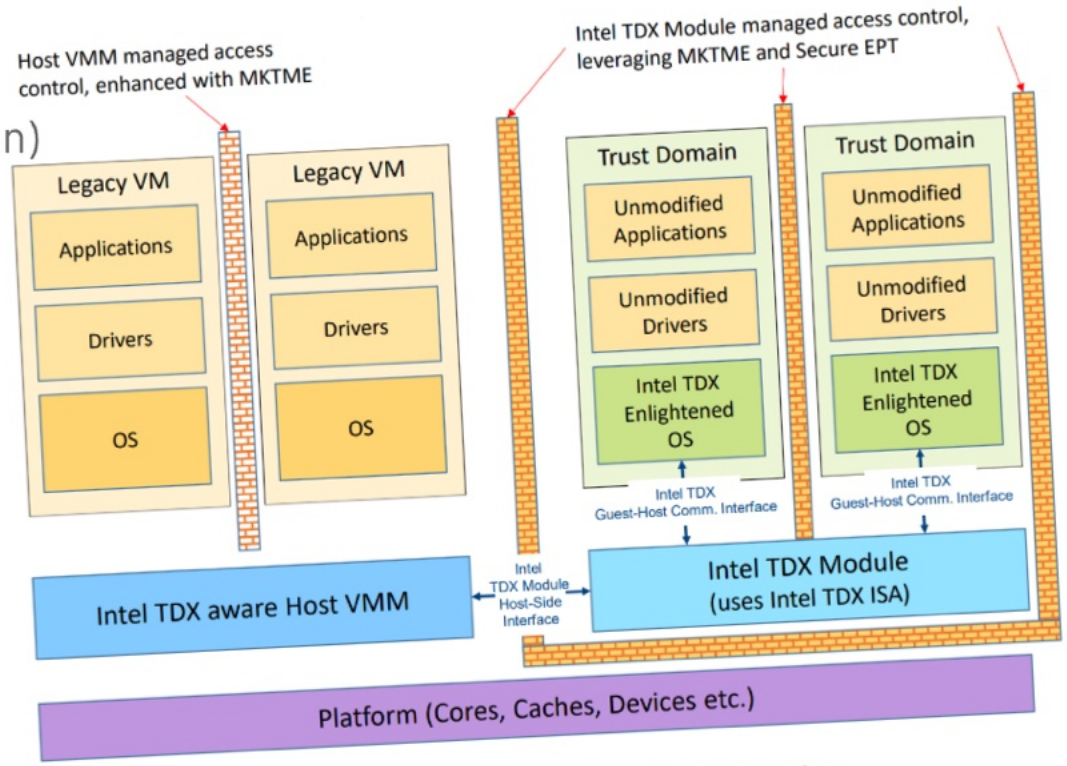
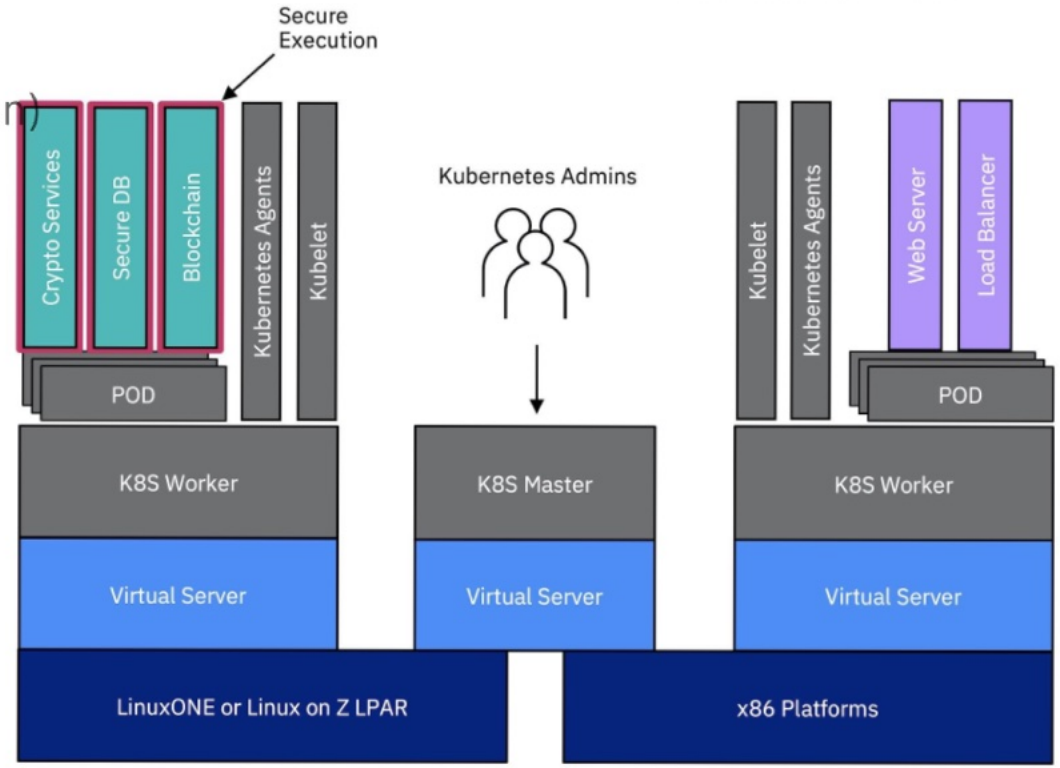


Figure 2.1: Components of Intel Trust Domain Extensions

Vendor landscape

Different vendors with different approaches?

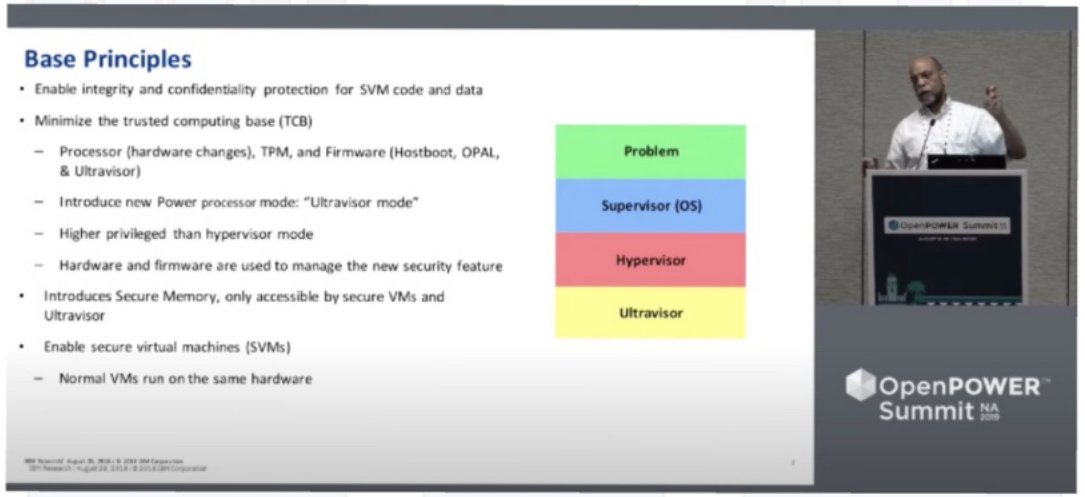
- AMD: Secure Encrypted Virtualization (SEV)
  - SEV-ES adds Encrypted State (e.g. CPU register file)
  - SEV-SNP adds Secure Nested Pages (integrity protection)
- Intel: Trusted Domain Extensions (TDX)
- IBM S390: Secure Execution (SE)



Vendor landscape

Different vendors with different approaches?

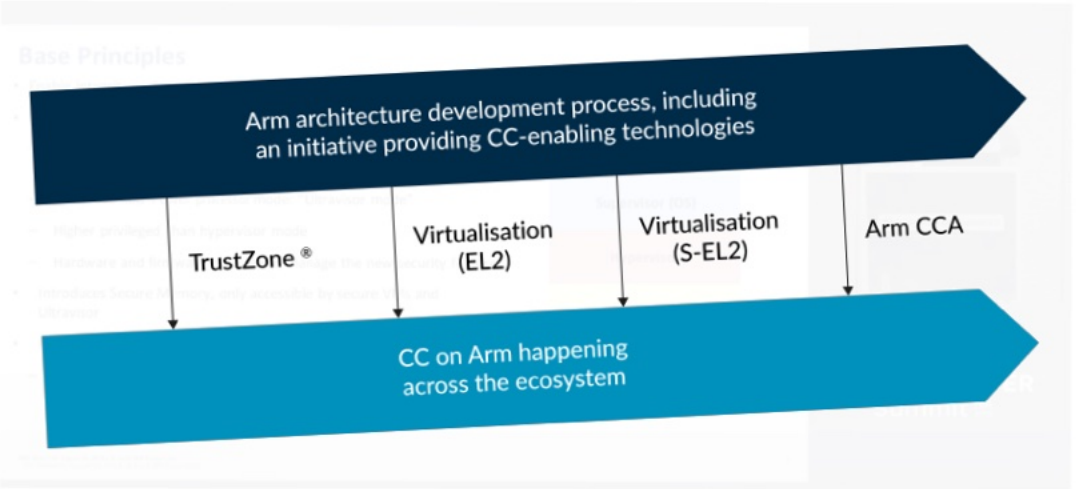
- AMD: Secure Encrypted Virtualization (SEV)
  - SEV-ES adds Encrypted State (e.g. CPU register file)
  - SEV-SNP adds Secure Nested Pages (integrity protection)
- Intel: Trusted Domain Extensions (TDX)
- IBM S390: Secure Execution (SE)
- Power: Protected Execution Facility (PEF)



Vendor landscape

Different vendors with different approaches?

- AMD: Secure Encrypted Virtualization (SEV)
  - SEV-ES adds Encrypted State (e.g. CPU register file)
  - SEV-SNP adds Secure Nested Pages (integrity protection)
- Intel: Trusted Domain Extensions (TDX)
- IBM S390: Secure Execution (SE)
- Power: Protected Execution Facility (PEF)
- Arm: Confidential Computing Architecture (CCA)



Vendor landscape

Different vendors with different approaches?

- AMD: Secure Encrypted Virtualization (SEV)
  - SEV-ES adds Encrypted State (e.g. CPU register file)
  - SEV-SNP adds Secure Nested Pages (integrity protection)
- Intel: Trusted Domain Extensions (TDX)
- IBM S390: Secure Execution (SE)
- Power: Protected Execution Facility (PEF)
- Arm: Confidential Computing Architecture (CCA)
- All these technologies are based on virtualization

Vendor landscape

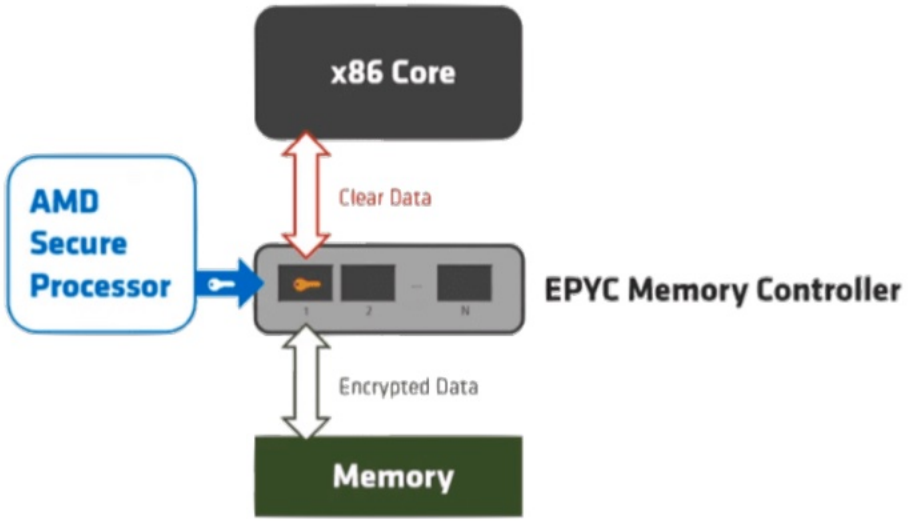
Different vendors with different approaches?

- AMD: Secure Encrypted Virtualization (SEV)
  - SEV-ES adds Encrypted State (e.g. CPU register file)
  - SEV-SNP adds Secure Nested Pages (integrity protection)
- Intel: Trusted Domain Extensions (TDX)
- IBM S390: Secure Execution (SE)
- Power: Protected Execution Facility (PEF)
- Arm: Confidential Computing Architecture (CCA)
- All these technologies are based on virtualization
- They all work differently

AMD SEV

Secure Encrypted Virtualization

- First generation technology, somewhat flawed
- Provides memory encryption through hardware
- Built on top of virtualization (unlike SME)
- Relies on a separate security processor
- Only features pre-attestation
- Several vulnerabilities gave it a bad reputation

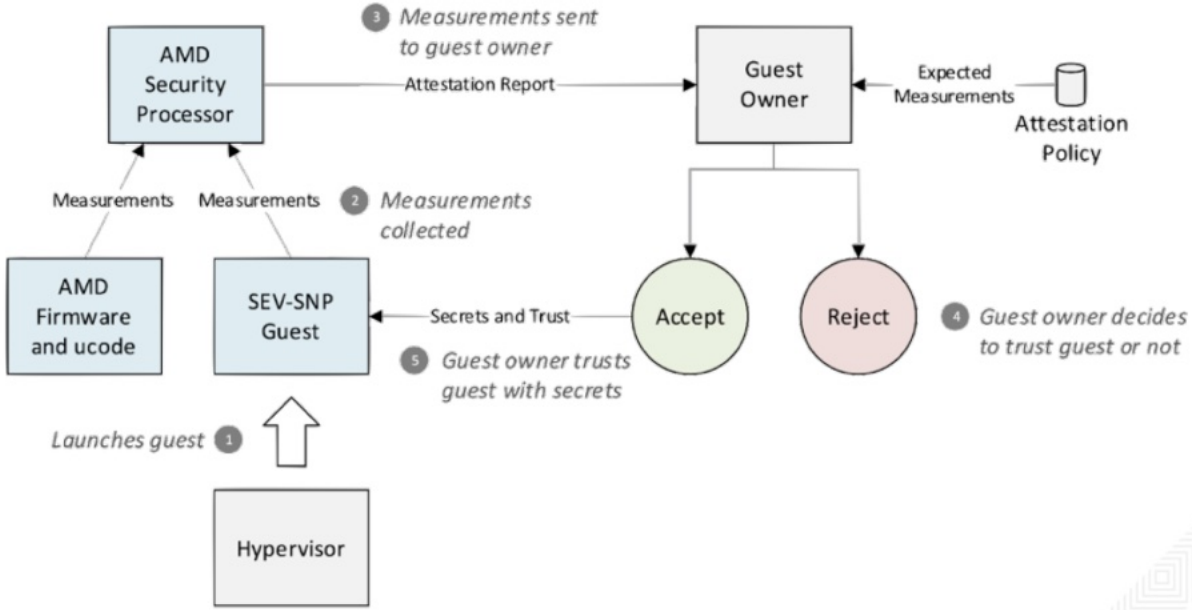


AMD SEV-ES and SEV-SNP

Encrypted State, Secure Nested Pages

- ES protects CPU state from tampering
- No major impact on the (pre-) attestation model

- SNP protects against malicious page mapping
- Can get attestation quote from within the guest
- VMPL gives additional protection levels for VMs
- VMPLs enable protected services, e.g. vTPM

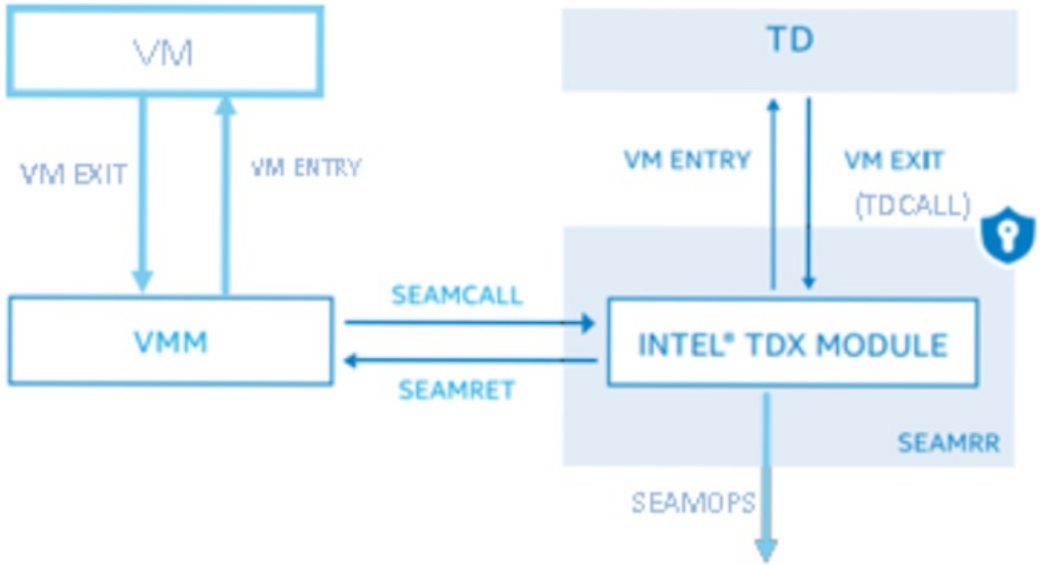


Intel TDX (and SGX)

Trust Domain Extensions (Software Guard Extensions)

- SGX is designed to create "Secure Enclaves"
- TDX is virtualization-based (like AMD-SEV)

- No separate security processor
- New CPU mode, Secure Arbitration Mode (SEAM)
- Various binary modules expose required services

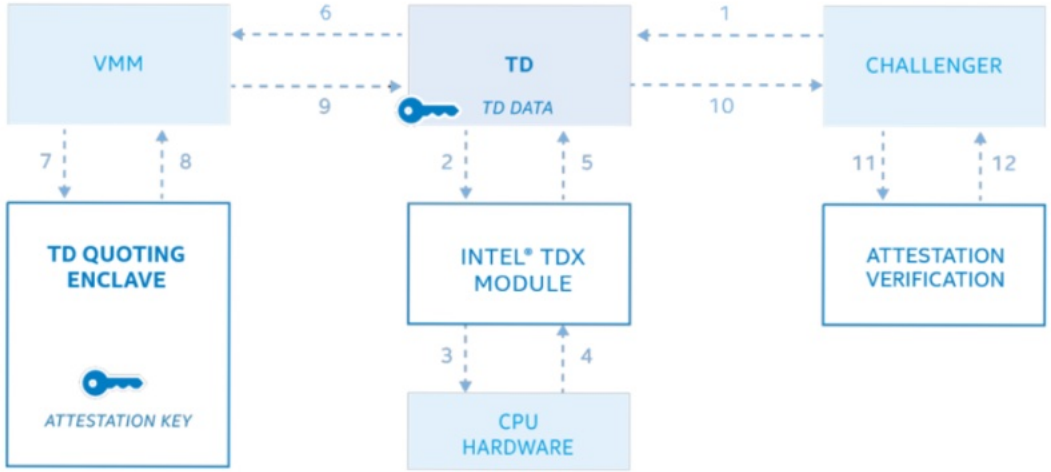


Intel TDX (and SGX)

Trust Domain Extensions (Software Guard Extensions)

- SGX is designed to create "Secure Enclaves"
- TDX is virtualization-based (like AMD-SEV)

- No separate security processor
- New CPU mode, Secure Arbitration Mode (SEAM)
- Various binary modules expose required services
- Attestation performed by a Quoting Enclave



KVM FORUM

Supporting technologies

The flavor is in the details

Red Hat

How do we get this to work?

Host, Guest, Firmware and Hypervisor support

- Host and guest Linux kernel support
- Hypervisor support
- Guest firmware support
- Host provisioning and support tools, e.g. sevctl
- Generic key brokering and attestation
- Compatibility layers, virtual TPM, SVSM

- Referring you to the blog for pointers

Upcoming attractions

Watch out of these upcoming KVM Forum talks

- 2:00 Trusted I/O (Jeremy Powell)
- 2:30 Secure VM Service Module (Jörg Rödel)
- 3:00 Zero-trust virtual TPM (Claudio Carvalho)

KVM FORUM

Conclusion

Attestation means many different things.

Red Hat

Key takeaways

Attestation? We only scratched the surface!

- Confidential Computing is a large collection of technologies
- Attestation can mean very different things even in a same context
- Preserving chains of trust requires careful thinking
- Technologies are not consistent

- Please see blog for more details and links

KVM FORUM

Thank you

Now is a good time for questions

This Tao3D presentation is available at
https://github.com/c3d/presentations
(branch kvm-2023-chains-of-trust)

Red Hat