Building Blocks of Confidential Computing

Jörg Rödel, Linux Kernel Engineer @ SUSE
jroedel@suse.com
Twitter: @joergroedel

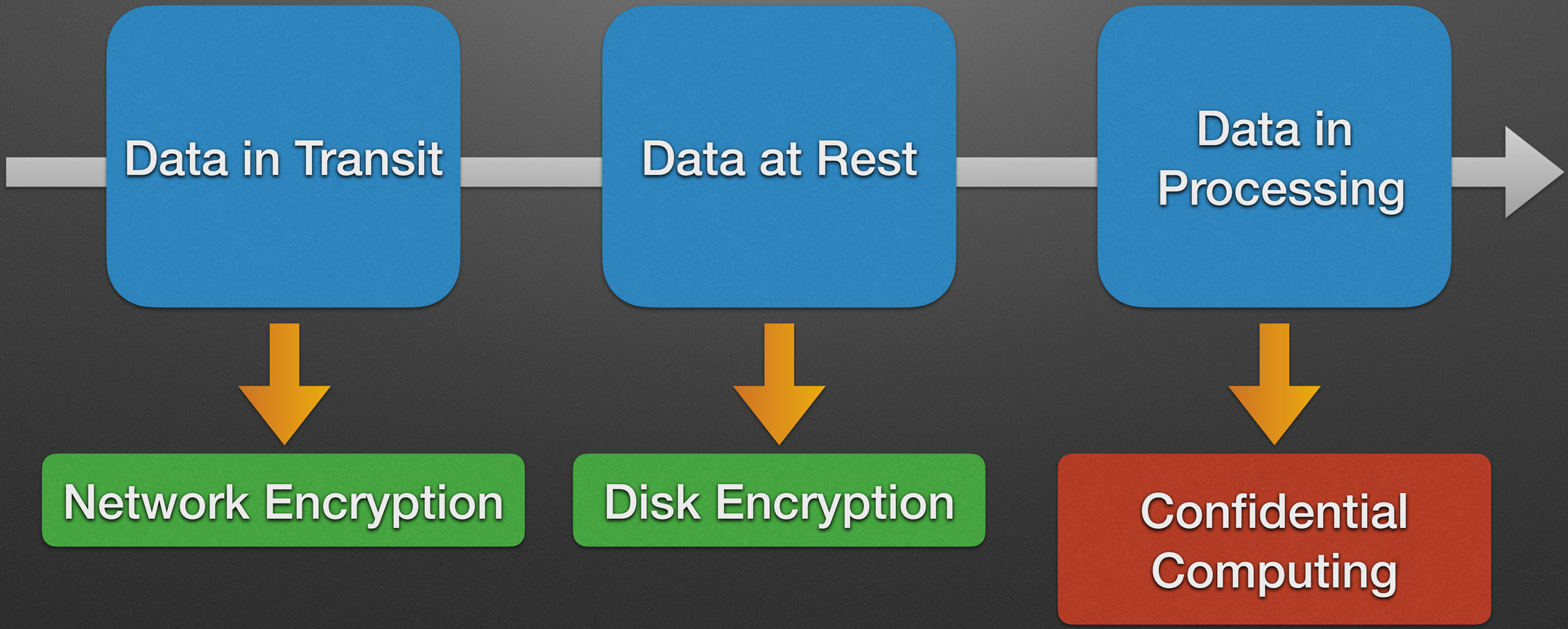
$ Whoami

- Linux Kernel Engineer @ SUSE
- Previously AMD and Amazon Web Services
- Focused on Virtulization, IOMMUs, x86 Architecture, PCI, …
- Working on Confidential Computing since 2018
- AMD SEV-ES Guest Support, PTI for x86-32, Nested Virtulization, …

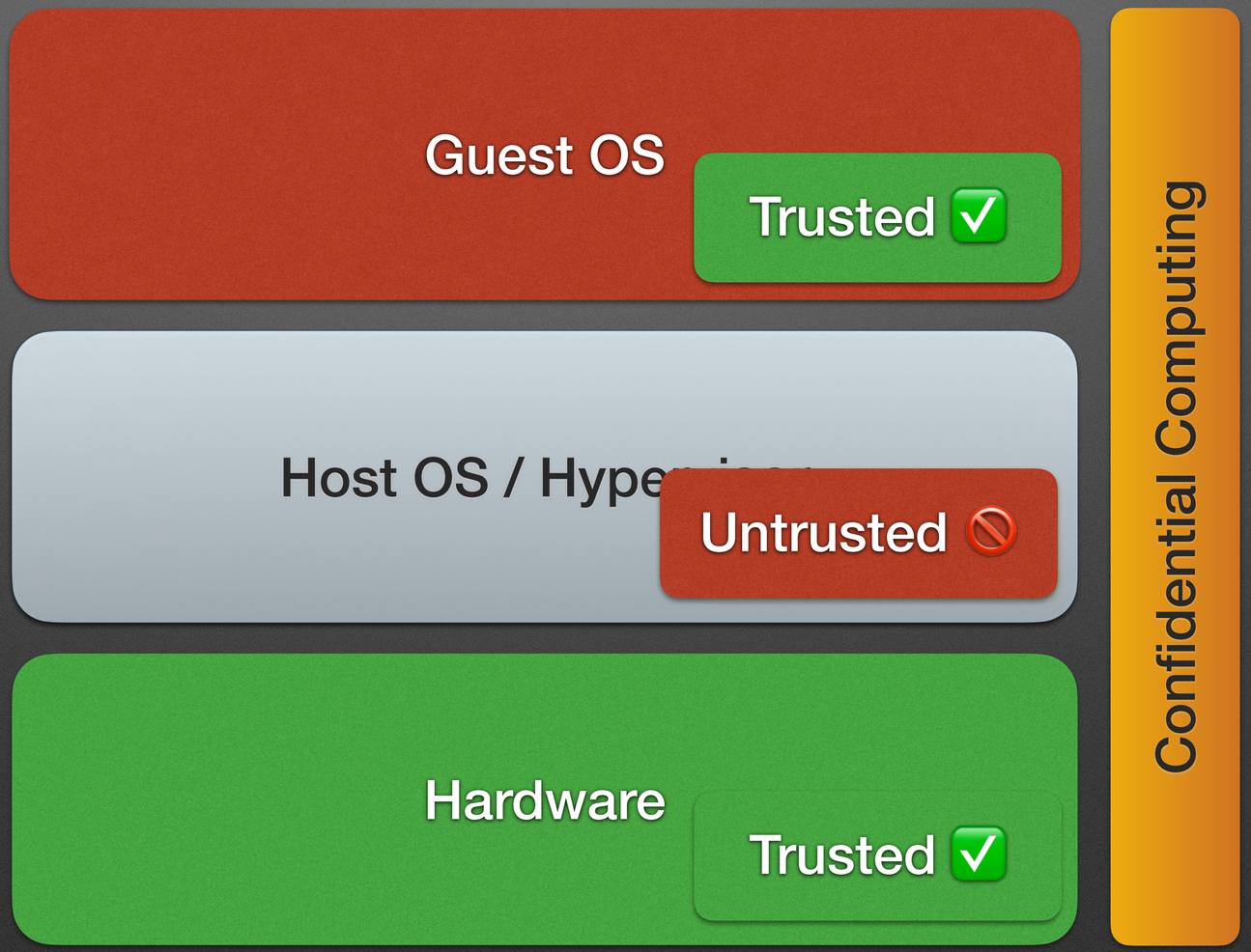
Confidential Computing

What it is and why it’s needed

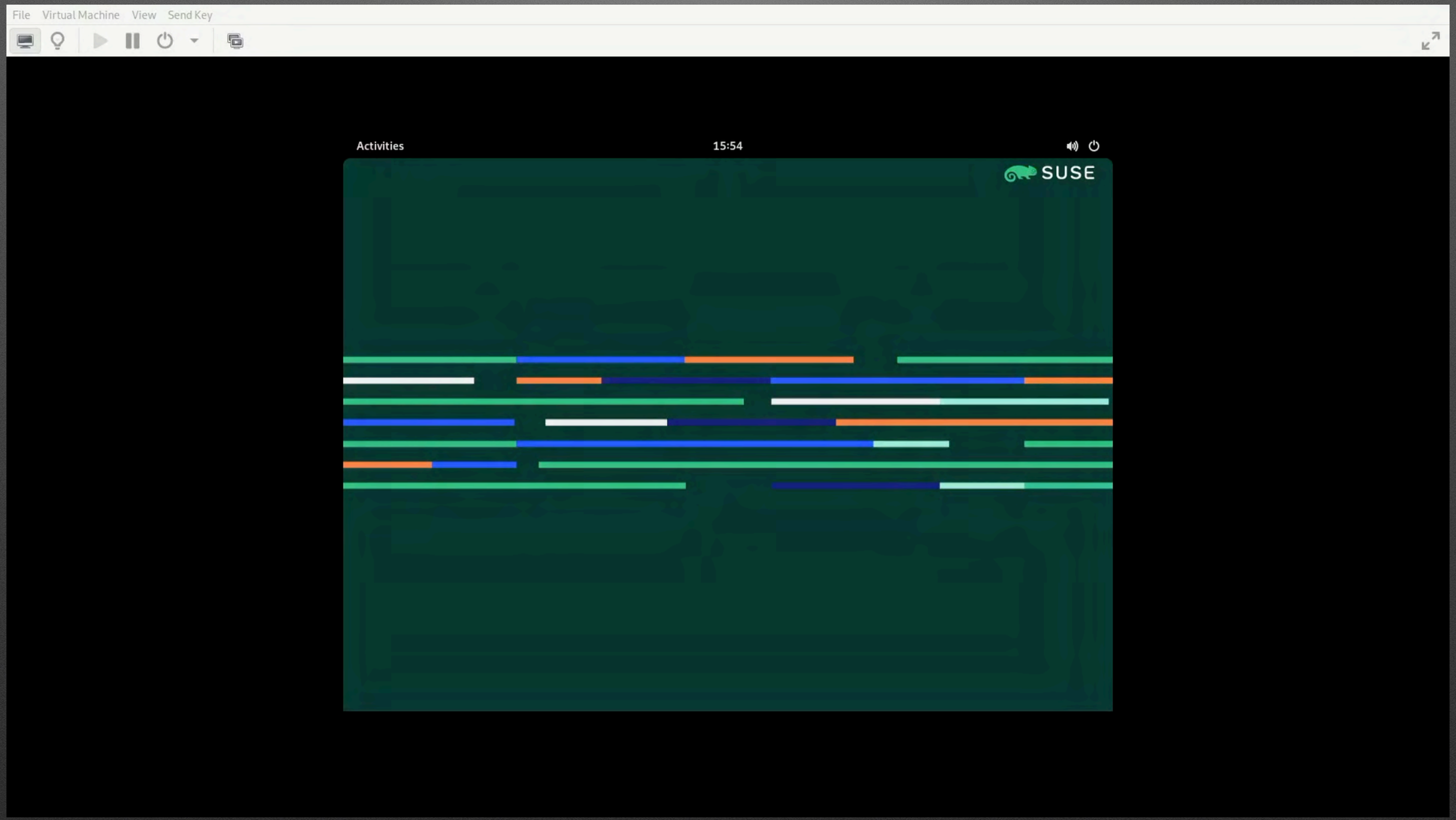
What is Confidential Computing?



Trusted Execution Base



Why Confidential Computing?



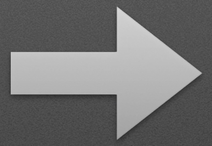
Building Blocks of a Confidential Computing System

Building Blocks

Hardware

Hardware

- Isolated Execution Environment
  - Memory encryption
  - State encryption
  - Replay and integrity protection
- Trusted verification services



Trusted Execution Environment

Hardware Extensions

VENDOR	EXTENSIONS
IBM	System Z Secure Execution
AMD	SEV, SEV-ES, SEV-SNP
Intel	TDX, SGX
ARM	Confidential Compute Architecture

Building Blocks

Host OS / Hypervisor

Hardware

Host OS / Hypervisor Kernel

- Setup and management of Trusted Execution Environment
  - Setting up initial state
  - Encrypting initial state
  - Handling requests from TEE
- Functionality moved from hypervisor into TEE

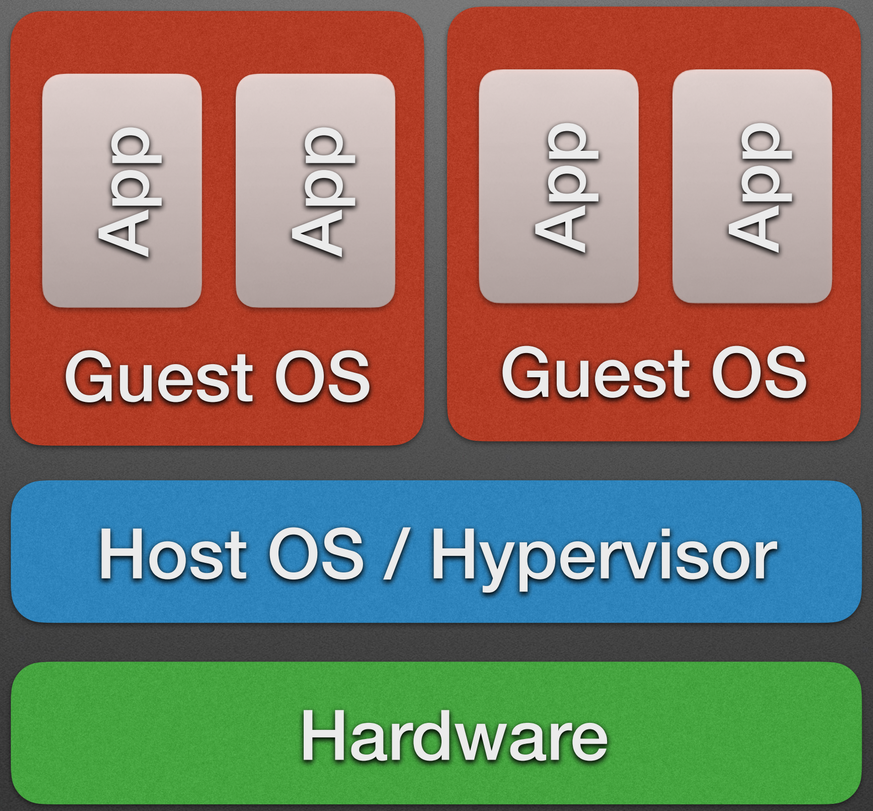
Host OS / Hypervisor Kernel

- Memory Model
  - Hardware can get mad when host tries to access private TEE memory
  - Depending on the hypervisor implementation this can get problematic
- Problematic for KVM - Work ongoing to implement a dedicated memory model

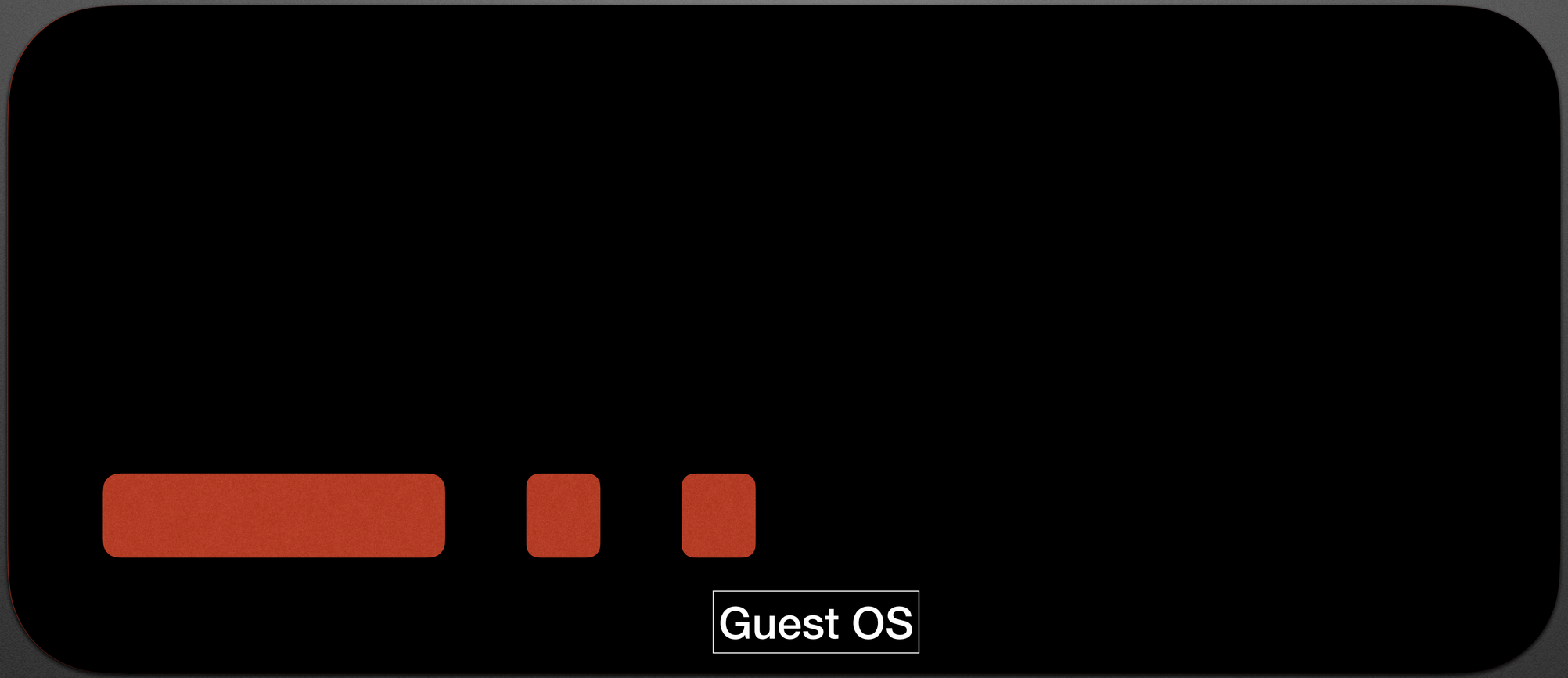
Host OS / Hypervisor Control Plane

- Secure channel between Trusted Verification Service and …
  - Guest Owner
  - Guest
- TEE disk image verification

Building Blocks



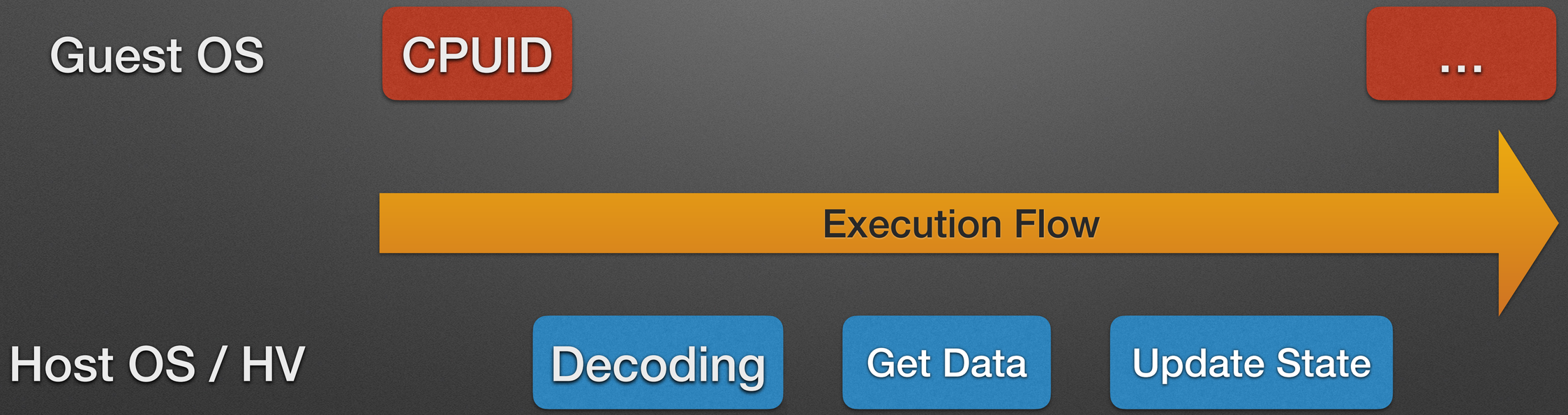
Guest OS Memory



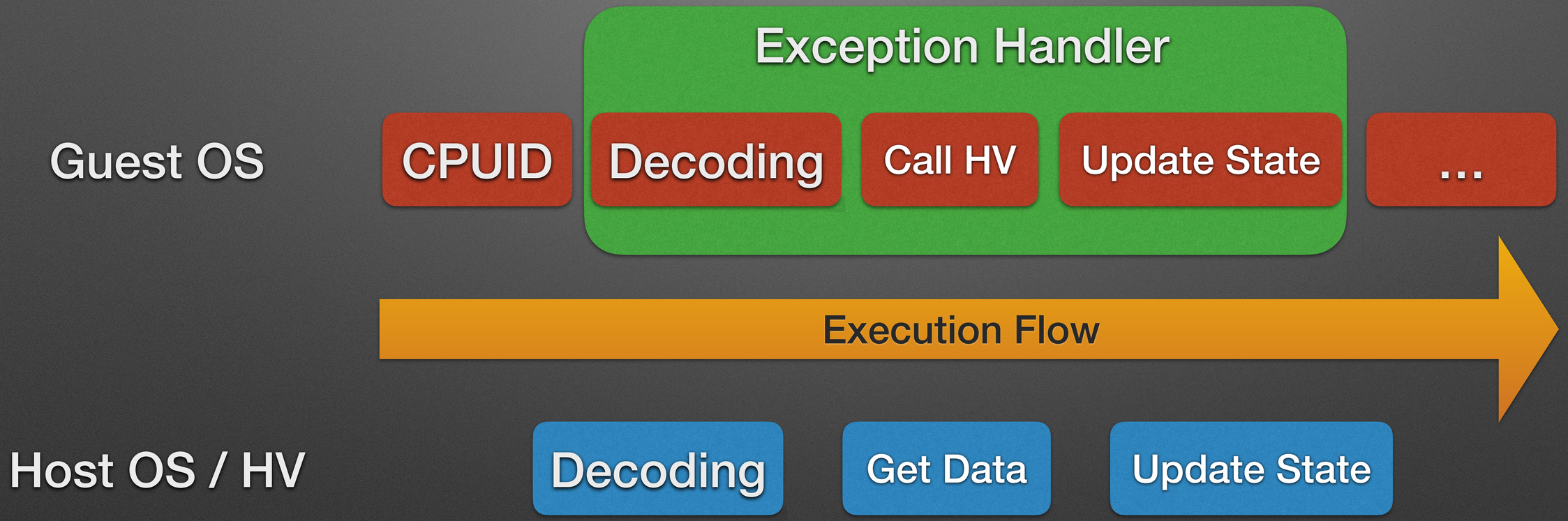
Guest OS State

- Hardware encrypts CPU register state of TEE
  - Invisible to hypervisor
  - Hypervisor can not handle all requests anymore
  - Requests need to be partially handled inside the TEE
  - Implemented with a new exception vector
- Paravirtualized protocol between TEE and Host OS / Hypervisor

Normal Guest-Host Flow



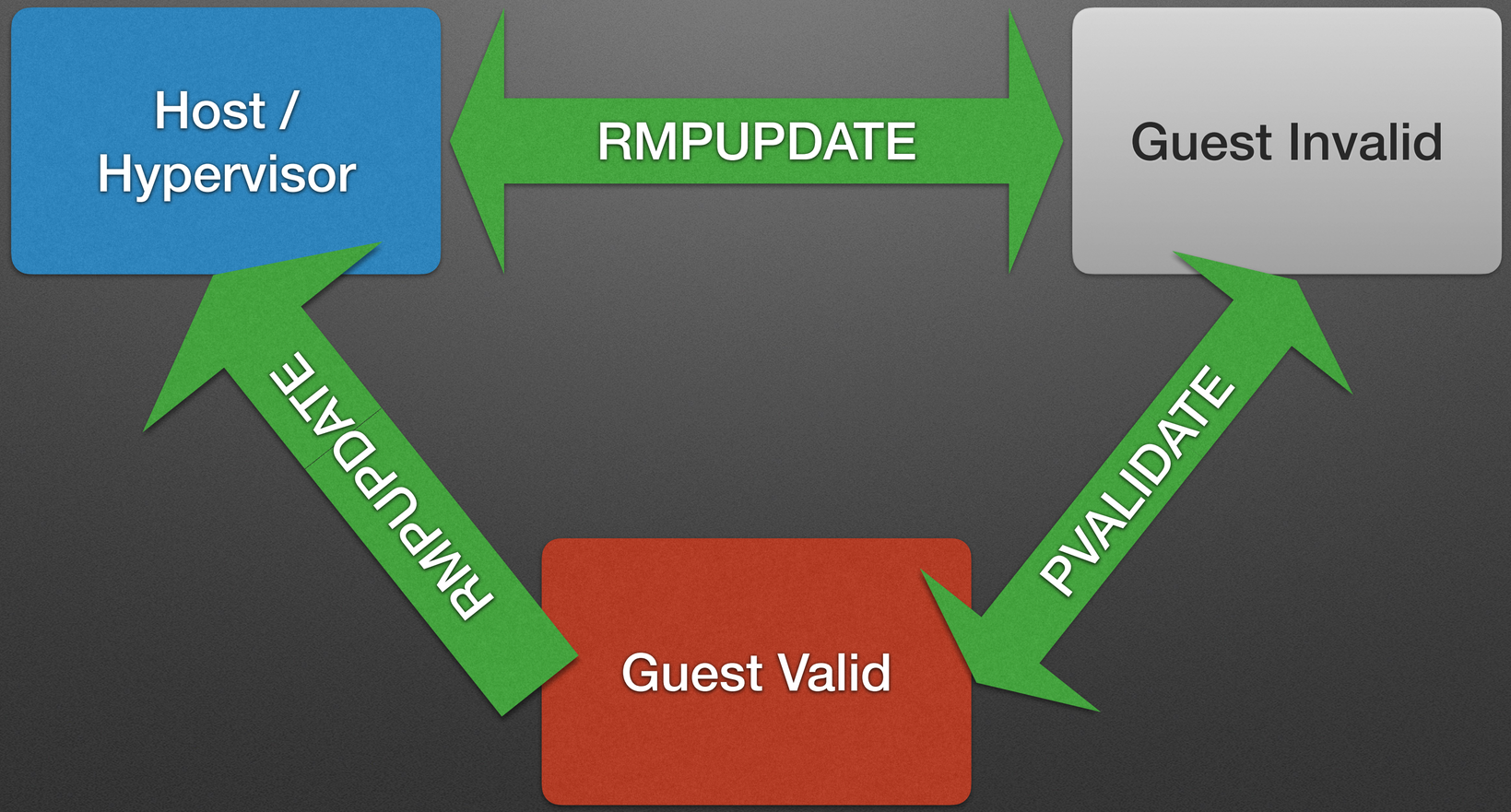
State Protected Guest-Host Flow



Memory Attacks

- Protecting encrypted data
  - HV could replay previous encrypted data
  - HV could remap encrypted pages
- Hardware provides protection
  - Implemented via page-states
  - RMP table on AMD SEV-SNP systems

Page States in AMD SEV-SNP



Page State Tracking

- Page states need to be tracked by Guest OS
  - Required to detect HV attacks
  - Avoid double validation
- Impact of double validation is hardware dependent
  - On AMD SEV-SNP it opens an attack vector

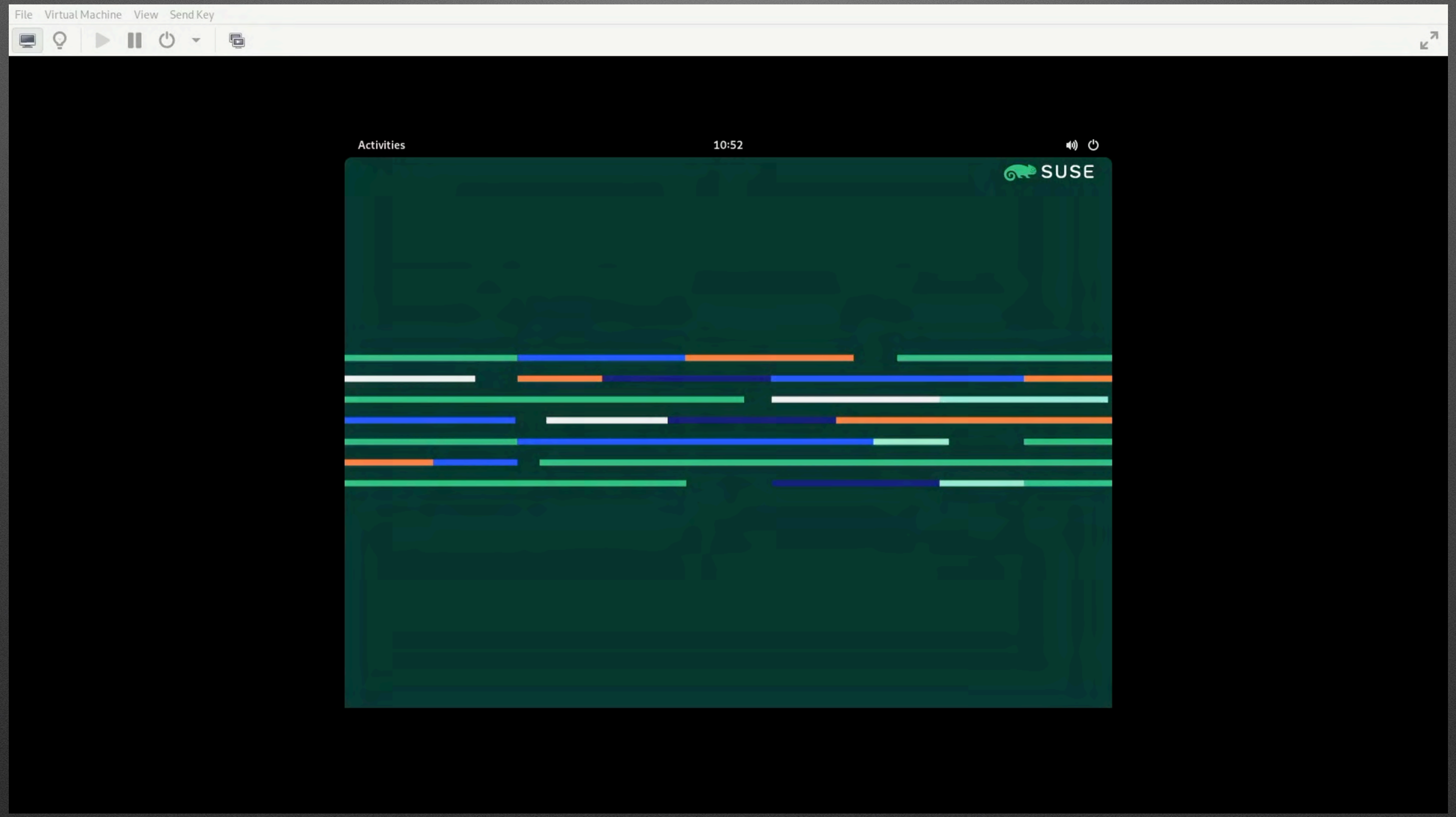
Guest-Controlled Disk Encryption

- Disk image managed by Hypervisor
  - HV can read/write disc contents
  - No secure channel from guest to disc
- All storage containing executable or sensitive data needs encryption and integrity protection
  - DM_CRYPT, DM_INTEGRITY, DM_VERITY, IMA, …

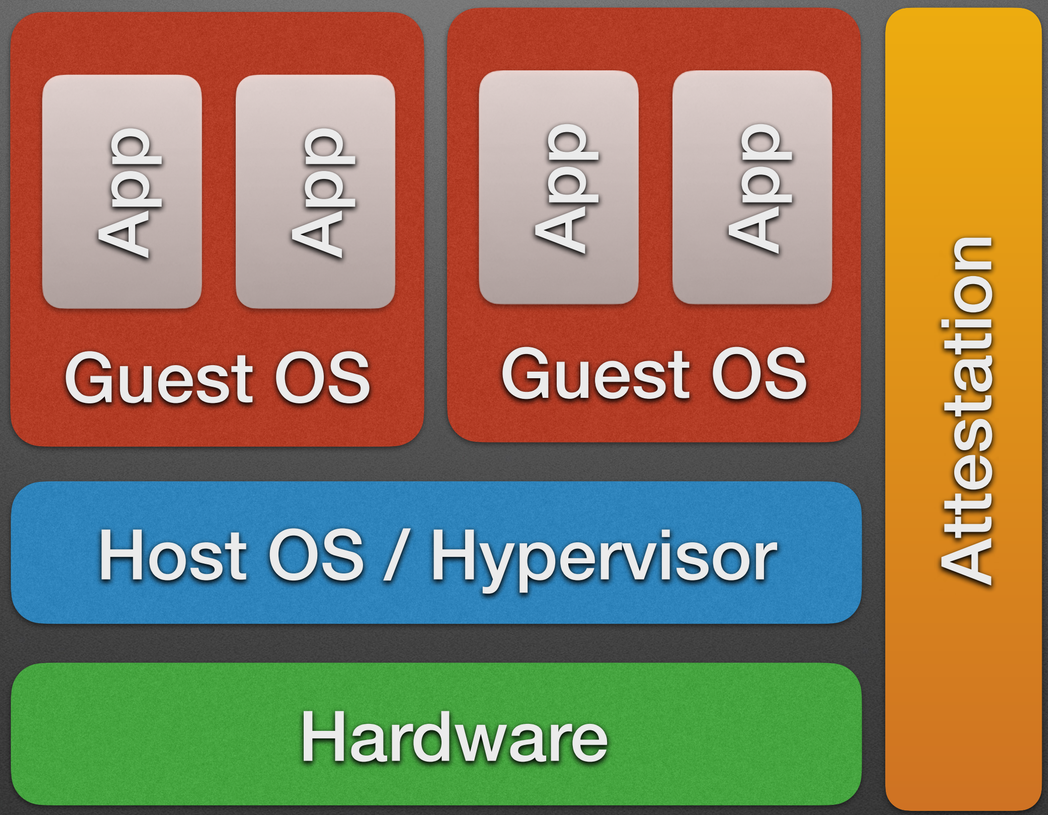
Device Driver Hardening

- Device drivers talk to the hypervisor
- Hypervisor-emulated devices are untrusted
- Device drivers need hardening to cope with malicious device input
- Ongoing work
  - Code Inspection
  - Device driver fuzzing

Encrypted Guest Demo



Building Blocks



Why Attestation is Needed

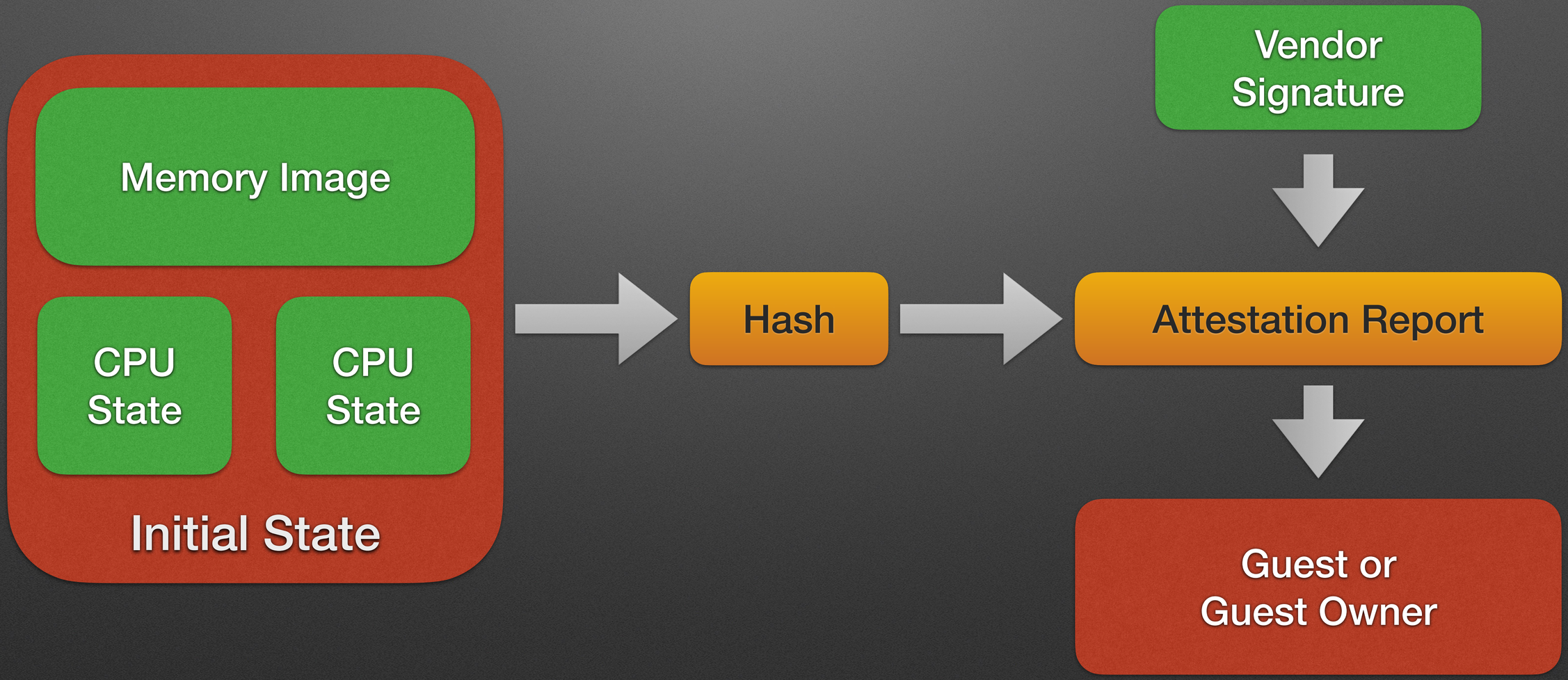
A Trusted Execution Environment requires trusted code!

Runtime Attestation

Boot Attestation

Hardware Attestation

Hardware Attestation



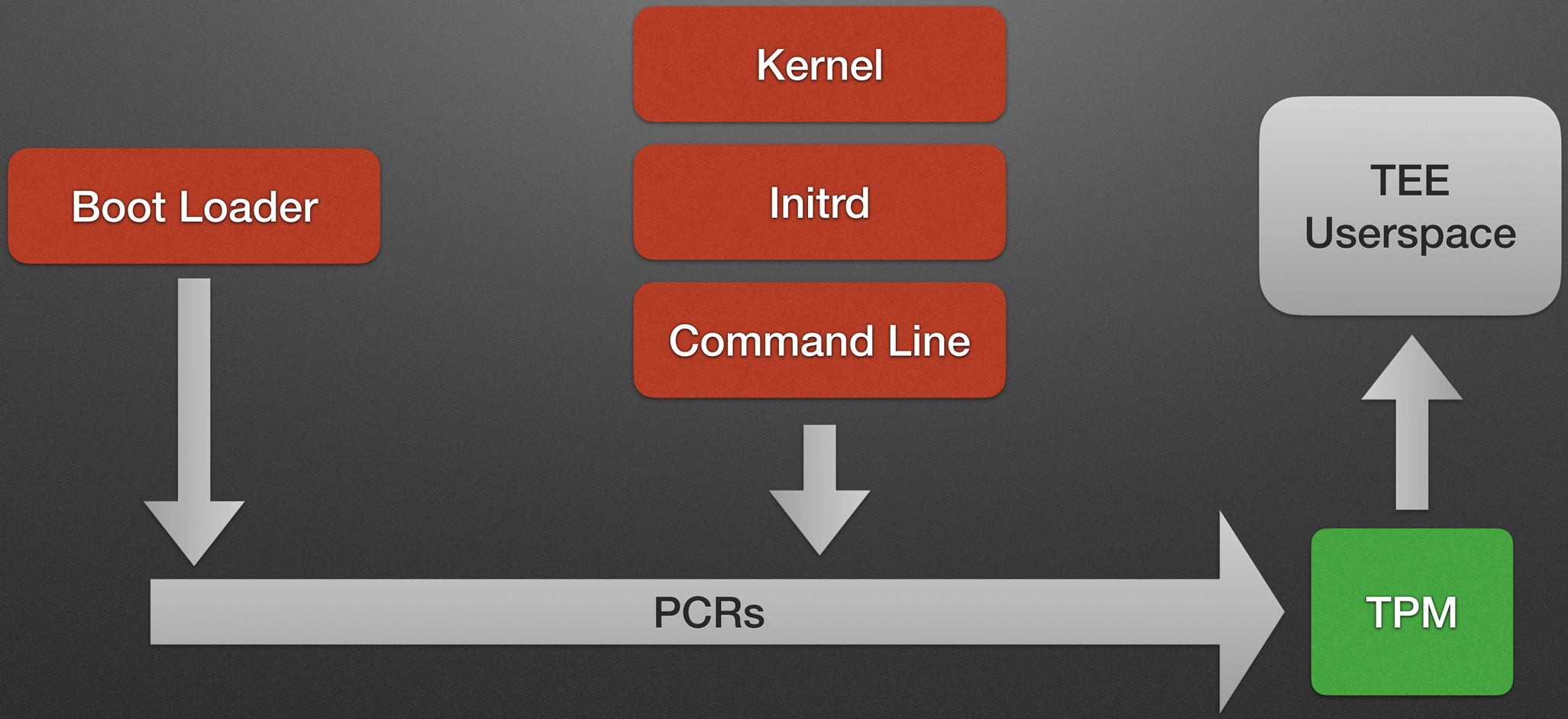
Boot Attestation

- Secure Boot
- Measure Boot Compents with TPM
- Signed Booting

Secure Boot

- OS vendor is the only trust source
- Secure Secure Boot requires SMM, which is not available in all TEEs
- Does not include the Initrd or kernel command line or boot loader configuration
- Attacker could still replace boot components with older ones from the same vendor

Boot Attestation with TPM



Signed Booting

- IBM System Z Secure Execution supports Signed Booting
- Kernel, Initrd and Commandline are combined to a single file
  - Encrypted
  - Signed with a Hardware Key
  - Bound to a single mainframe
- With disk encryption it provides full boot measurement without TPM

Runtime Attestation

- Measure binaries before execution
- Filesystem contains signed file hashes as extended attributes
- Puts trust in the OS and software vendors
- Linux: IMA and EVM

Thank You!

Questions?