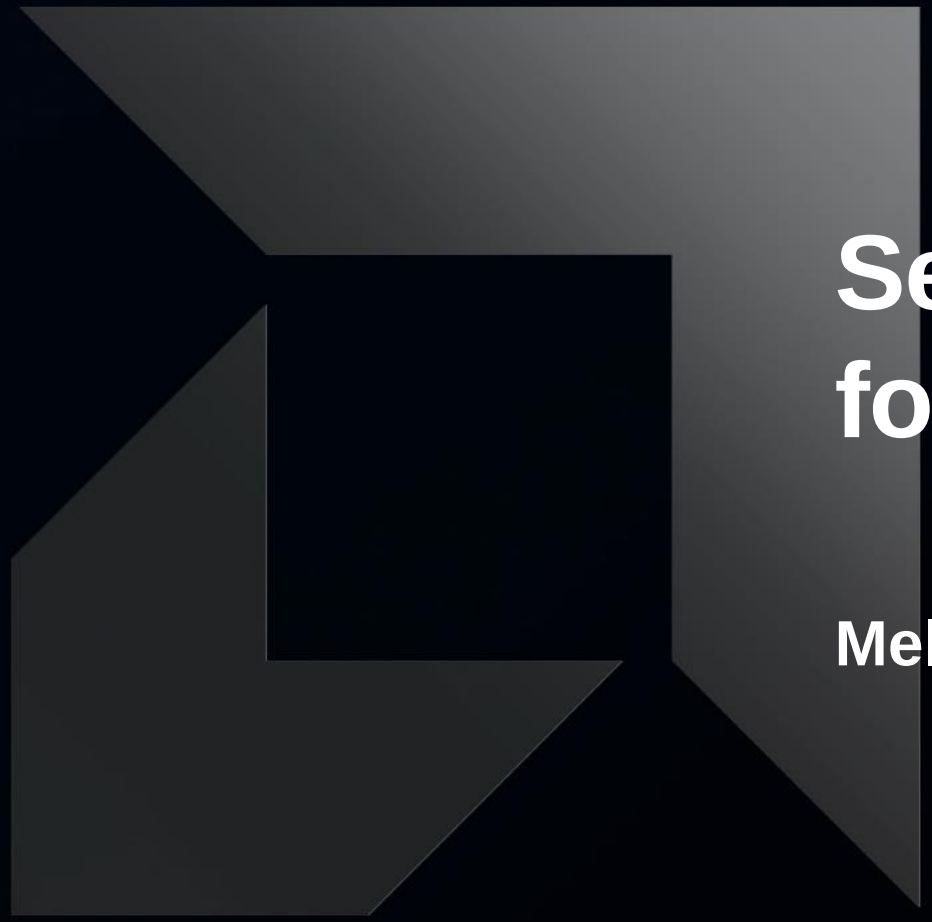


Secure Interrupt Delivery for SEV-SNP Guests

Melody Wang

Agenda

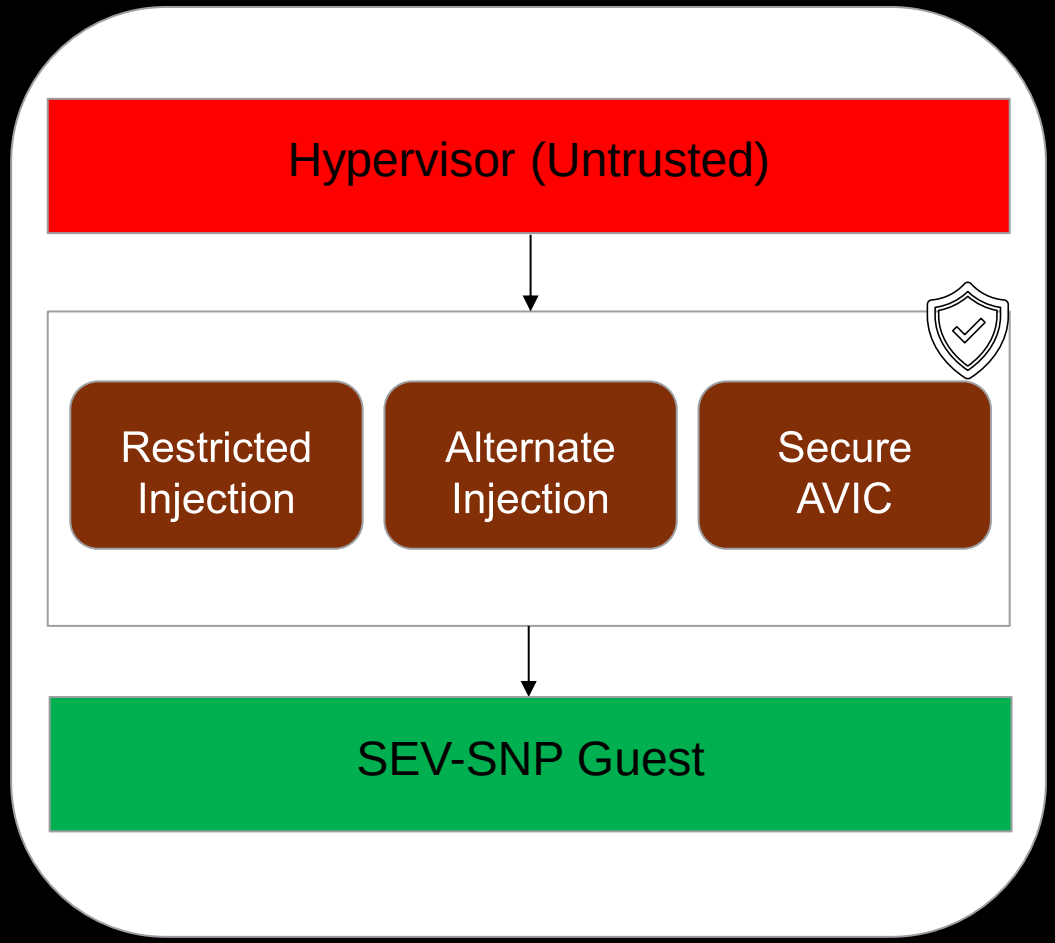
- Introduction
- Overview of Alternate Injection
- KVM support for Restricted Injection
- Alternate Injection support
  - KVM
  - SVSM
  - Linux Guest

Introduction

- Why is secure interrupt delivery needed?
  - With SEV-SNP, the hypervisor is untrusted and in control of interrupts injected
  - Examples: recent CVEs show the hypervisor can inject malicious interrupts to break the confidentiality and integrity of the guest
    - Virtual interrupt 29 (#VC)
    - Virtual interrupts 0 and 14
    - Int80
- Solution
  - A more restricted interface between VM and hypervisor regarding interrupts
  - VM can selectively accept/drop interrupts

Introduction

- Restricted Injection
  - Disable the virtual interrupt queuing and partially the interrupt injection interface
- Alternate Injection
  - Standard virtual interrupt queuing and injection interfaces
  - But controlled by the guest itself
- Secure AVIC
  - Advanced Virtual Interrupt Controller
  - Hardware acceleration for performance-sensitive APIC accesses
  - Support for managing guest-owned APIC state for SEV-SNP guests
  - Added to AMD64 architecture, appearing in a future part



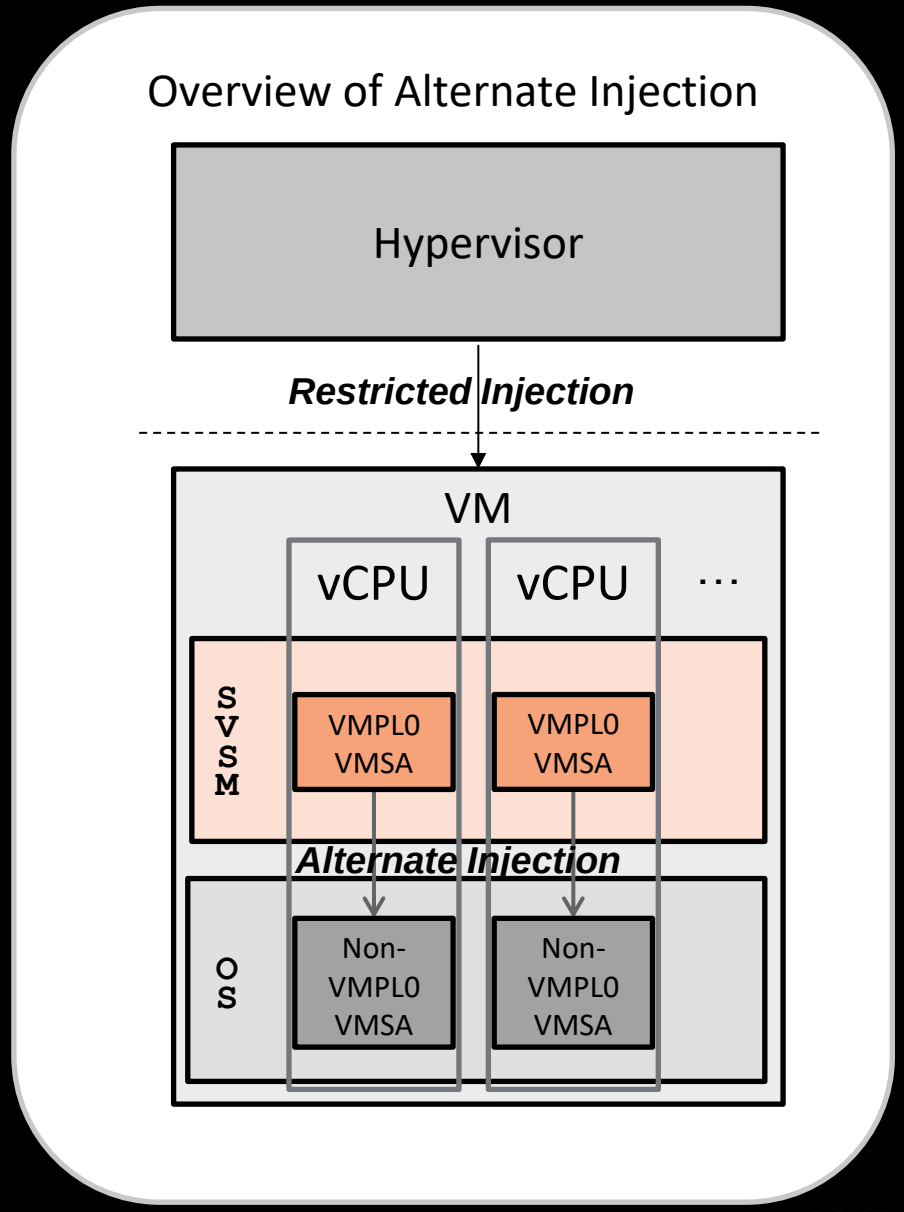
Restricted Injection vs. Alternate Injection

- Restricted Injection
  - Better performance than Alternate Injection
    - No Virtual Machine Privilege Levels (VMPL) transition
  - Not preferred for a Linux guest, facing obstacles upstreaming
    - X86 exception handling – nested exception detection problematic

- Alternate Injection
  - Restricted Injection would be into a Secure VM Service Module (SVSM)
    - Require VMPL transitions
  - Does not have the concerns related to Restricted Injection
    - Simpler, can handle nested exceptions properly

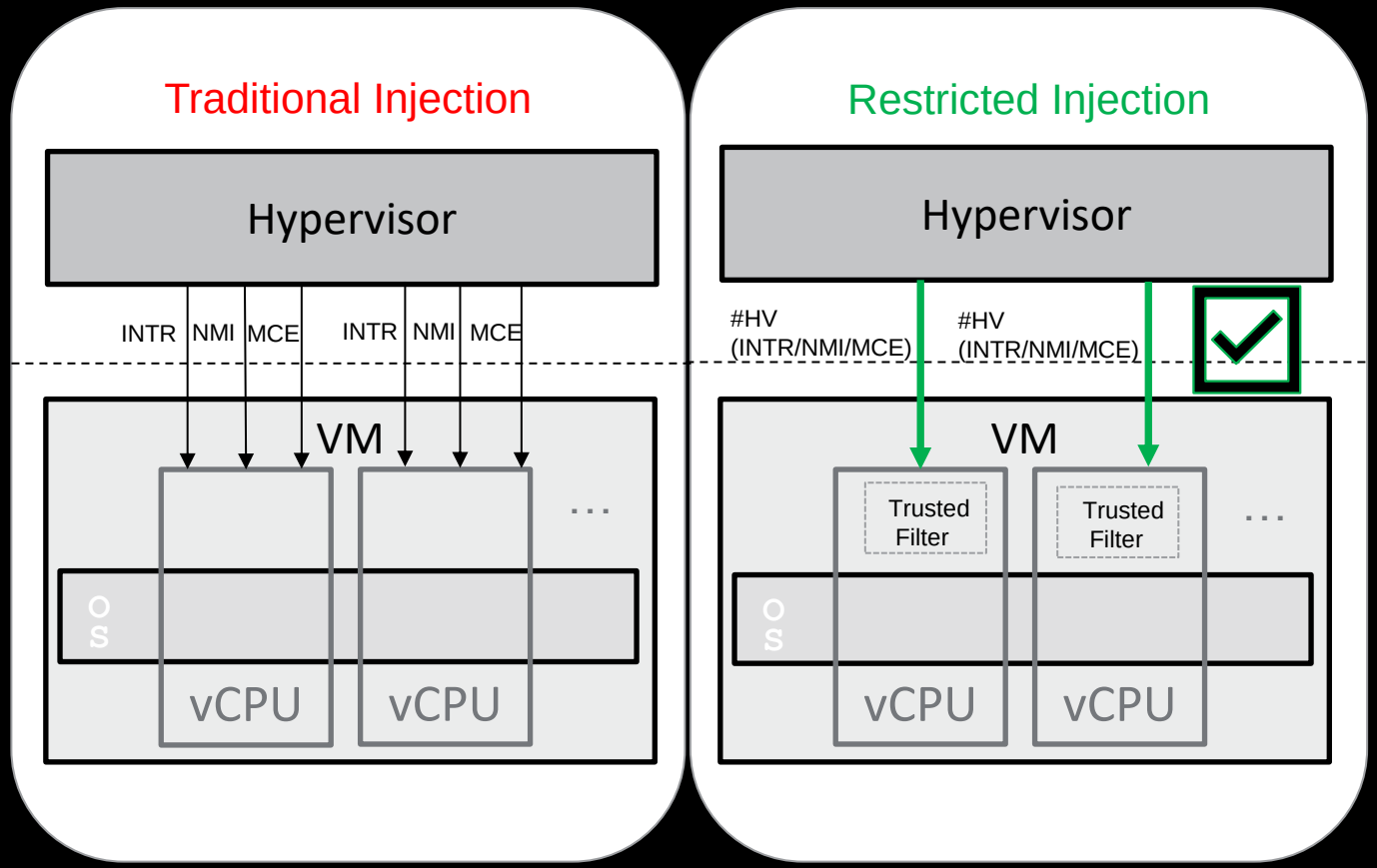
Overview of Alternate Injection

- Alternate Injection
  - Ensure interrupt presentation can only be performed by a trusted entity – the SVSM (Secure VM Service Module)
- Restricted Injection (Hypervisor -> SVSM)
  - Insulate the SVSM itself from malicious interrupts injected by the hypervisor
- SVSM (Secure VM Service Module)
  - Running at VMPL0 presents interrupts to the guest OS by writing to its VM Save Area (VMSA)



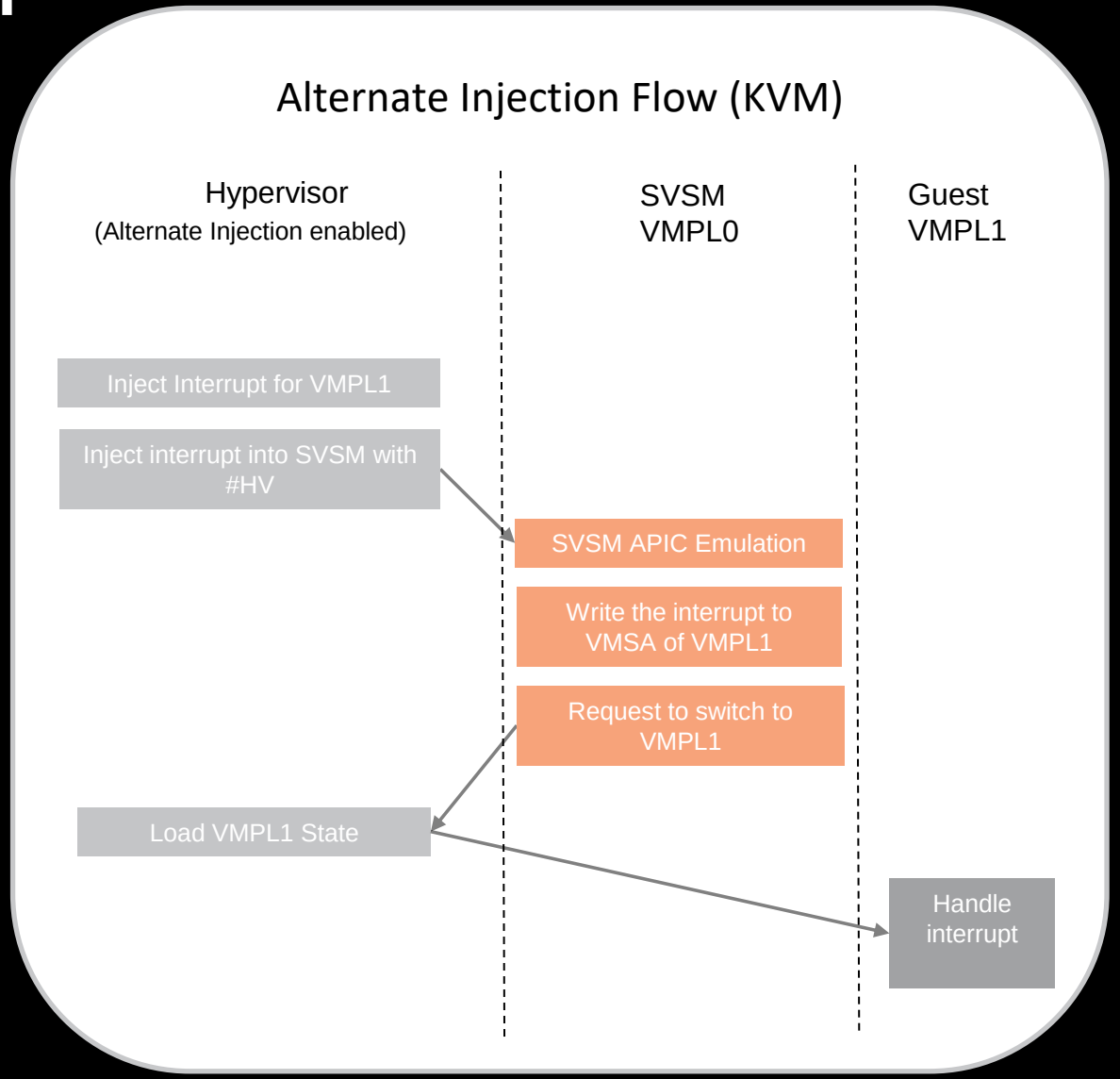
KVM support for Restricted Injection

- Restricted Injection into VMPL0 only guest
  - Restricted Injection enabling
    - Hardware support
    - Guest request
  - #HV – a new exception vector
    - Disable all hypervisor-based interrupt queuing and event injection of all vectors except #HV
    - Interrupts, NMI, and MCE are only allowed to be injected with #HV
  - KVM support for GHCB Restricted Injection Non-Automatic Exits (NAE) events
    - #HV doorbell page – register a doorbell page for use with #HV
    - #HV IPI – send an IPI to other vCPUs
  - Restricted Injection Timer
    - #HV Timer – request timer support from the hypervisor



KVM Support for Alternate Injection

- Alternate Injection enabling
  - Alternate Injection enabled in KVM
    - Hardware support
    - Guest request
  - Restricted Injection enabled in the SVSM
- Interrupt Injection for different VMPLs
  - The hypervisor needs to track a separate set of interrupt sources for each VMPL enabled for a given vCPU
  - Alternate Injection is not supported for VMPL0 itself
  - Interrupts for non-VMPL0 will be injected into the SVSM with #HV and then presented to the target VMPL by the SVSM writing the interrupt into its VMSA

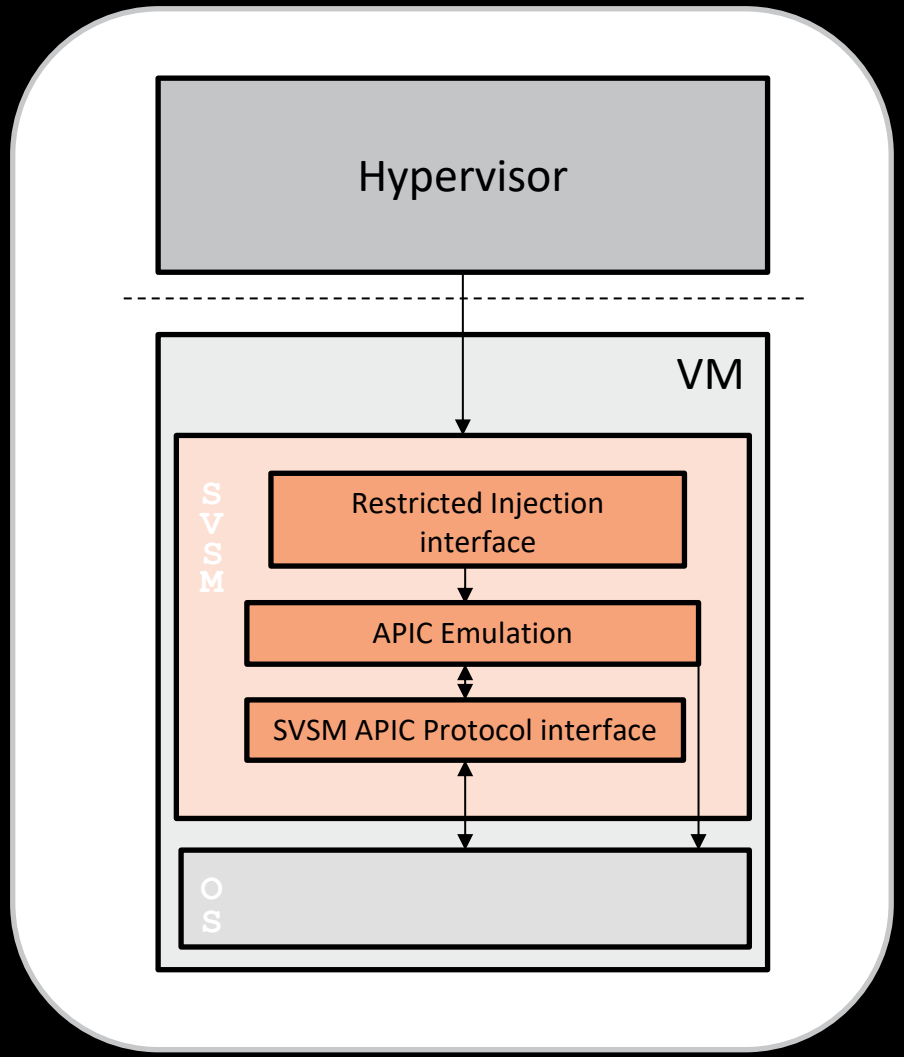


Proposed Changes to the GHCB to Support Alternate Injection

- Changes to GHCB Non-Automatic Exits (NAE)
  - Configure Injection Notification Vector
    - Configure the interrupt vector used by the hypervisor to notify the SVSM that interrupt injection processing is required
    - Can only be called by VMPL0
  - Disable Alternate Injection
    - Disable Alternate Injection for a specific VMPL
    - Interrupt will be placed into the IRR of hypervisor-emulated APIC and delivered to the target VMPL using direct event injection
    - Can only be called by VMPL0
  - #HV Timer
    - Extend the current #HV Timer NAE to permit signaling of timer interrupts by a lower VMPL
  - Specific EOI
    - Perform an EOI cycle on a level-sensitive interrupt
- Extended Interrupt Information
  - The interrupt descriptor is extended to two 16-bit words
  - Extra 32 bytes for each VMPL containing the interrupt information

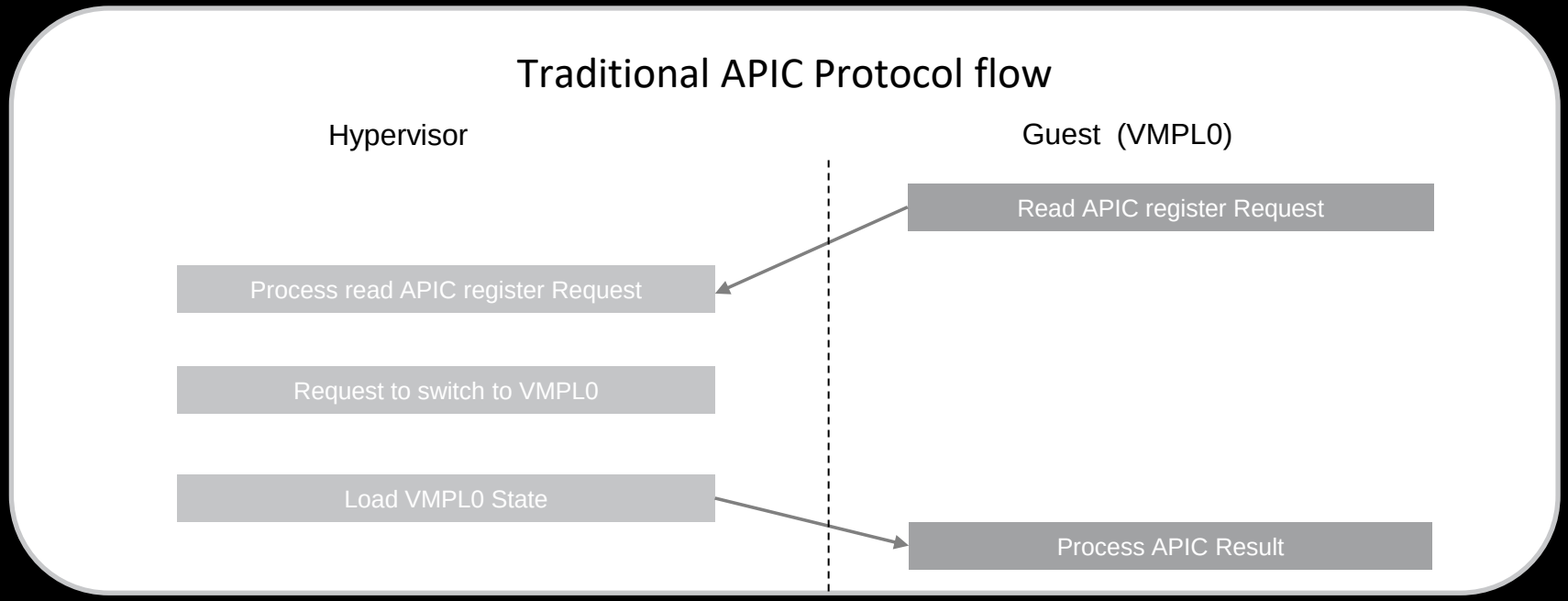
SVSM support for Alternate Injection

- Restricted Injection
  - Restricted Injection enabling for SVSM
  - #HV exception handling
- Alternate Injection protocol support
  - Alternate Injection enabling for serving the lower VMPLs
  - APIC emulation
    - Present interrupts to the different VMPLs
    - Protect guest states from being leaked to the hypervisor
  - SVSM APIC Protocol for the guest
    - Enable the lower VMPL to configure its APIC state through the SVSM
    - Query Features, APIC Emulation Configuration, Read/Write APIC Register, Configure Interrupt Vector



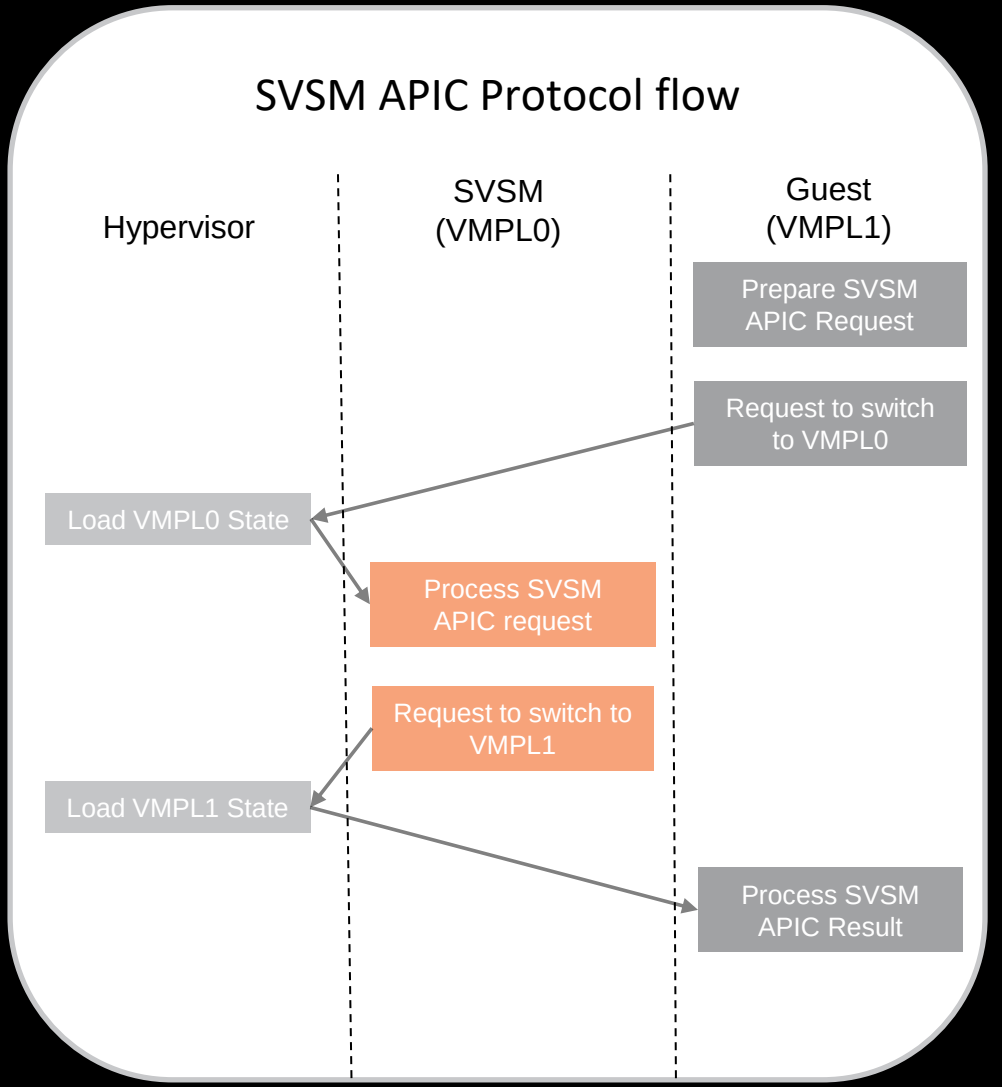
Traditional APIC Protocol

- Traditional APIC Protocol
  - KVM APIC emulation
  - The guest modifies its local APIC register by using this APIC protocol



Linux Guest Support for Alternate Injection

- Alternate Injection Enabling
  - Supporting multiple guest runtimes
    - Permit different configurations for virtual firmware and the guest OS
    - Three states are defined for handoff: Enabled, Disabled, and Locked
- Enlightening the guest for SVSM APIC Protocol
  - Recognize Alternate Injection is enabled
  - Detect attempts by the guest to modify its local APIC register and redirect to this APIC protocol
  - Use SVSM APIC protocol to communicate with the SVSM
    - SVSM Calling Area



Summary

- Where are we at…
  - KVM support for Restricted Injection into VMPL0 only guest
    - Sent upstream for review
    - Support for most of the GHCB requirements
    - #HV IPI - Under development
    - NoEoiRequired – Under development
  - KVM support for Alternate Injection
    - SVSM hypervisor support (Multi-VMPL support) - Under development
    - Work with the Multi-VMPL support to track a separate set of interrupt sources for each VMPL enabled for a given vCPU – To be done
  - Linux guest support for Alternate Injection
    - Under development
  - SVSM support
    - Restricted Injection for VMPL0
    - APIC emulation
    - SVSM APIC protocol
    - Restricted Injection for lower VMPLs – To be done

References

- Upstream thread of Restricted Injection for a Linux guest
  - https://lore.kernel.org/all/20230515165917.1306922-1-ltykernel@gmail.com
- Alternate Injection Specification
  - http://mail.8bytes.org/pipermail/svsm-devel/attachments/20240517/e74aeb2f/attachment.pdf
- SVSM (Multi-VMPL) Hypervisor talk
  - SVSM and VM Privilege Level instantiation and execution :: KVM Forum 2024 :: pretalx
- Code / Patches
  - KVM support for Restricted Injection into VMPL0 only guest
    - https://lore.kernel.org/kvm/cover.1722989996.git.huibo.wang@amd.com/
  - SVSM support patches
    - https://github.com/coconut-svsm/svsm

DISCLAIMER

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED 'AS IS." AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

AMD, the AMD Arrow logo, AMD-V and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.


AMD