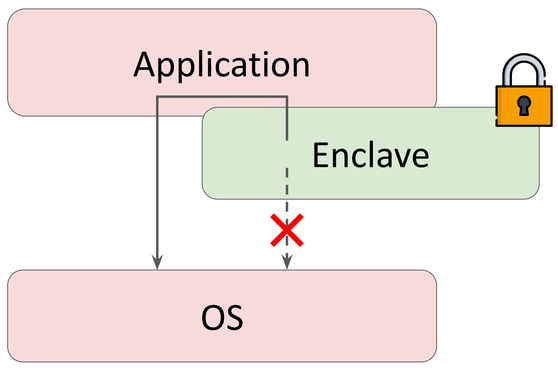
Toward Process-Level TEEs with OS Compatibility and a Minimal TCB

Guojun Wu Keisuke Iida Satoru Takekoshi Takahiro Shinagawa
The University of Tokyo
{wu,iida,takekoshi}@os.is.s.u-tokyo.ac.jp shina@is.s.u-tokyo.ac.jp

Background: Problems of Existing TEEs

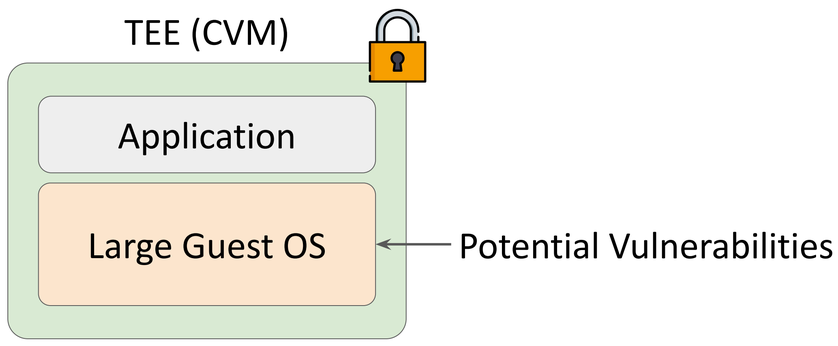
Enclave-based TEEs (Intel SGX)

- ✓ Offer strong isolation with a small TCB
- ✗ Strict code constraints cause compatibility issues
  e.g., Enclaves cannot invoke system calls directly



VM-based TEEs (AMD SEV, Intel TDX)

- ✓ Allow unmodified applications to run inside a TEE
- ✗ Traditional guest OSes lead to bloated TCB size
  Their broad attack surface has been exploited

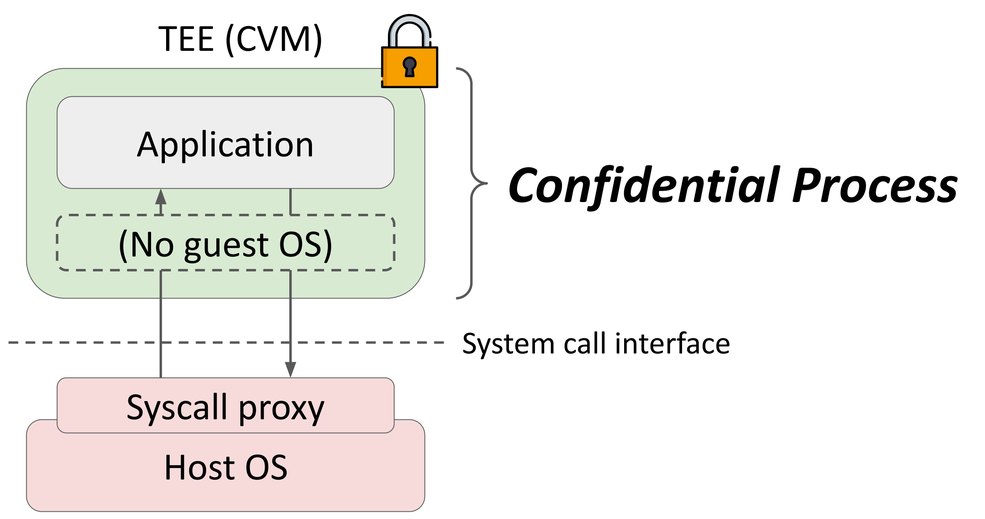


Proposal: Confidential Process — A New TEE Abstraction

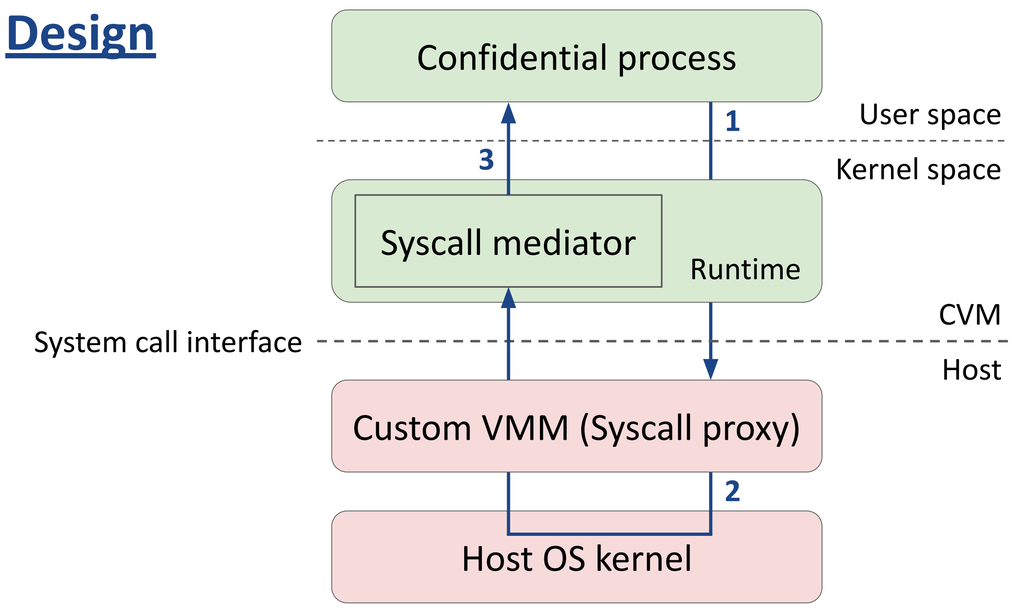
Goal **Achieve both compatibility and TCB minimization**

Approach **Run a single user-level process inside a CVM**

- ✓ Process-only TCB, eliminating guest OS stacks
- ✓ Preserve compatibility by reusing system call interface for host OS interaction



Design



System call handling flow for confidential processes

1. **Runtime** writes system call arguments to shared pages
   - Shared pages are encrypted memory regions accessible to host
   - Data in I/O buffer are also written to shared pages
2. **Custom VMM** forwords system call to host kernel and writes back result
3. **System call mediator** validates response and copies data into memory process
   - Process memory always resides in encrypted private pages

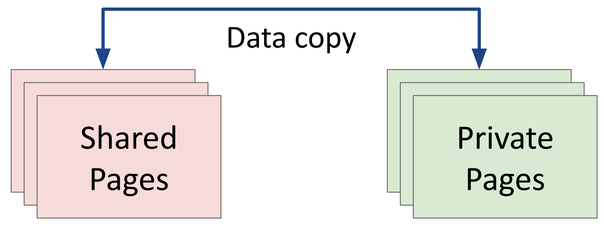
Prototype Implementation

- Implemented confidential processes on **AMD SEV-SNP**
  - Using **KVM** to create a custom VMM
- Successfully run unmodified, simple Linux ELF programs
  - At startup, VMM loads program and runtime into CVM
  - Basic I/O system calls are supported
  - Memory management system calls are directly handled within CVM
- Next step:
  - Adjust shared pages dynamically to accommodate large I/O
  - Add multi-thread support

Future Extensions

Content-based Encryption Avoidance

- Copying between shared and private pages is a bottleneck
- No encryption needed for data originating outside CVM
- Apply copy-on-write to avoid redundant encryption



Further reduce TCB

- Memory management code in runtime enlarges TCB
- Decouple privileged and non-privileged components
  - Run non-privileged component in a less privileged VMPL