AMD

CONFIDENTIAL GUEST SERVICES WITH SECURE VM SERVICE MODULE ON SEV-SNP

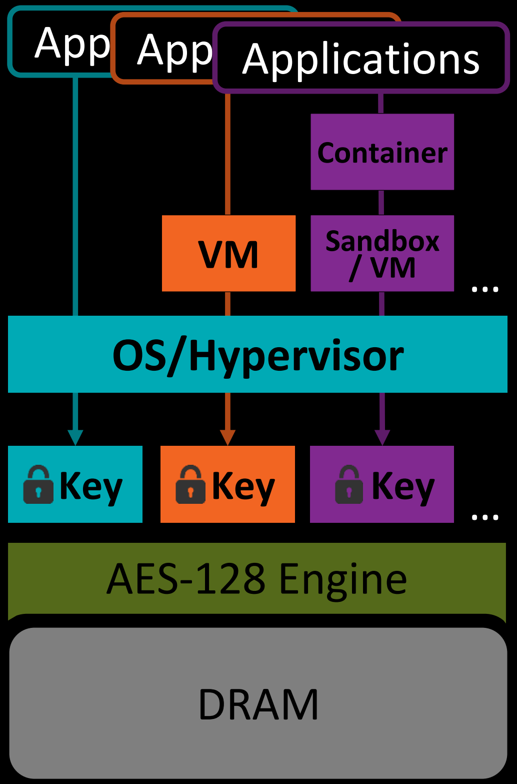
KVM FORUM - 2022

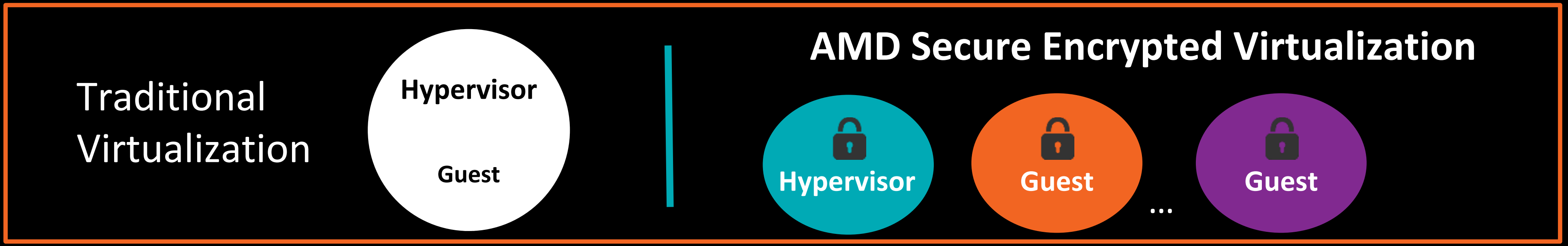
AGENDA

- SEV
  - Review
  - VM Privilege Levels
- SVSM
  - Overview
  - Benefits

SEV REVIEW

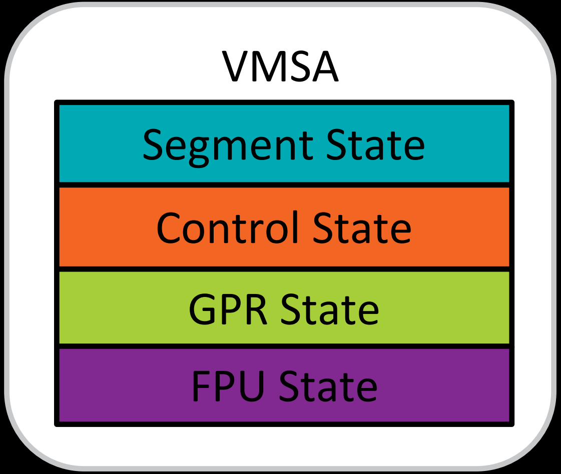
- Protects VMs/Containers from each other, administrator tampering, and untrusted Hypervisor
- One key for Hypervisor and one key per VM or VM/Sandbox with multiple containers
- Cryptographically isolates the hypervisor from the guest VMs
- Integrates with existing AMD-V™ technology
- System can also run unsecure VMs

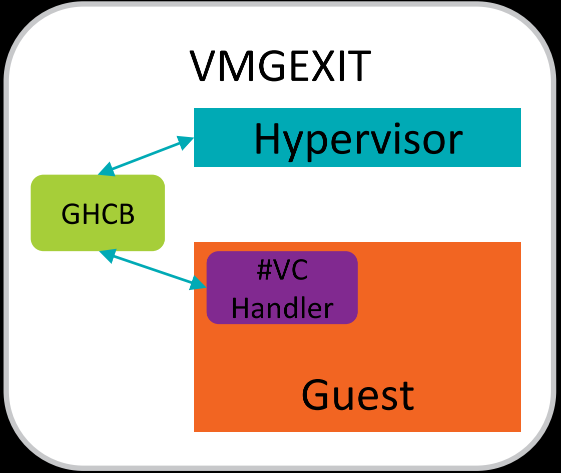



Traditional Virtualization
Hypervisor
Guest
AMD Secure Encrypted Virtualization
Hypervisor
Guest
…
Guest

SEV-ES REVIEW

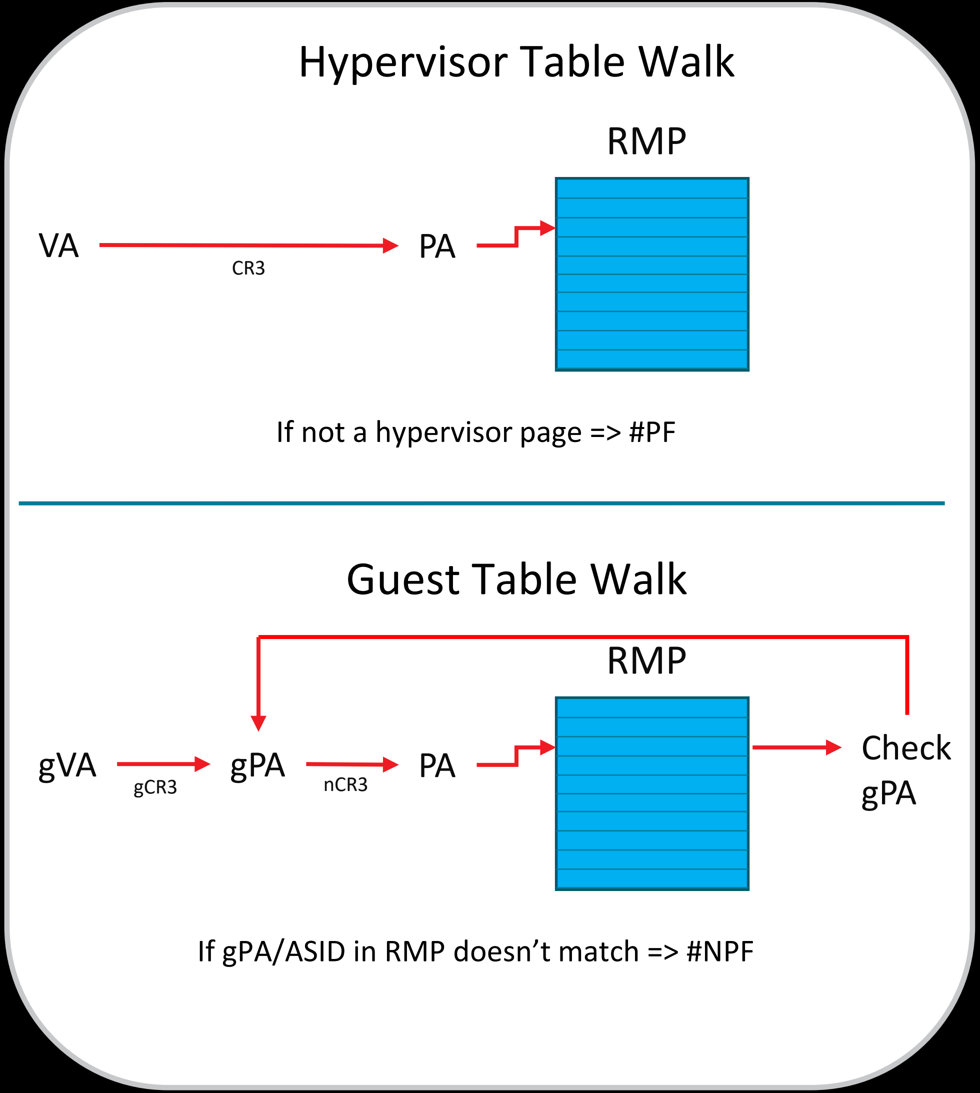
- Guest register state protection
  - Initialized with known state (Initial Processor State)
  - Encrypted and measured as part of the SEV LAUNCH process
  - Integrity check performed on each VMRUN
  - World switches now swap ALL register state
- VMCB under SEV-ES
  - Control Area (VMCB) and Save Area (VMSA) now separated
    - VMCB now points to VMSA
  - VMSA extended to save more state
- Guest-Hypervisor Communication Block (GHCB)
  - Allows guest ←→ hypervisor communication of the state needed to satisfy the guest service request
  - Shared (un-encrypted) page between the hypervisor and the guest
  - GHCB specification
    - Defines the format of the GHCB and how to communicate with the hypervisor





SEV-SNP

- Secure Nested Paging
  - Next step in the evolution of SEV
  - SEV/SEV-ES provides Confidentiality
    - SEV – Encryption of VM memory
    - SEV-ES – Adds Encryption of VM registers
  - SEV-SNP builds on SEV-ES and adds Integrity Protection
    - Prevents replay attacks, corruption attacks, remapping attacks
    - Utilizes the Reverse Map Table (RMP) and RMPUPDATE instruction to track:
      - Page Ownership: Hypervisor, Guest/PSP
      - Page Size: 4KB or 2MB
      - Guest Physical Address and ASID
      - VMSA (can be used with a VMRUN instruction)
        - LAUNCH_UPDATE or RMPADJUST instruction
      - Validation
        - PVALIDATE instruction
        - #VC if validation is changed by the hypervisor

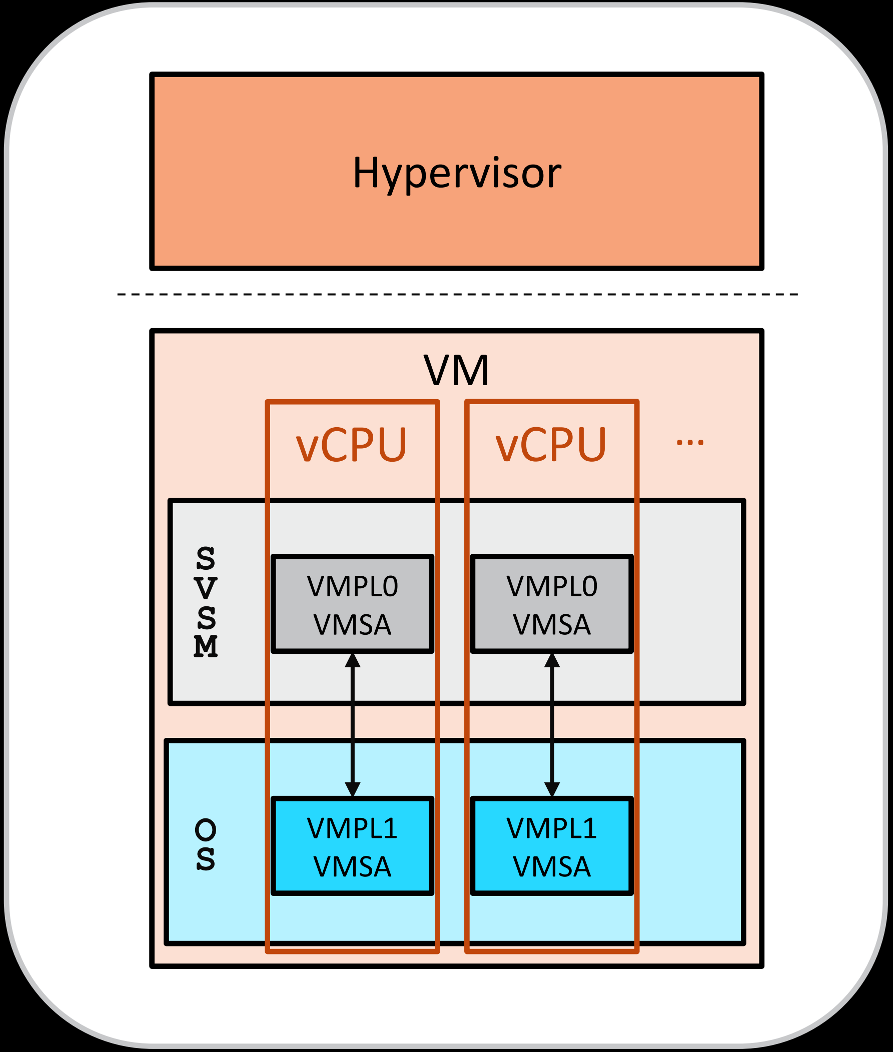


SEV-SNP…

- Virtual Machine Privilege Levels
  - Allows a guest to divide its address space in up to 4 levels
  - VMPL0 - VMPL3 (VMPL0 being most privileged)
    - Higher privileged VMPL can provide secure services for lower privileged VMPL
      - e.g.: VMPL0 can provide secure services for VMPL1
    - VMPL level set in the VMSA
    - KVM/Linux SNP guests run at VMPL0 today
  - Each RMP entry has page permissions for each VMPL level
    - Read, Write, Execute (User/Supervisor)
    - Guest can set permissions for a lower VMPL privilege level using RMPADJUST
      - Only VMPL0 can set the VMSA attribute for use in running a vCPU
    - Allows a higher VMPL to protect itself from a lower VMPL

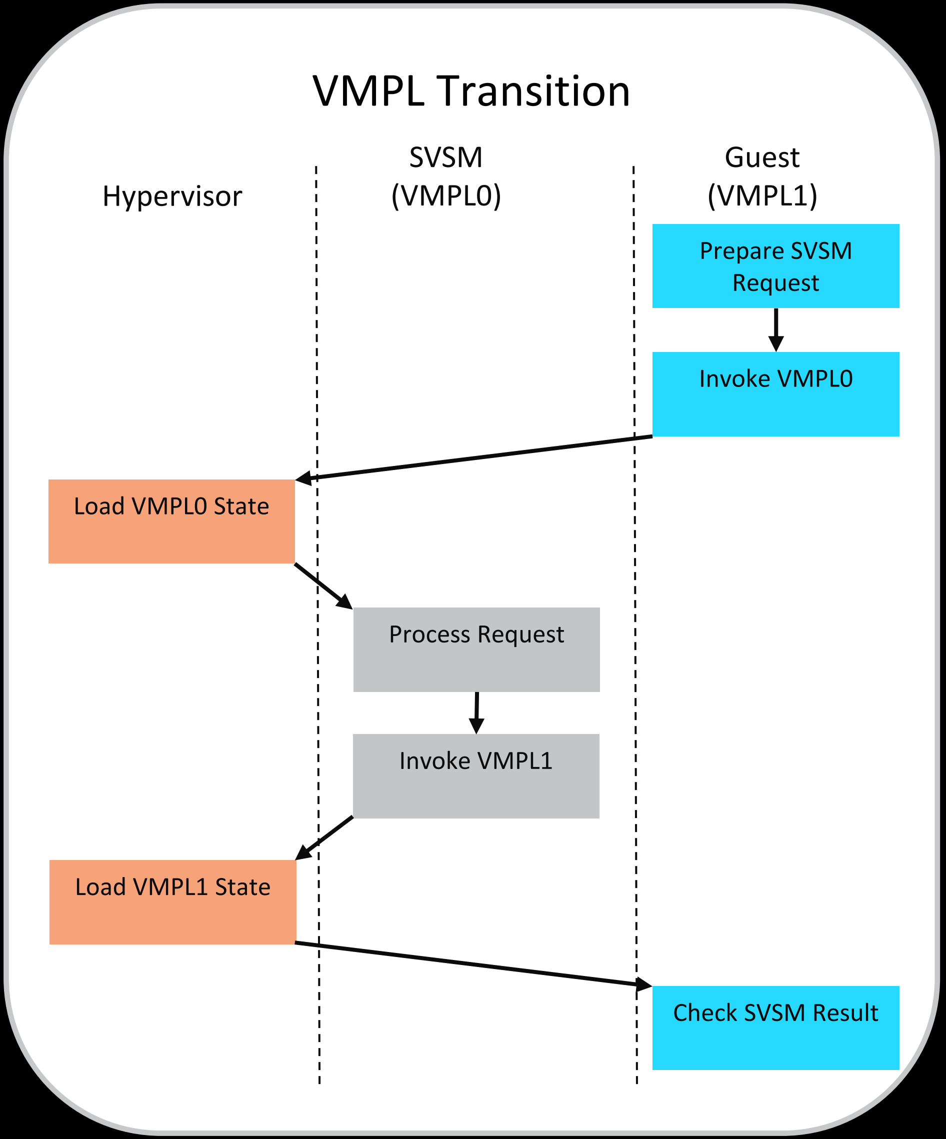
SVSM

- SVSM – Secure VM Service Module
  - Runs at VMPL0
    - BSP VMSA created and measured by the hypervisor
  - Creates VMPL0 VMSA pages for all APs
    - APs started using the GHCB AP Create Event
  - Creates a VMPL1 VMSA for the BIOS BSP
    - BIOS BSP VMSA contents measured by the hypervisor
    - Up to the BIOS to create VMSA's for APs
  - Uses:
    - Live Migration
    - vTPM
    - … and more

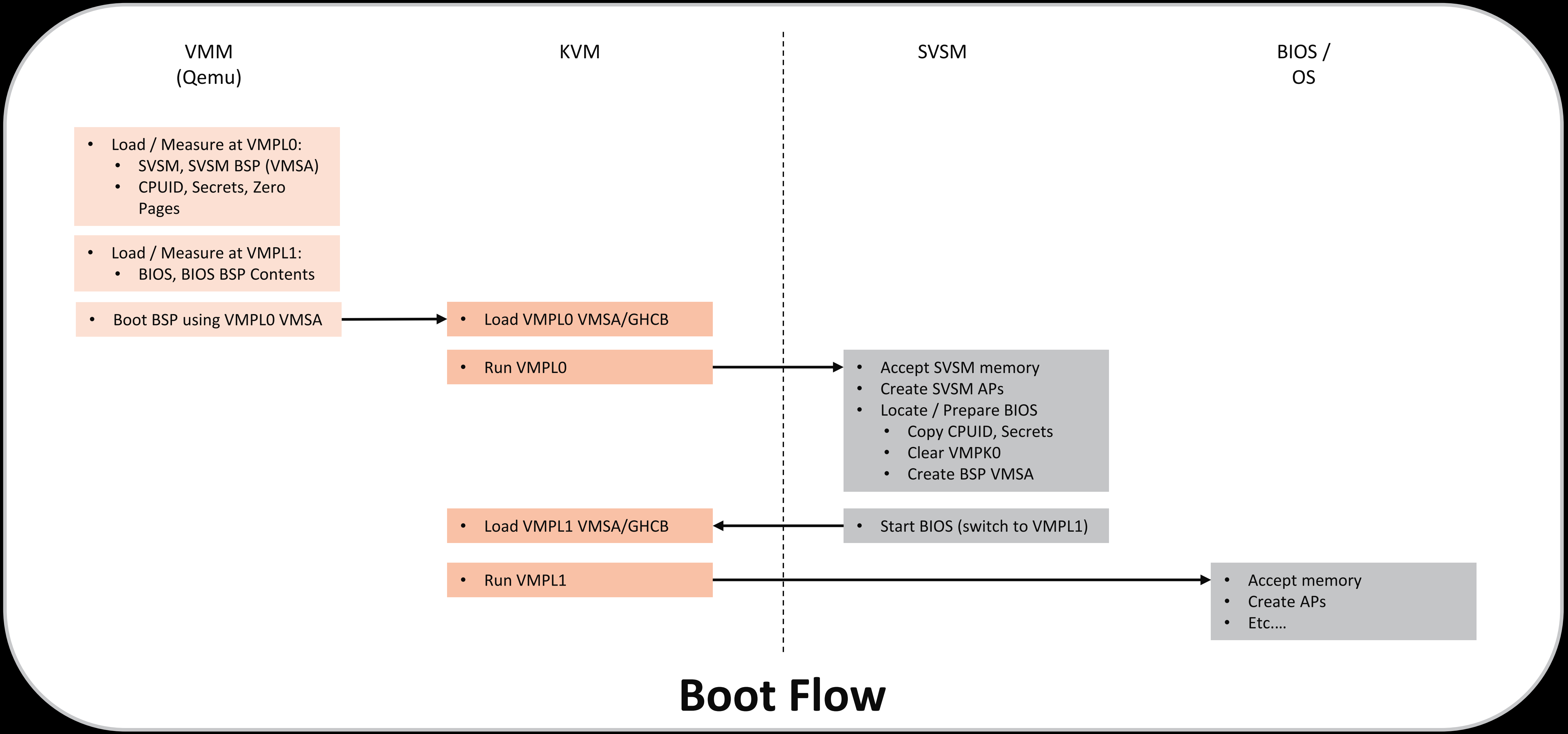


SVSM…

- SVSM / Guest Interaction
  - Switching between SVSM and Guest
    - New GHCB NAE Event to request running a VMPL level
  - Communicating with SVSM from Guest
    - SVSM specification defines:
      - Calling area page
        - Used to ensure a request has been made by the guest
      - Supported protocols and functions
      - Function input and output
        - Accessed through the Guest VMSA



SVSM…



Boot Flow

SVSM…

- SVSM VMM/Hypervisor Support
  - Load and measure SVSM and BIOS/OVMF
    - BSP state set to boot SVSM
  - Support one VMSA per VMPL level
    - GHCB AP Create enhancement for VMPL level
    - GHCB APIC ID list for guest
  - Switch between VMPL/VMSA

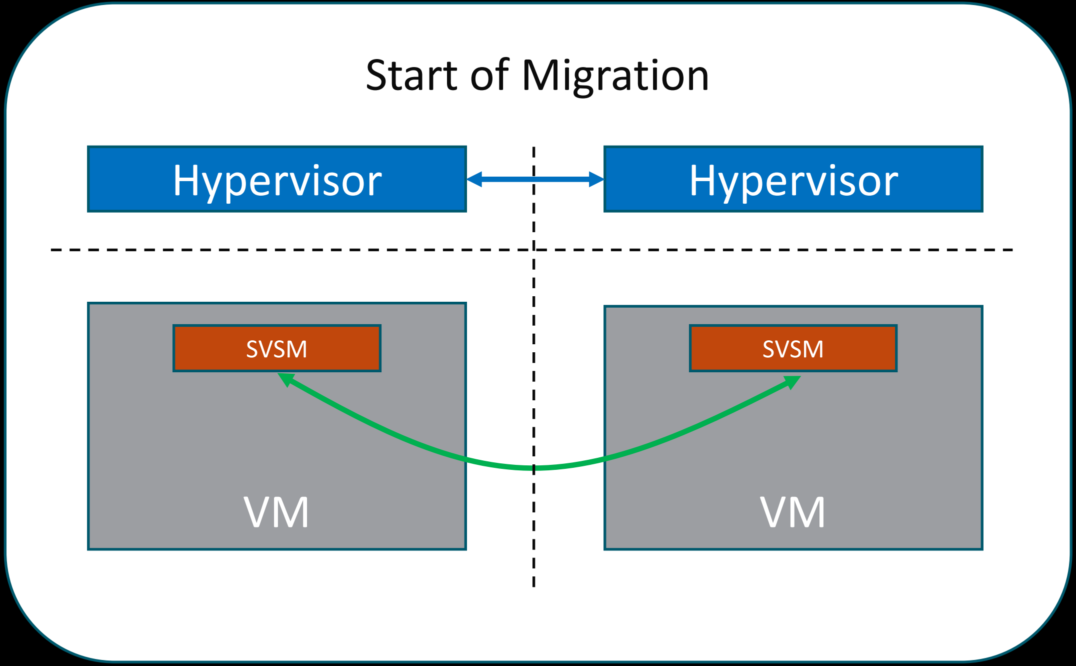
- SVSM Guest Support
  - Detect VMPL execution level
  - Invoke SVSM for PVALIDATE
  - Use GHCB AP Create NAE event for all APs
    - Retrieve APIC ID list from hypervisor
    - Invoke SVSM for RMPAJDUST (to make a page a VMSA page)

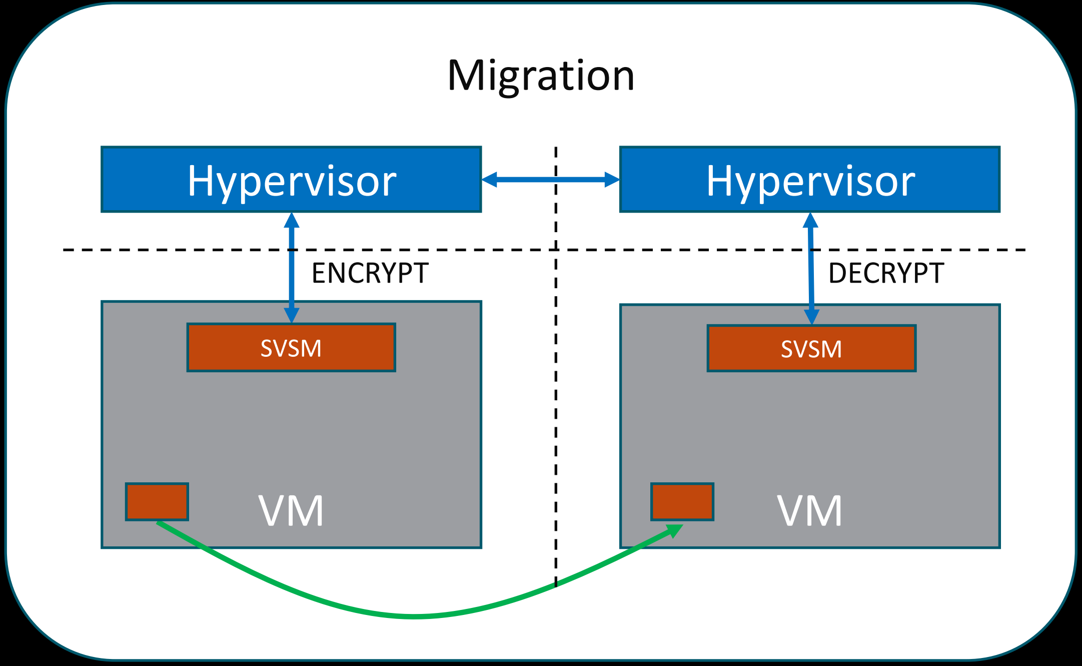
SNP LIVE MIGRATION

- KVM ←→ SVSM API required
- Hypervisor maintains a list of guest page encryption state
  - Page State Change from guest notifies of change in encryption state
- Source Hypervisor
  - Invokes the SVSM to prepare encrypted pages
    - Transforms the page for transport
    - Marks the page read-only at the guest VMPL level
- Destination Hypervisor
  - Invokes the SVSM to receive encrypted pages
    - Transforms the page for use in the guest
- At completion, guest state and guest page encryption state is migrated
  - Guest execution terminated on source
  - Guest execution initiated on target

SNP LIVE MIGRATION…

- SVSM started on destination
  - Transport key negotiated by source using destination attestation report
  - Any pre-migration state sent to destination
- SVSM migrates the guest memory
  - Hypervisor invokes SVSM to get/receive pages wrapped in the transport key
  - Source marks guest pages Read-Only (VMPL permissions) to ensure updated pages are migrated

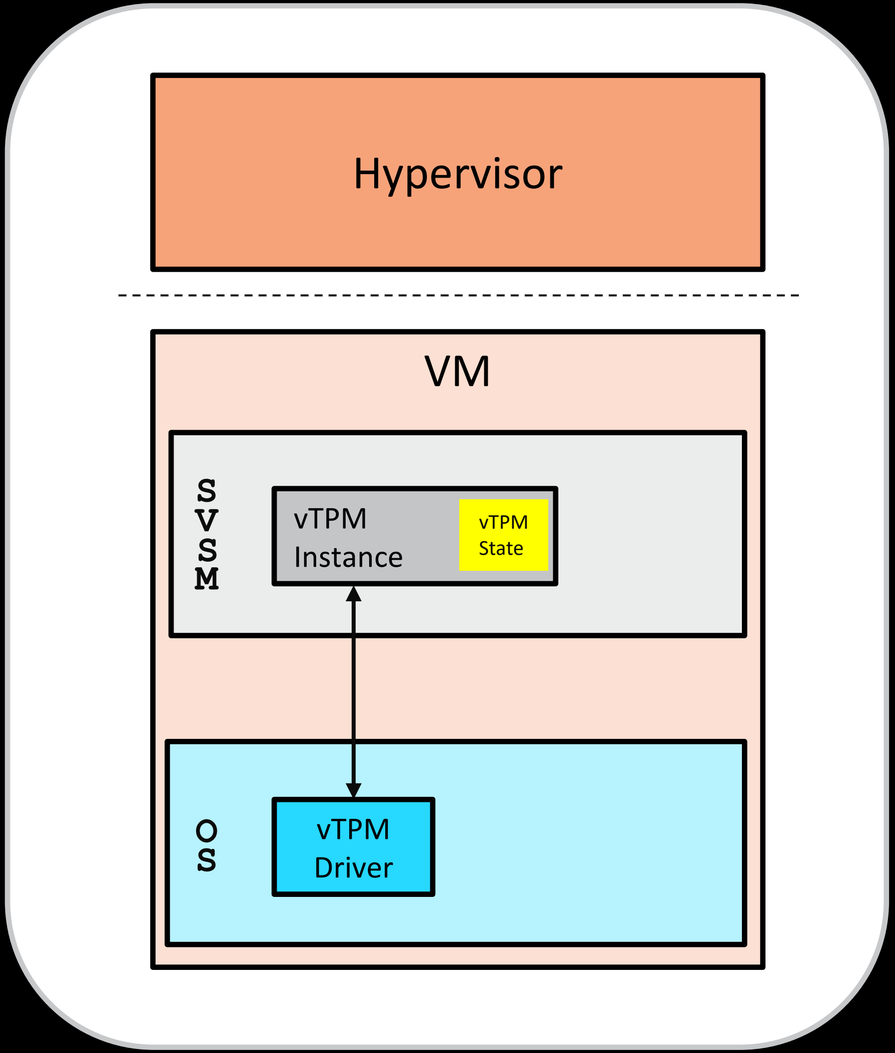




SVSM VTPM

- Virtual TPM instance
  - Service within the SVSM
    - New SVSM protocol and functions
    - Becomes part of the attestation report
    - Make VMPL0 attestation report available to VMPL1+
  - State maintained in the SVSM
  - Allows for secure boot of VMPL1+

- Questions on how to…
  - Persist TPM state?
  - Provide initial Endorsement Key?



SUMMARY

- Where are we at…
  - Proof-of-Concept Rust SVSM available
    - Support for most of SVSM protocol 0
    - Able to boot OVMF/Linux kernel at VMPL1
  - Guest Support (OVMF and Linux):
    - SVSM discovery
    - Page validation through SVSM
    - vCPU (VMSA) creation through SVSM
  - Hypervisor Support (Qemu and Linux):
    - SVSM recognition
      - SVSM measured at VMPL0
      - OVMF measured at VMPL1
      - BSP state from SVSM binary
    - vCPU multiple VMSA support (per VMPL)
    - vCPU run specific VMPL API

REFERENCES

- Links to the following reference material can be found at https://developer.amd.com/sev
  - White Papers & Specifications
    - AMD SEV-SNP: Strengthening VM Isolation with Integrity Protection and More
    - Guest Hypervisor Communication Block Specification
    - Secure VM Service Module Specification
    - AMD64 Architecture Programmer’s Manual Volume 2: System Programming

- Code / Patches (https://github.com/AMDESE)
  - SVSM Repo: https://github.com/AMDESE/linux-svsm
  - Hypervisor Patches:
    - Linux Kernel: https://github.com/AMDESE/linux/tree/svsm-preview-hv
    - Qemu: https://github.com/AMDESE/qemu/tree/svsm-preview
  - Guest Patches:
    - OVMF: https://github.com/AMDESE/ovmf/tree/svsm-preview
    - Linux Kernel: https://github.com/AMDESE/linux/tree/svsm-preview-guest

DISCLAIMER

AMD