Veil: A Protected Services Framework for Confidential Virtual Machines

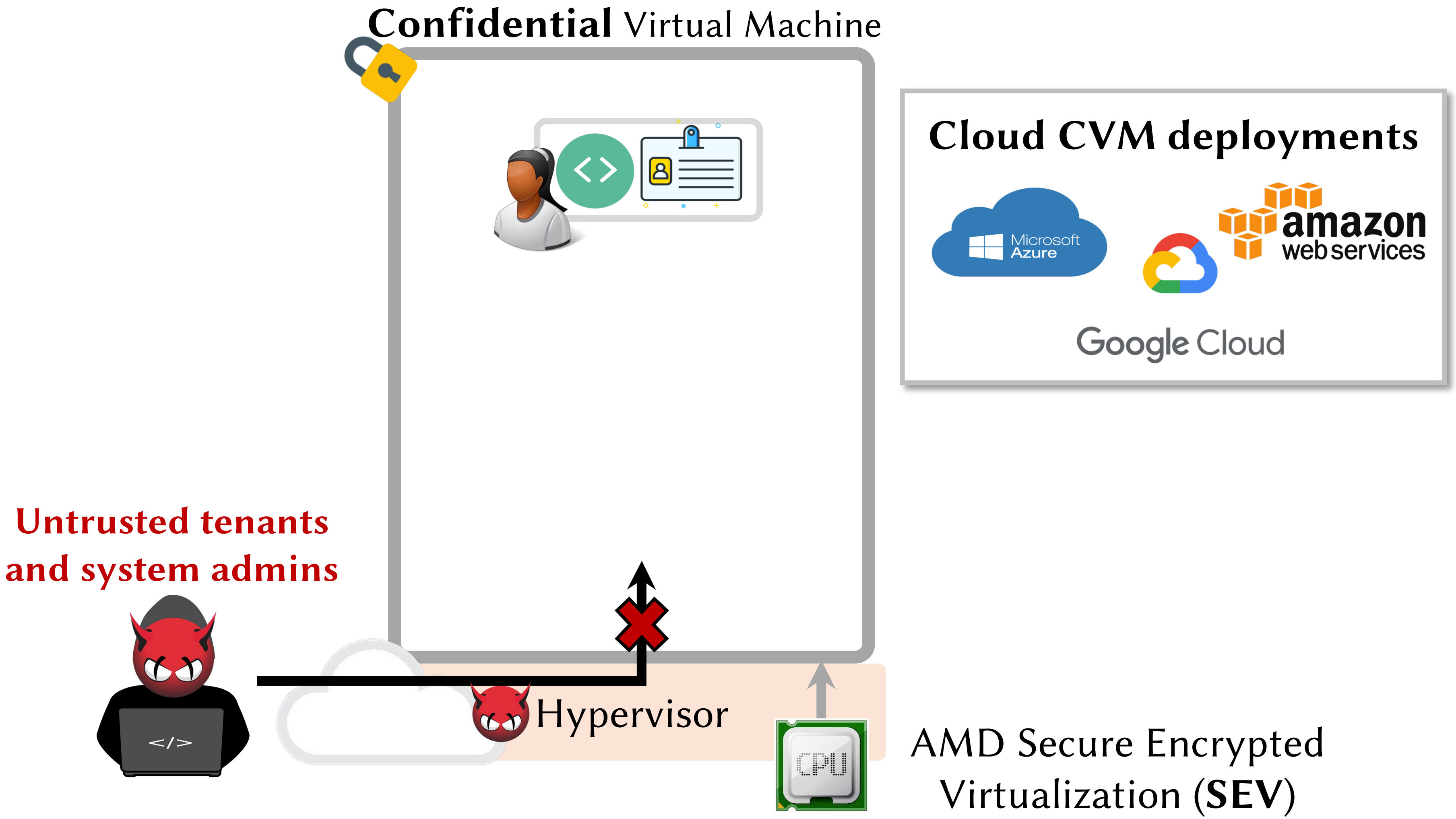
Adil Ahmad[1], Botong Ou[2], Congyu Liu[2], Xiaokuan Zhang[3], Pedro Fonseca[2]

1 ARIZONA STATE UNIVERSITY · 1885 ·

2 PURDUE UNIVERSITY

3 GEORGE MASON UNIVERSITY

CVMs restrict data access from outside components



Confidential Virtual Machine

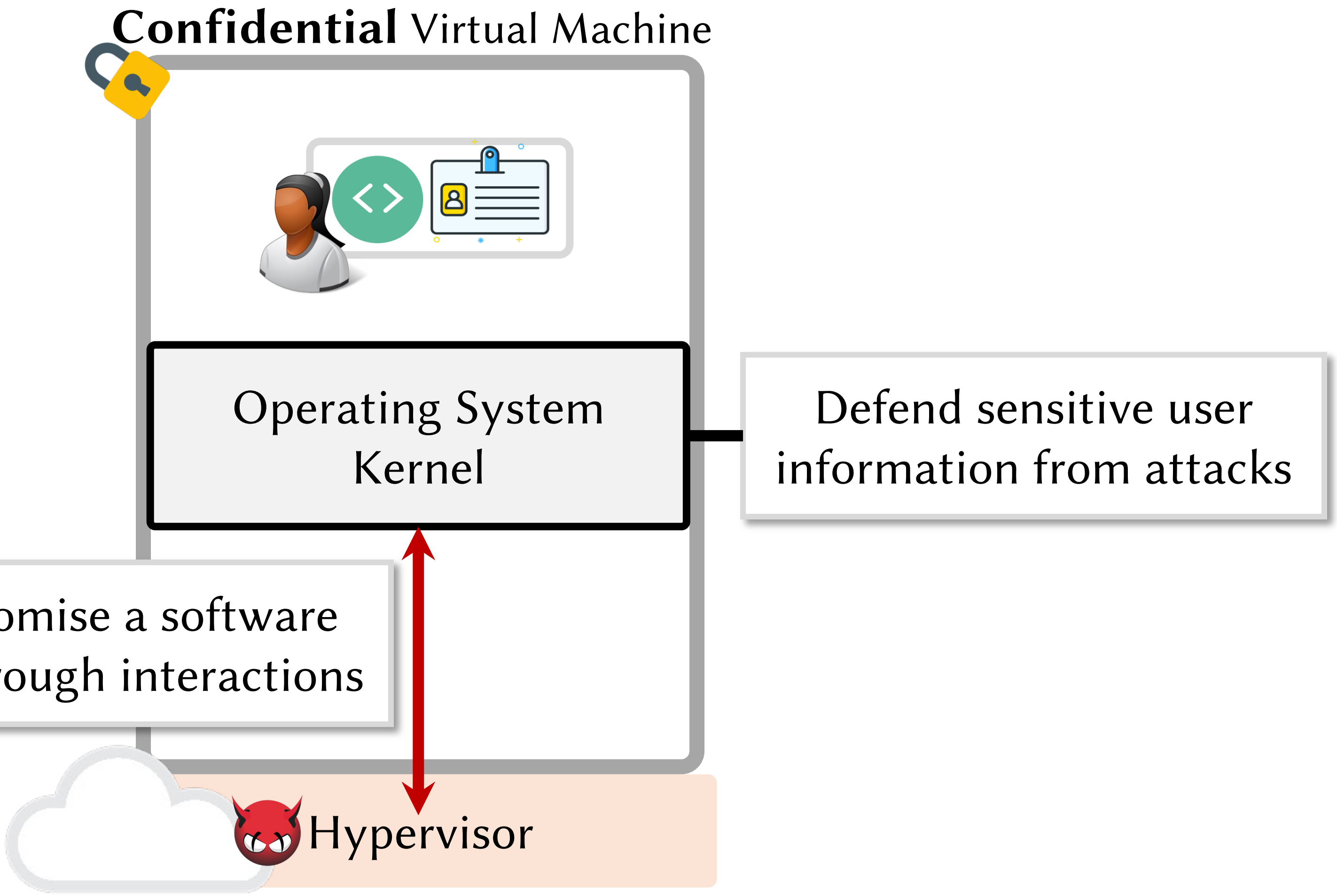
Cloud CVM deployments

Untrusted tenants and system admins

Hypervisor

AMD Secure Encrypted Virtualization (**SEV**)

CVMs rely on OS for defense against remaining attacks



Confidential Virtual Machine

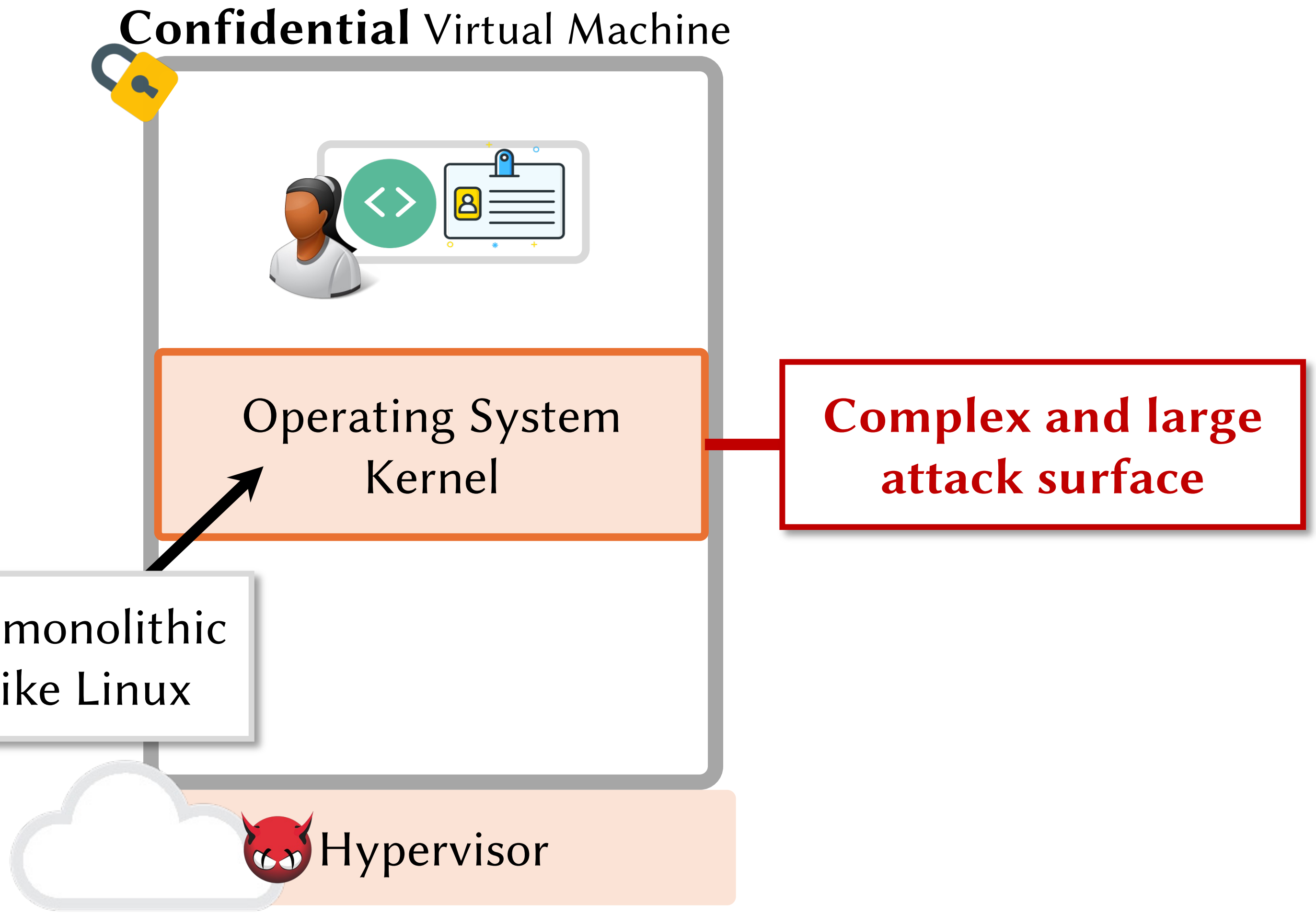
Operating System Kernel

Defend sensitive user information from attacks

Try to compromise a software component through interactions

Hypervisor

Assumption of trust on the CVM OS is misplaced



Confidential Virtual Machine

Operating System Kernel

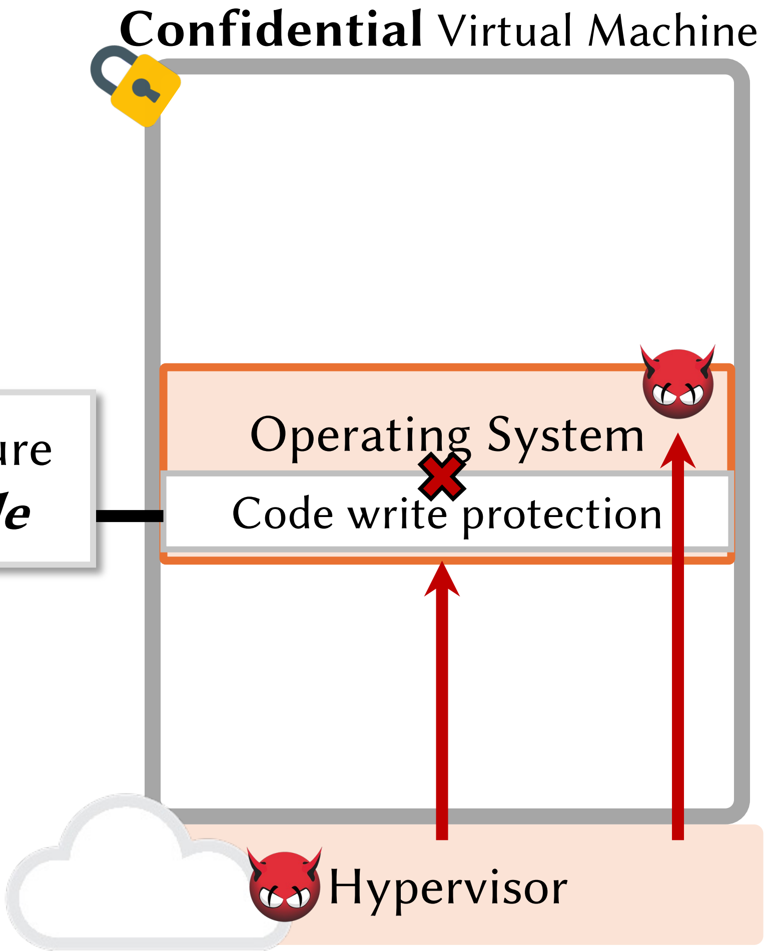
Complex and large attack surface

Typically run monolithic OS kernels like Linux

Hypervisor

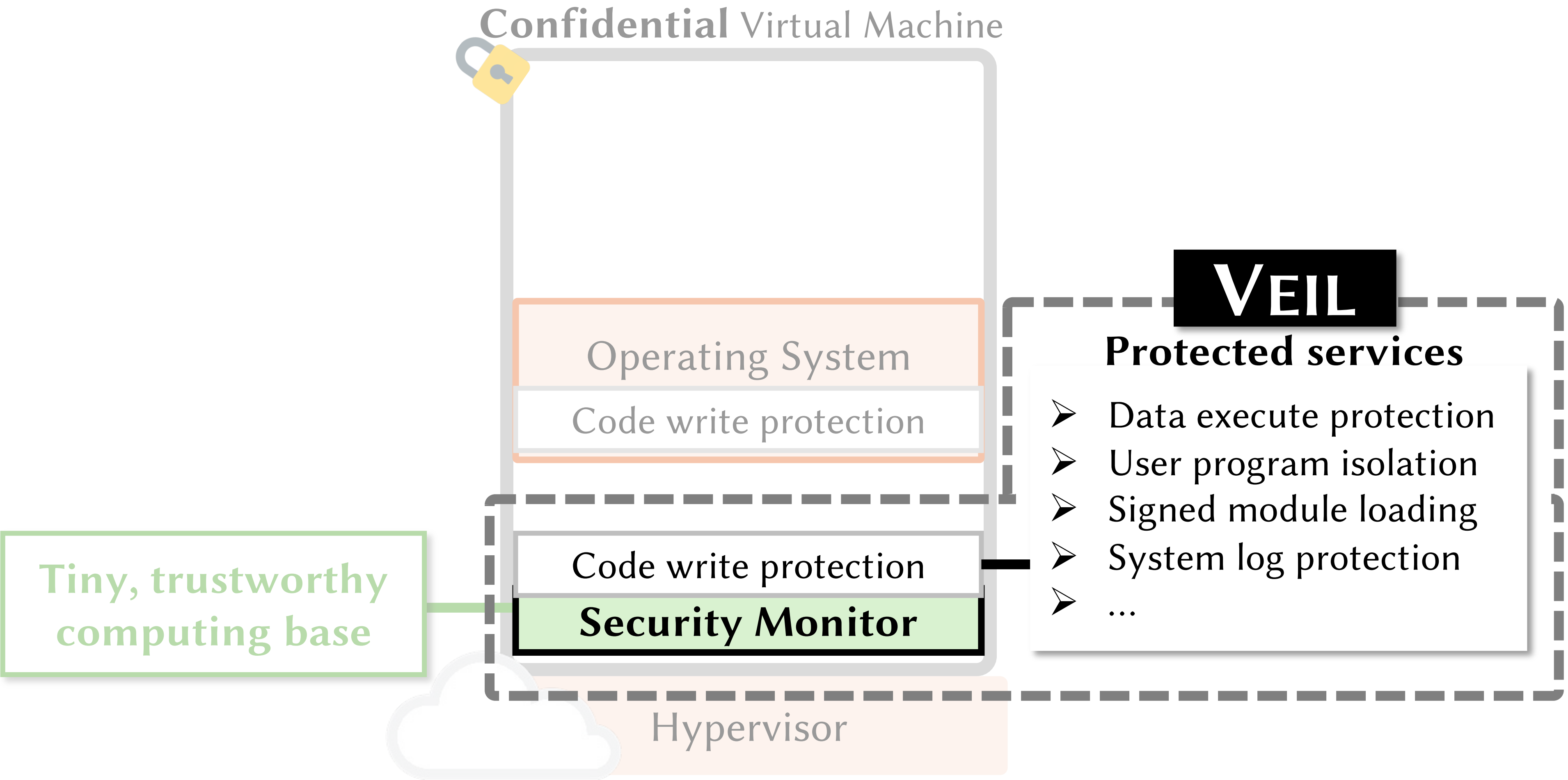
Kernel code integrity as an *example* of misplaced trust

Uses page tables to ensure its code is ***not writable***



Modify kernel code and break integrity assumption

How can we solve the problem of misplaced trust?



Protected services

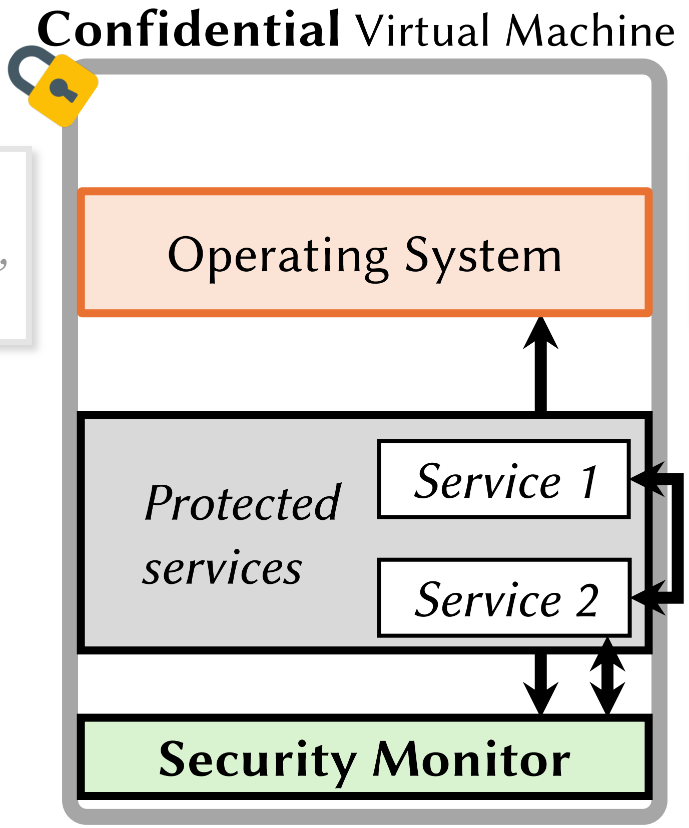
- ➢ Data execute protection
- ➢ User program isolation
- ➢ Signed module loading
- ➢ System log protection
- ➢ …

Two requirements for VEIL's monitor and protected services

☐ **Versatile permissions**

➢ Services rely on *read*, *write*, *or execute* restrictions

E.g., code write protection

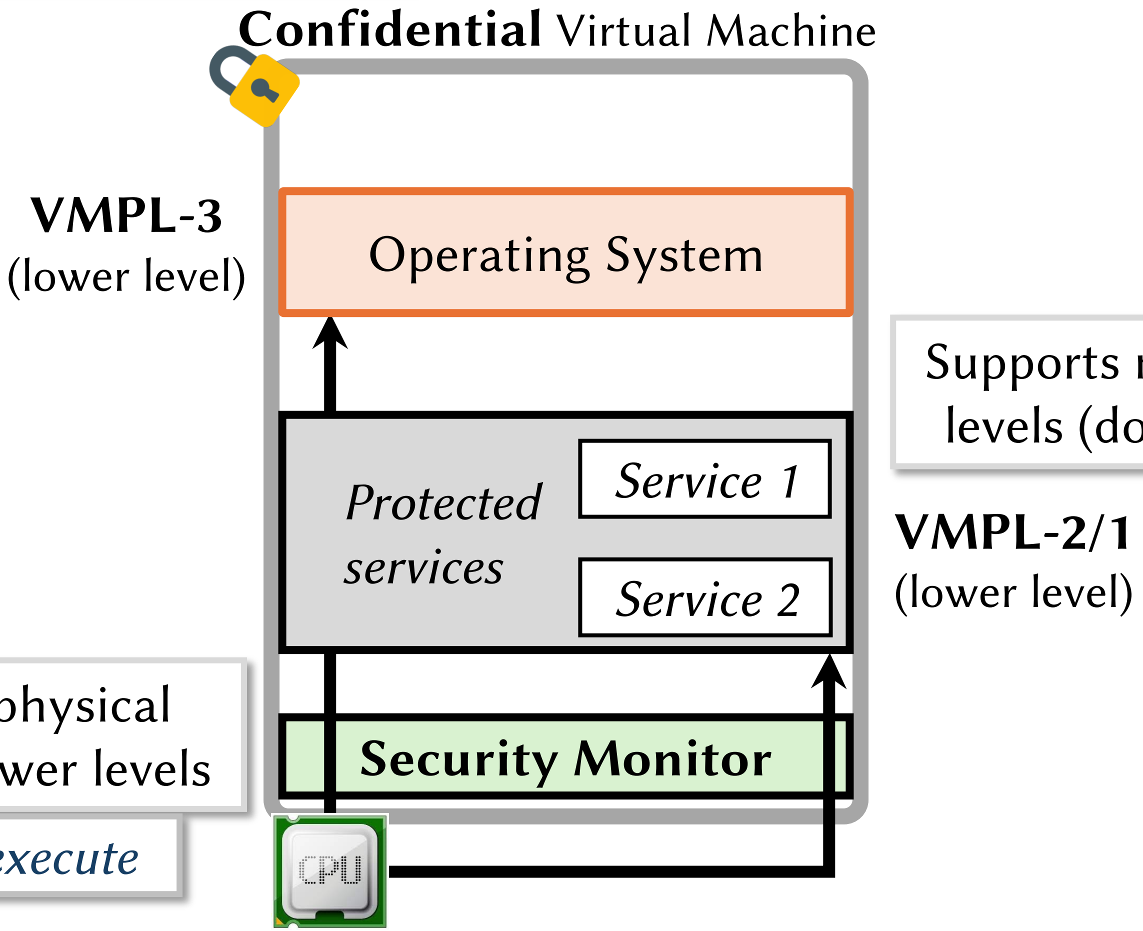


☐ **Multi-domain isolation**

➢ Minimize risk by isolating all components from each other

VEIL leverages Virtual Machine Privilege Levels (VMPL)

- Available in AMD SEV-SNP servers

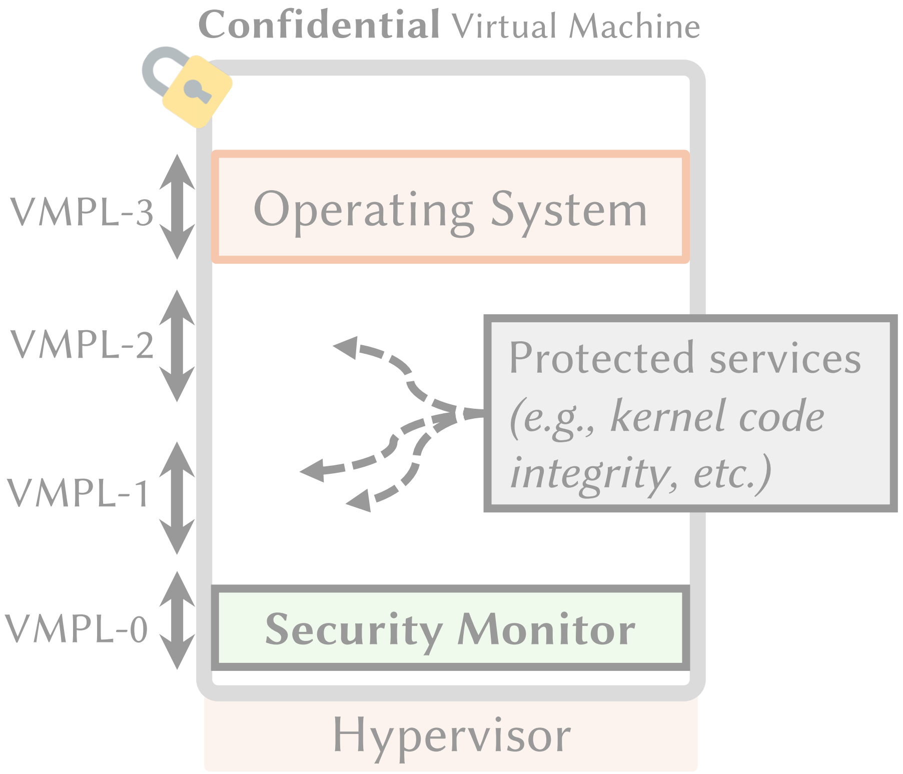


Supports multiple levels (domains)

VMPL-0 → restrict physical page permissions of lower levels

restrict *read*, *write*, *execute*

What are the challenges in using VMPL for VEIL?



Confidential Virtual Machine

VMPL-3

Operating System

VMPL-2

Protected services *(e.g., kernel code integrity, etc.)*

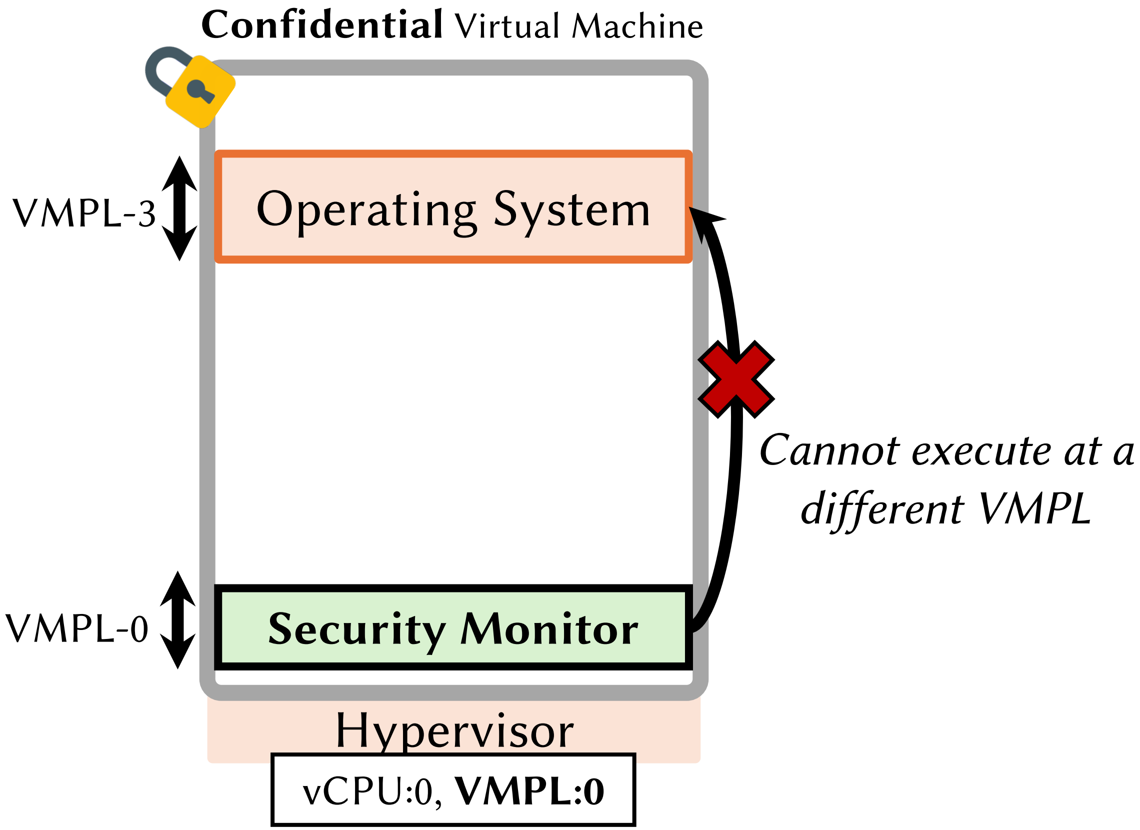
VMPL-1

VMPL-0

Security Monitor

Hypervisor

4 domains cannot isolate N>2 protected services



Confidential Virtual Machine

VMPL-3

Operating System

VMPL-0

Security Monitor

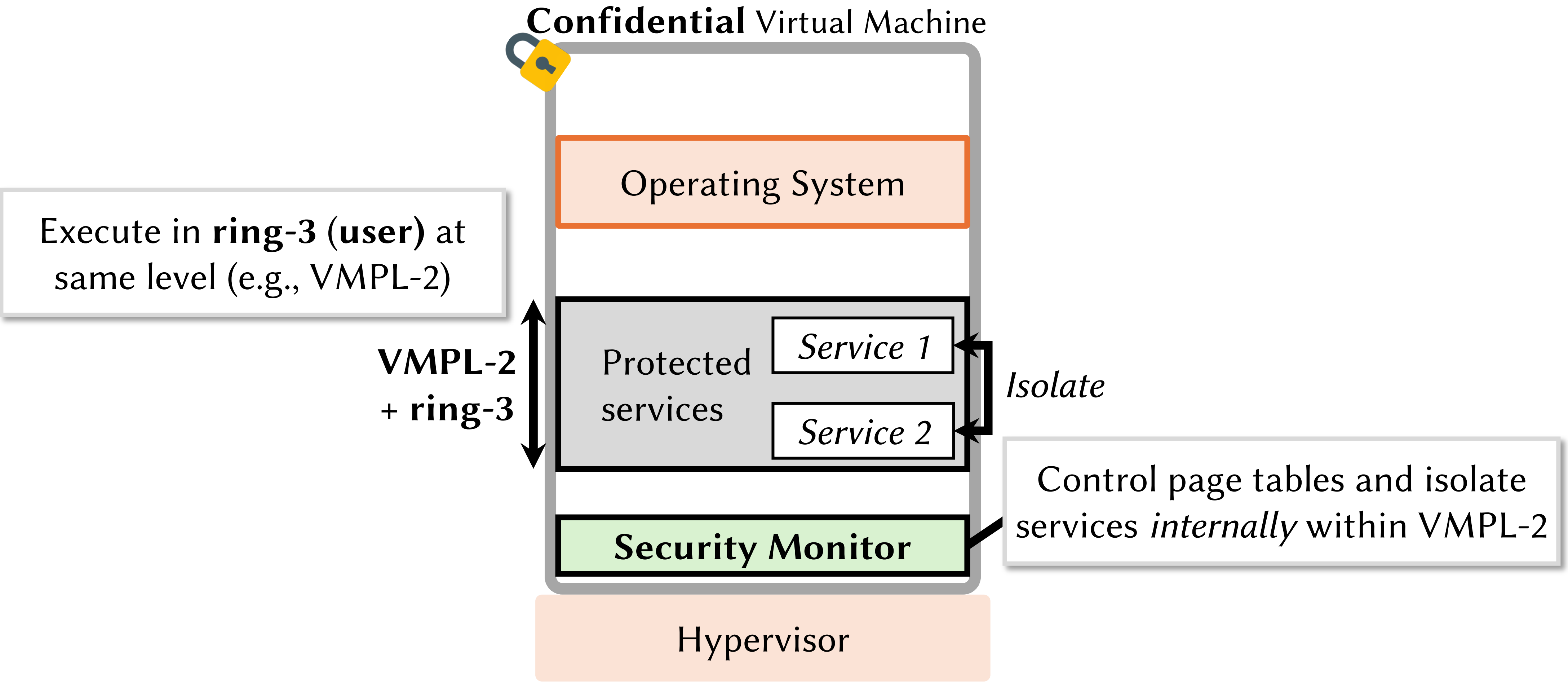
Cannot execute at a different VMPL

Hypervisor

vCPU:0, **VMPL:0**

Reserved CPUs at each domain wastes resources

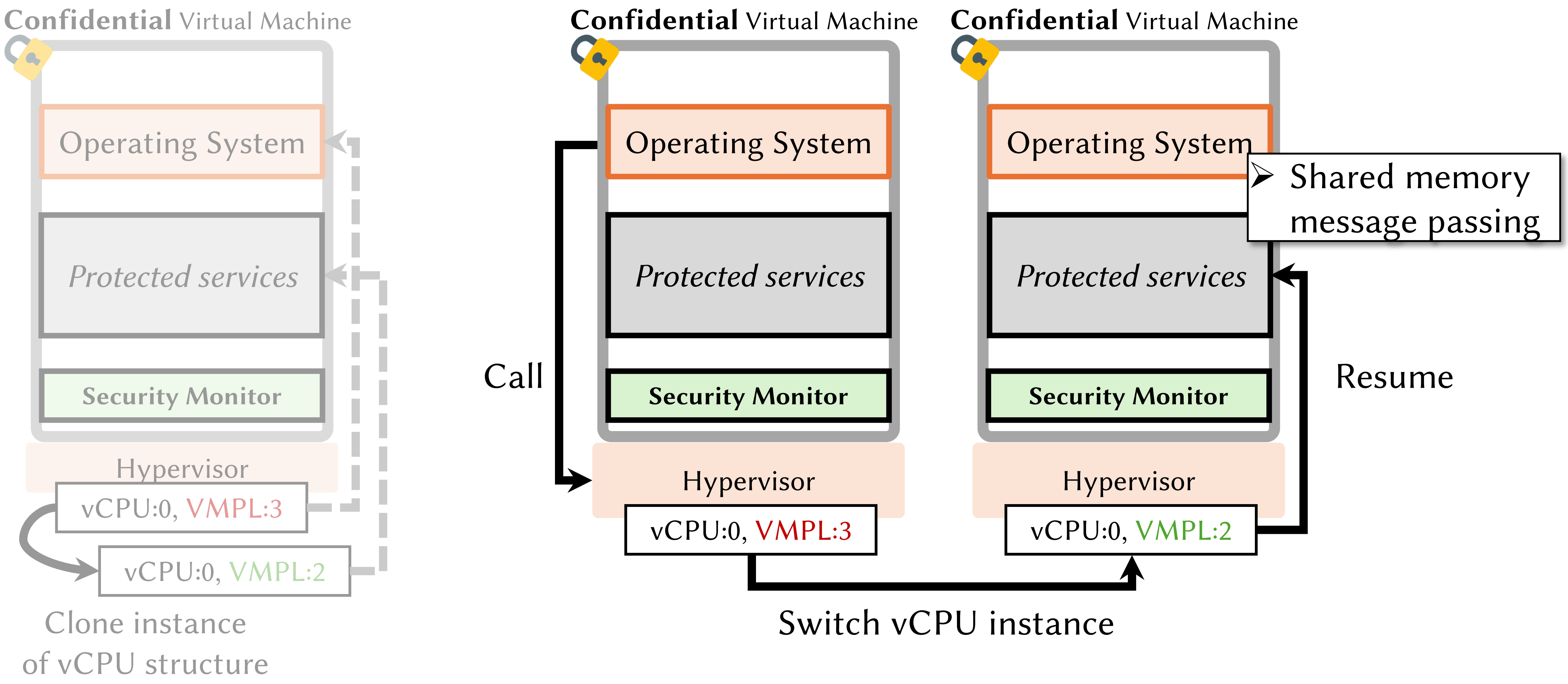
1: Combine VMPL-rings for numerous domains



Execute in **ring-3 (user)** at same level (e.g., VMPL-2)

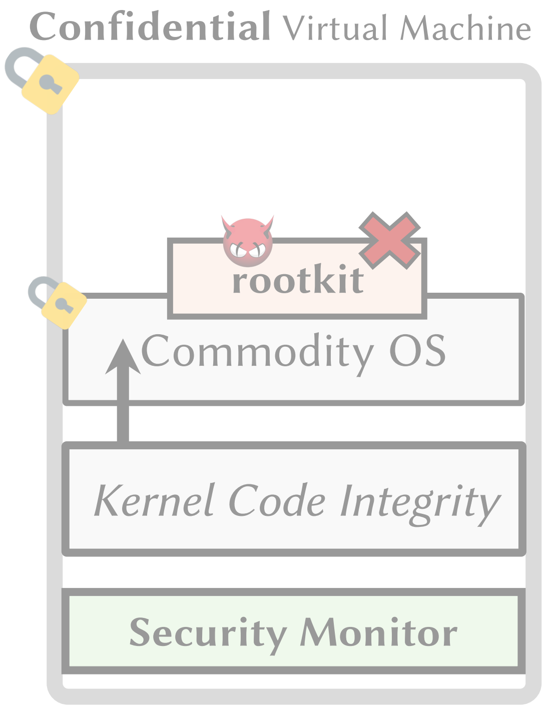
Control page tables and isolate services *internally* within VMPL-2

2: Replicate vCPU instances to avoid reservation



- Shared memory message passing

Veil supports a diverse set of protected services


Confidential Virtual Machine

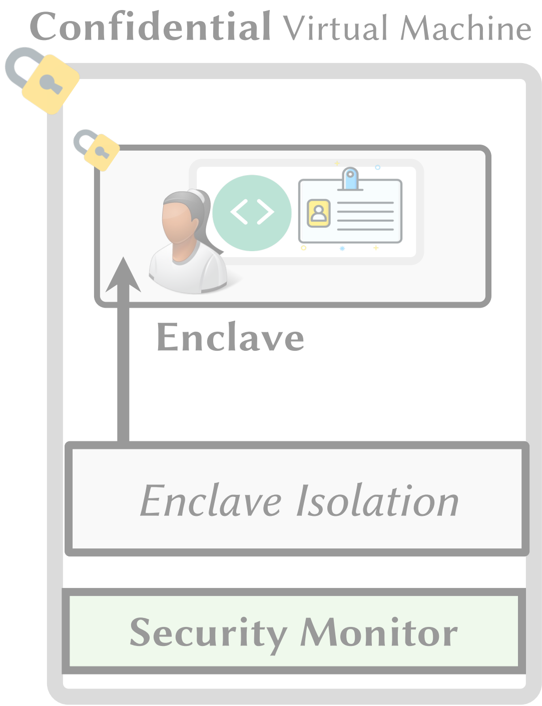
rootkit

Commodity OS

Kernel Code Integrity

Security Monitor

Kernel code integrity
(prevent code injection/rewrite)

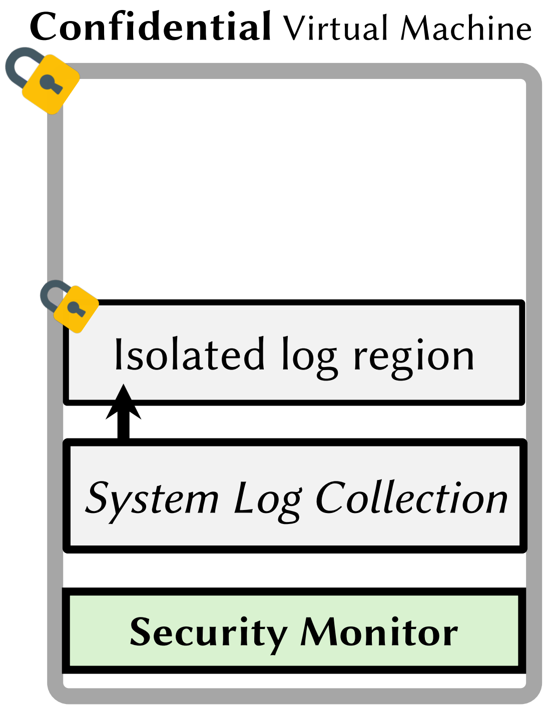

Confidential Virtual Machine

Enclave

Enclave Isolation

Security Monitor

Enclave isolation
(Isolate user programs from OS)


Confidential Virtual Machine

Isolated log region

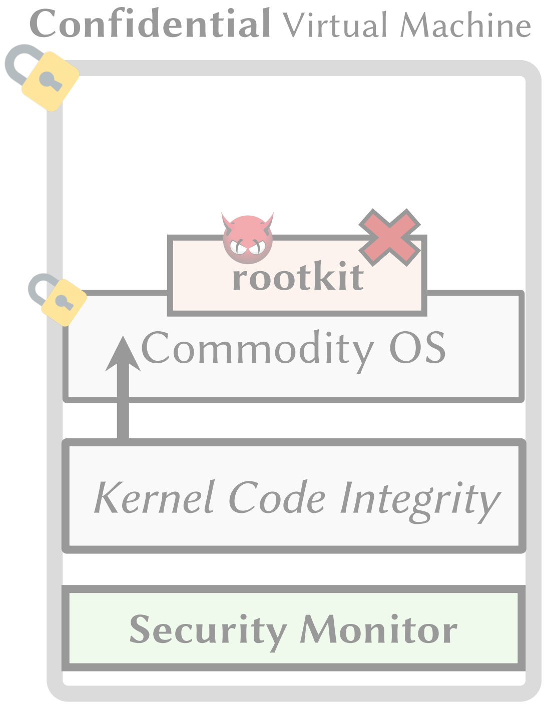
System Log Collection

Security Monitor

Secure log collection
(store logs in protected regions)

VEIL supports a diverse set of protected services

➢ Will briefly present today

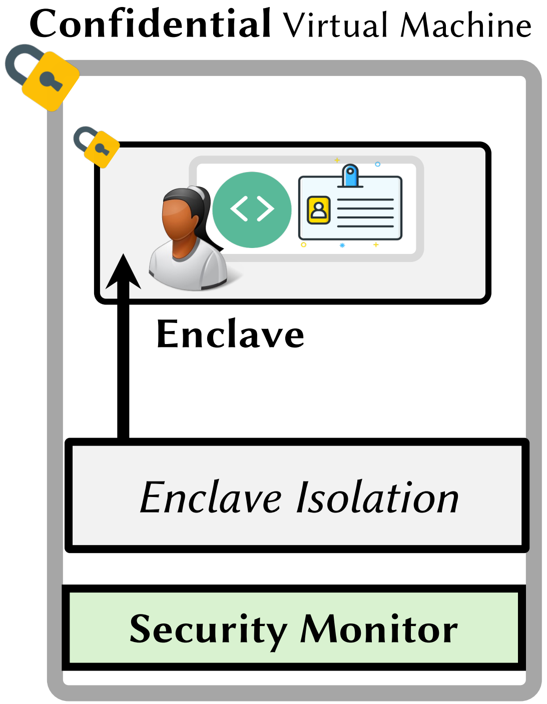

Confidential Virtual Machine

rootkit

Commodity OS

Kernel Code Integrity

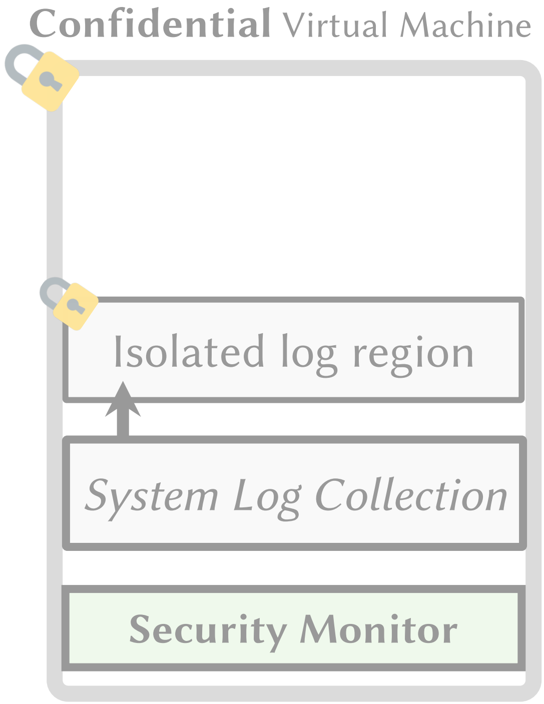
Security Monitor


Confidential Virtual Machine

Enclave

Enclave Isolation

Security Monitor


Confidential Virtual Machine

Isolated log region

System Log Collection

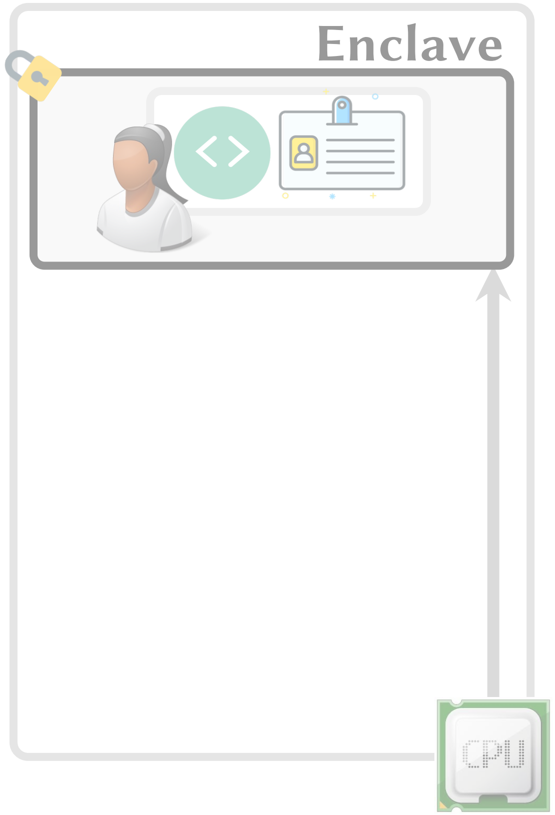
Security Monitor

Kernel code integrity
(prevent code injection/rewrite)

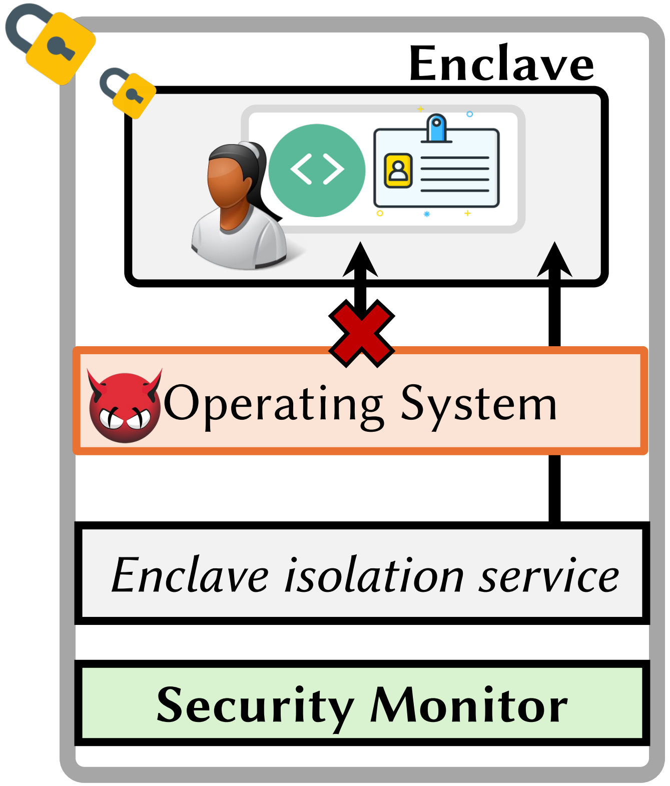
Enclave isolation
(Isolate user programs from OS)

Secure log collection:
(store logs in protected regions)

Realize SGX-like enclaves using VEIL in CVMs

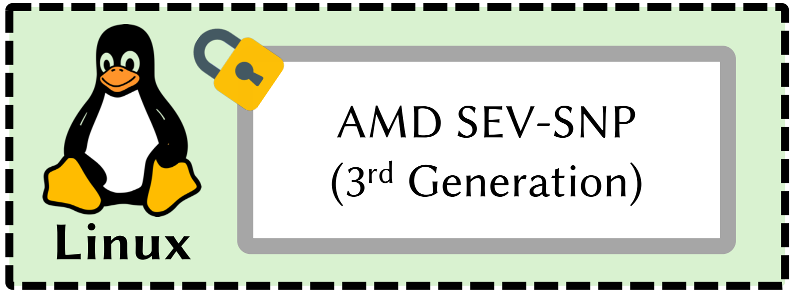


Intel Software Guard eXtensions (**SGX**)



- Memory protect. using VMPL
- Software page table protection
- System call sanitization

What is VEIL’s runtime performance overhead?

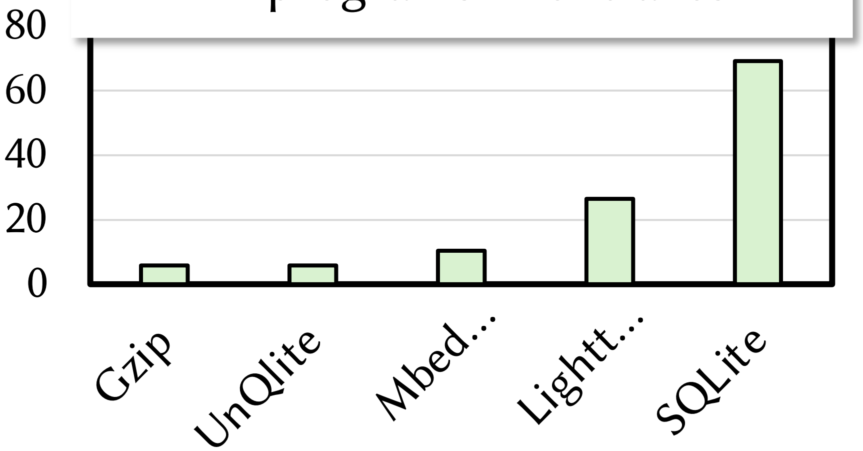


Linux

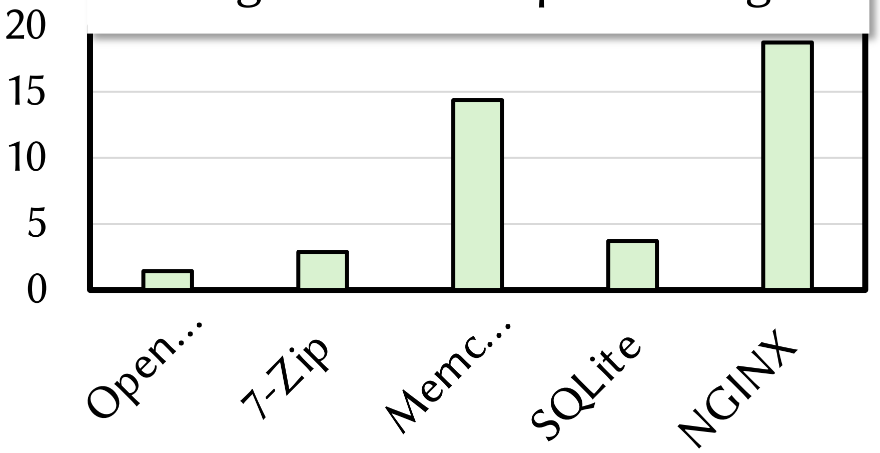
AMD SEV-SNP
(3rd Generation)

✓ 4 – 6% kernel module loading and unloading increase time

✓ 5 – 69% overhead to isolate programs in enclaves



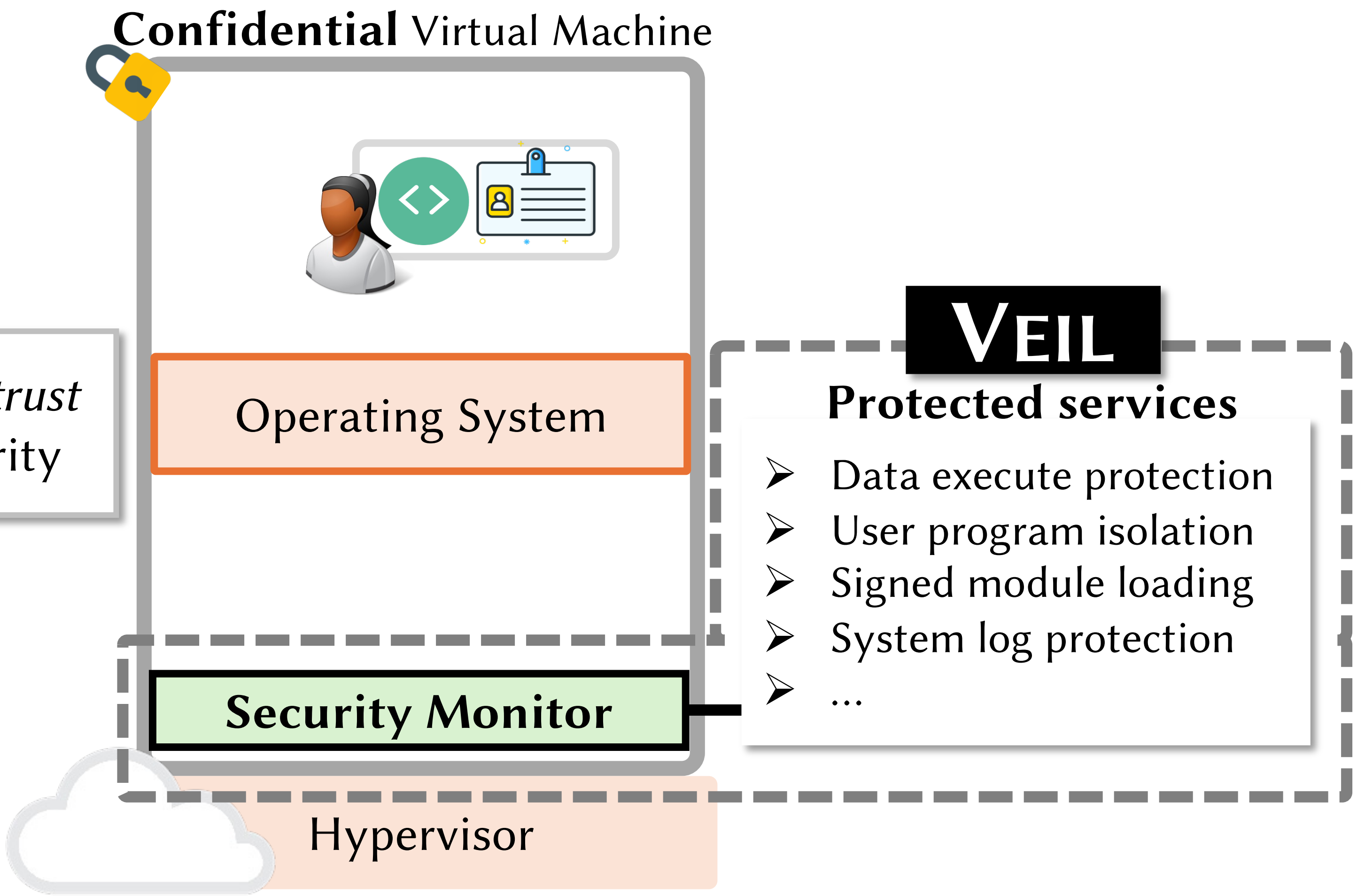
✓ 1 – 19% total overhead to generate and protect logs



➢ Modest overhead for many real-world scenarios

Conclusion

CVMs require a rethink of *trust assignment* for strong security



Confidential Virtual Machine

Operating System

Security Monitor

Hypervisor

VEIL

Protected services

- Data execute protection
- User program isolation
- Signed module loading
- System log protection
- …

https://github.com/adilahmad17/Veil