Scotch: Combining Software Guard Extensions and System Management Mode to Monitor Cloud Resource Usage

Kevin Leach[1], **Fengwei Zhang**[2], and Westley Weimer[1]

[1]University of Michigan, [2]Wayne State University

Summary

We use **Intel Software Guard Extensions (SGX)** and **System Management Mode (SMM)** to accurately monitor **resource consumption** of virtual machines (VMs) **in the presence of a compromised VM or hypervisor**

Motivation

- Multi-tenant cloud computing increases utilization
- Client agrees to pay Cloud provider for a particular service level
  - e.g., $1 per hour of CPU time

Motivation

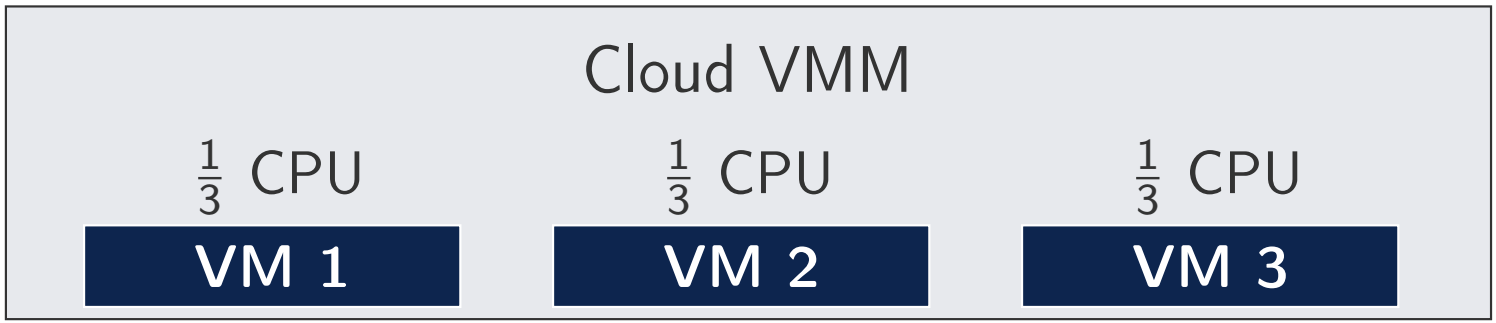
- Multi-tenant cloud computing increases utilization
- Client agrees to pay Cloud provider for a particular service level
  - e.g., $1 per hour of CPU time

- **Cloud provider depends on hypervisor/virtual machine monitor (VMM) platform to distribute resources**
  - Xen, QEMU, etc.

Motivation

- Multi-tenant cloud computing increases utilization
- Client agrees to pay Cloud provider for a particular service level
  - e.g., \$1 per hour of CPU time

- **Cloud provider depends on hypervisor/virtual machine monitor (VMM) platform to distribute resources**
  - Xen, QEMU, etc.



If all 3 VMs peg the CPU, the VMM must decide how to allocate CPU time based on each client's service level.

Motivation

- Cloud provider depends on VMM platform to distribute resources

Motivation

- Cloud provider depends on VMM platform to distribute resources
- Two issues

Motivation

- Cloud provider depends on VMM platform to distribute resources
- Two issues

1. What if the VMM/cloud provider is malicious?
   - Manipulate resource consumption to bill customers more

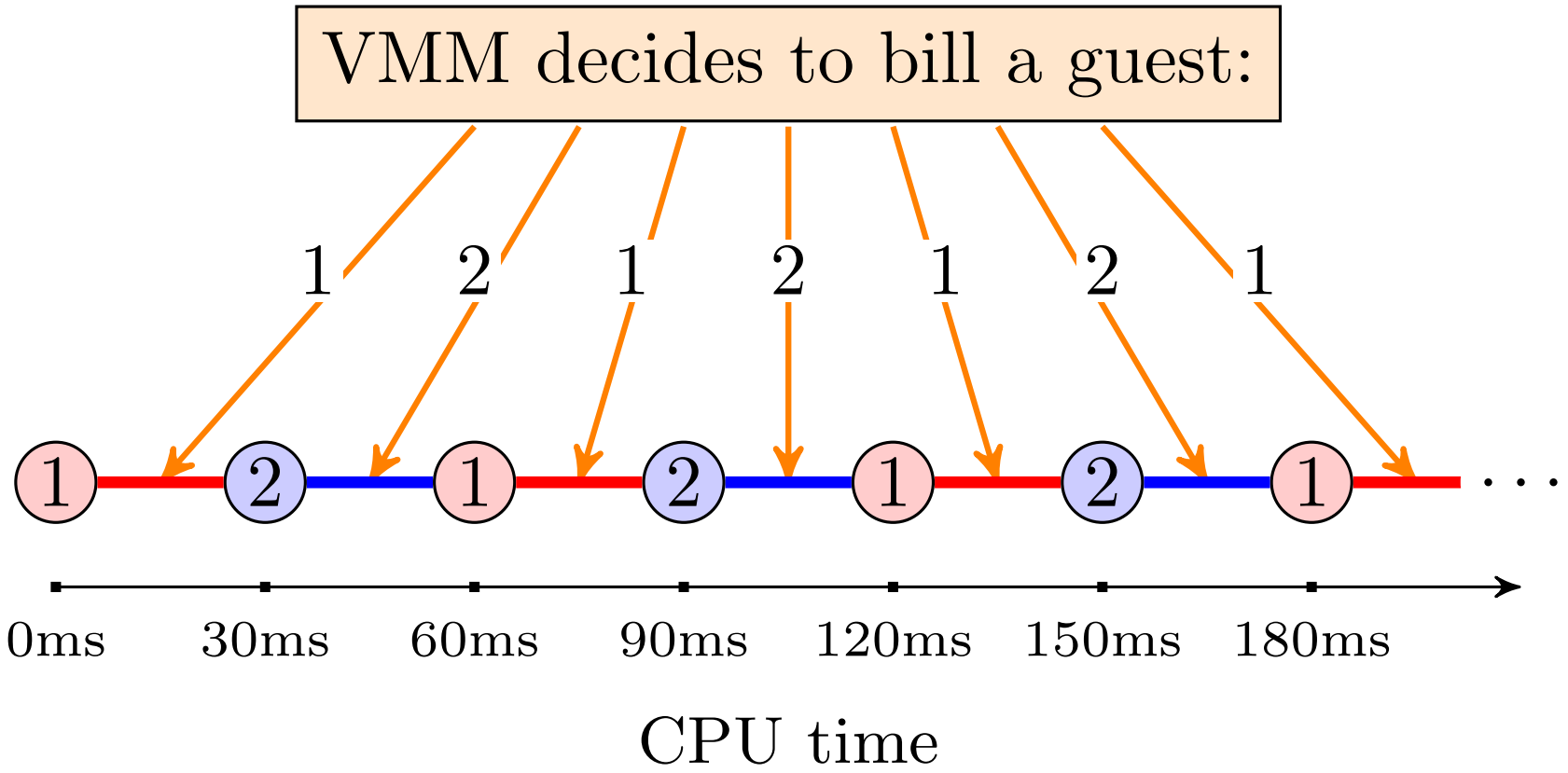
Motivation

- Cloud provider depends on VMM platform to distribute resources
- Two issues

1. What if the VMM/cloud provider is malicious?
   - Manipulate resource consumption to bill customers more
2. What if the VMM is vulnerable to malicious VMs?
   - Malicious VM manipulates resource consumption to steal resources from benign customers

Resource Accounting Attacks

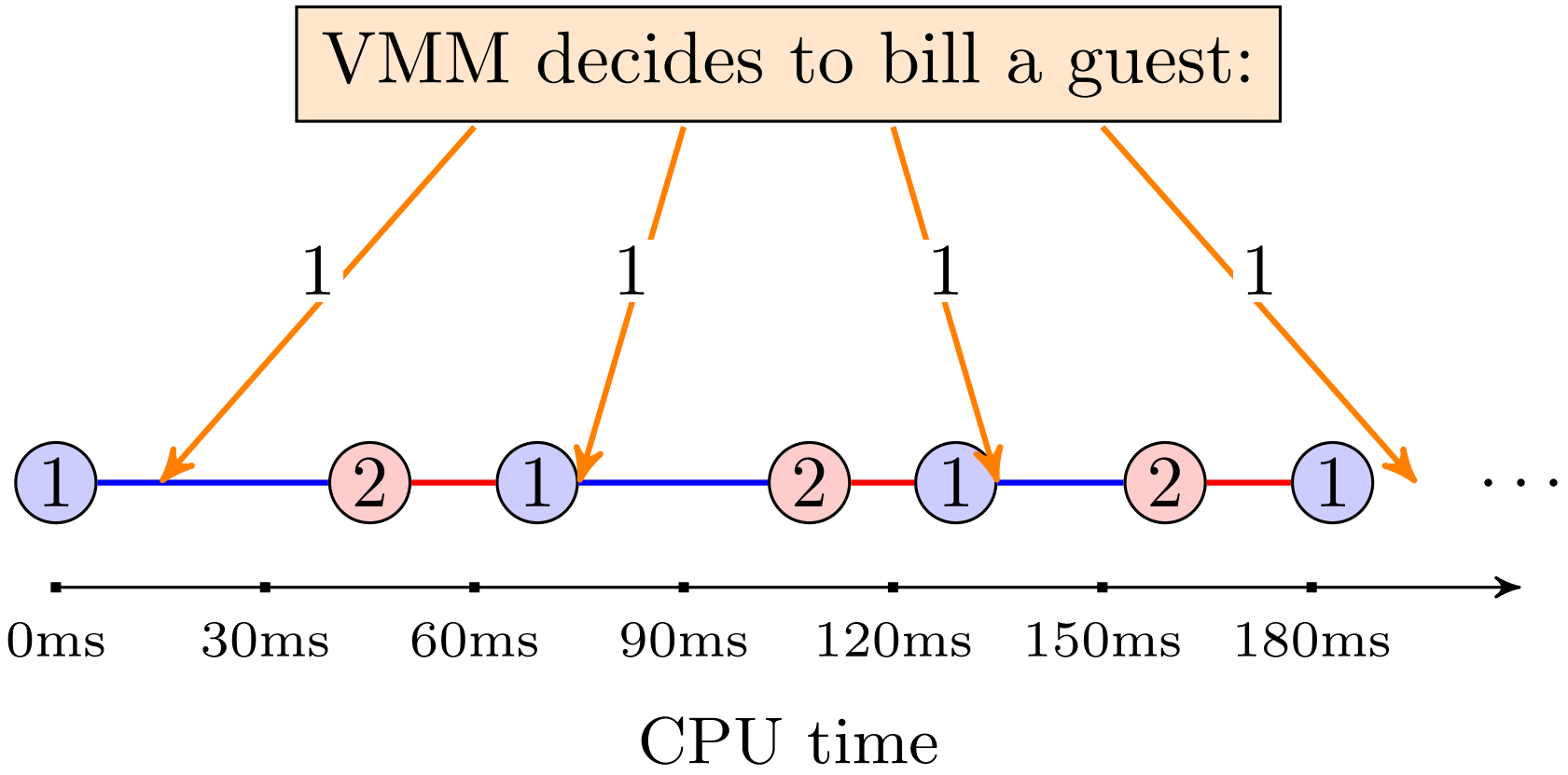
- Benign Behavior



The Xen hypervisor regularly checks which VM is active to determine how much CPU time each VM uses

Resource Accounting Attacks

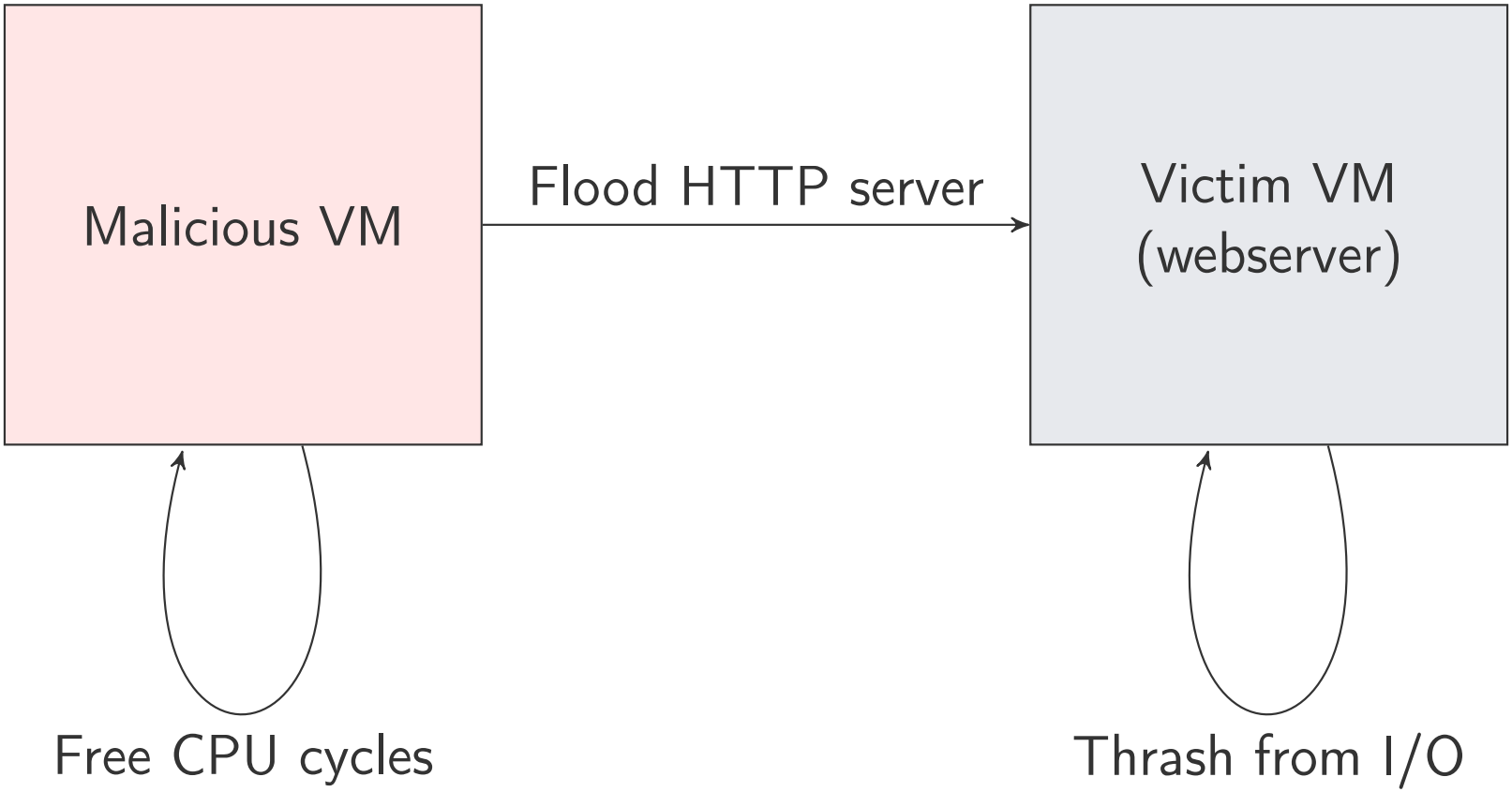
- Malicious Behavior



A malicious VM (2) with knowledge of the VMM can affect the appearance of resource consumption by itself and benign VMs.

Resource Interference Attacks

Attacker can take advantage of known victim behavior



Malicious VM can cause benign VM to free up resources for itself

VM Escape Attack

Malicious VM can exploit buggy VMM implementation, allowing code execution with VMM privilege

- Could potentially alter resource consumption to hide itself

Scotch: Transparent Cloud Resource Accounting

Two desired properties

1. **Transparent**
   - The underlying VMM and VMs are not aware accounting occurs
2. **Tamper-resistant**
   - A malicious VMM or VM guest cannot reliably alter accounting data

Insights for Scotch

- System Management Mode
  High priority System Management Interrupt causes CPU to atomically execute SMM handler code

Insights for Scotch

- System Management Mode
  High priority System Management Interrupt causes CPU to atomically execute SMM handler code
- Use SMM to collect raw resource consumption data

Insights for Scotch

- System Management Mode
  High priority System Management Interrupt causes CPU to atomically execute SMM handler code
- Use SMM to collect raw resource consumption data
- SMM logically collects data, then relays it to SGX enclave

Insights for Scotch

- Software Guard Extensions
  Enclave-based trusted execution environment (TEE); userspace code runs in isolation

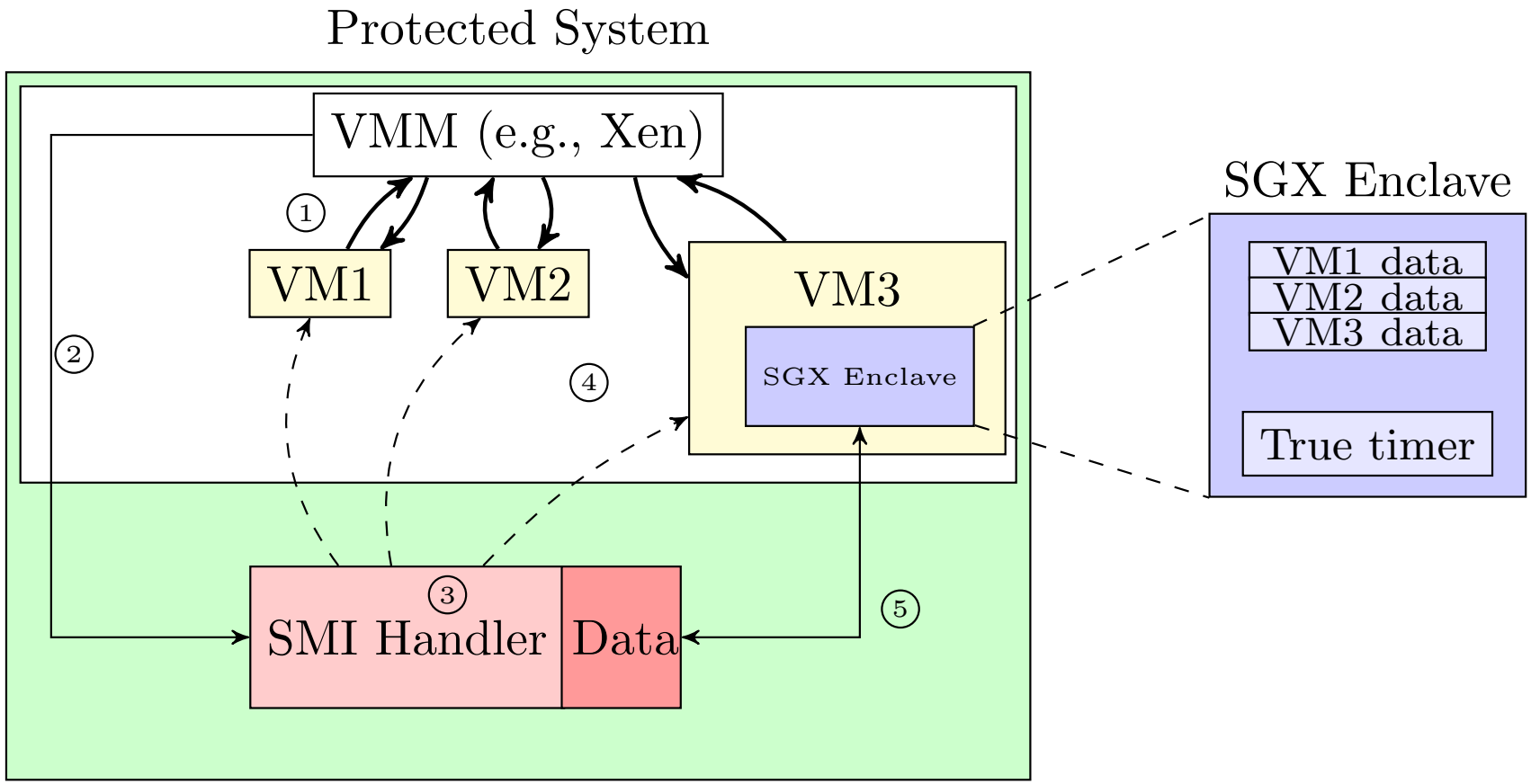
Insights for Scotch

- Software Guard Extensions
  Enclave-based trusted execution environment (TEE); userspace code runs in isolation
- Use SGX enclave so that benign user can monitor and verify their resource consumption

Insights for Scotch

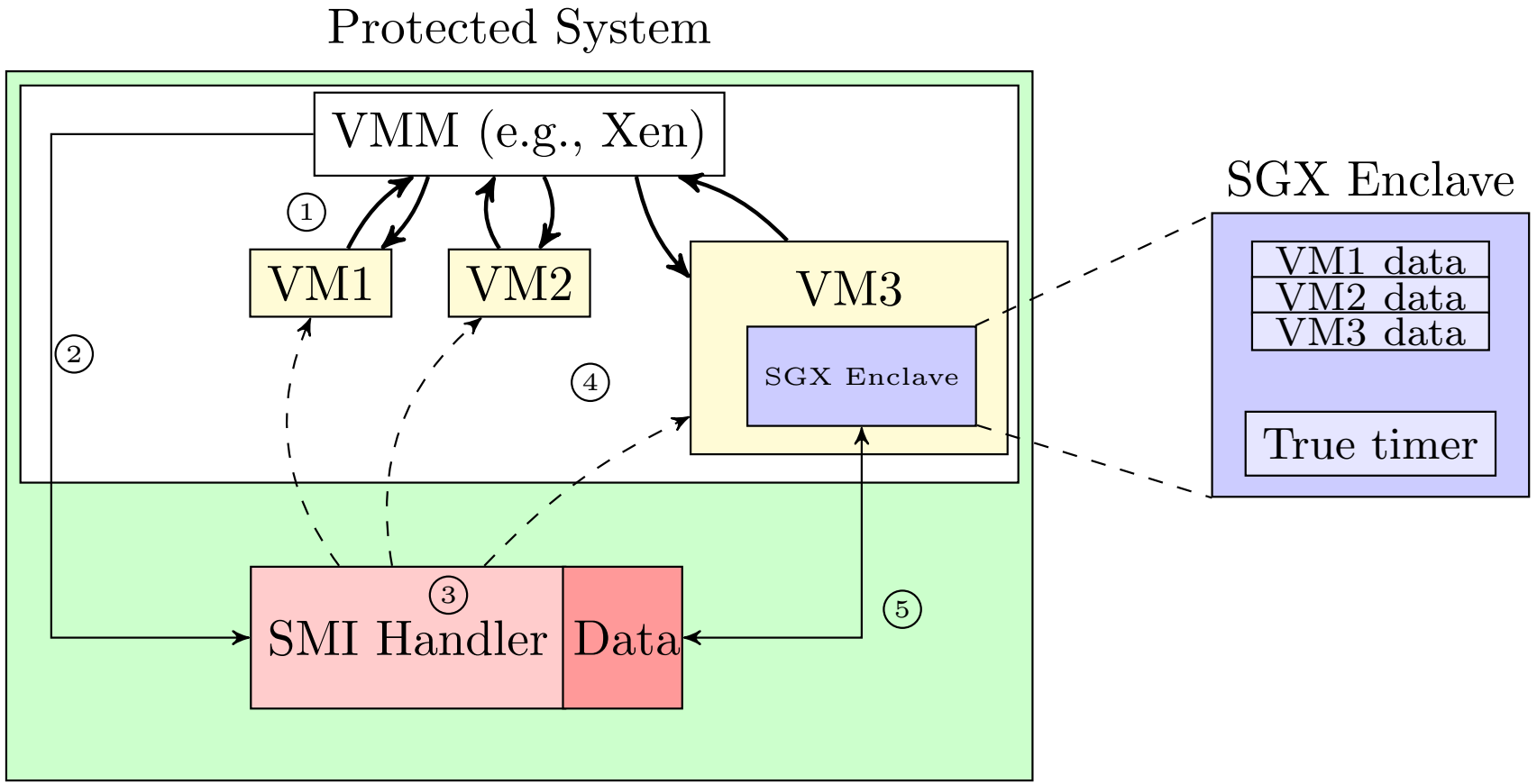
- Software Guard Extensions
  Enclave-based trusted execution environment (TEE); userspace code runs in isolation
- Use SGX enclave so that benign user can monitor and verify their resource consumption
- Raw data collected by SMM is relayed to SGX enclave

Scotch: Transparent Cloud Resource Accounting



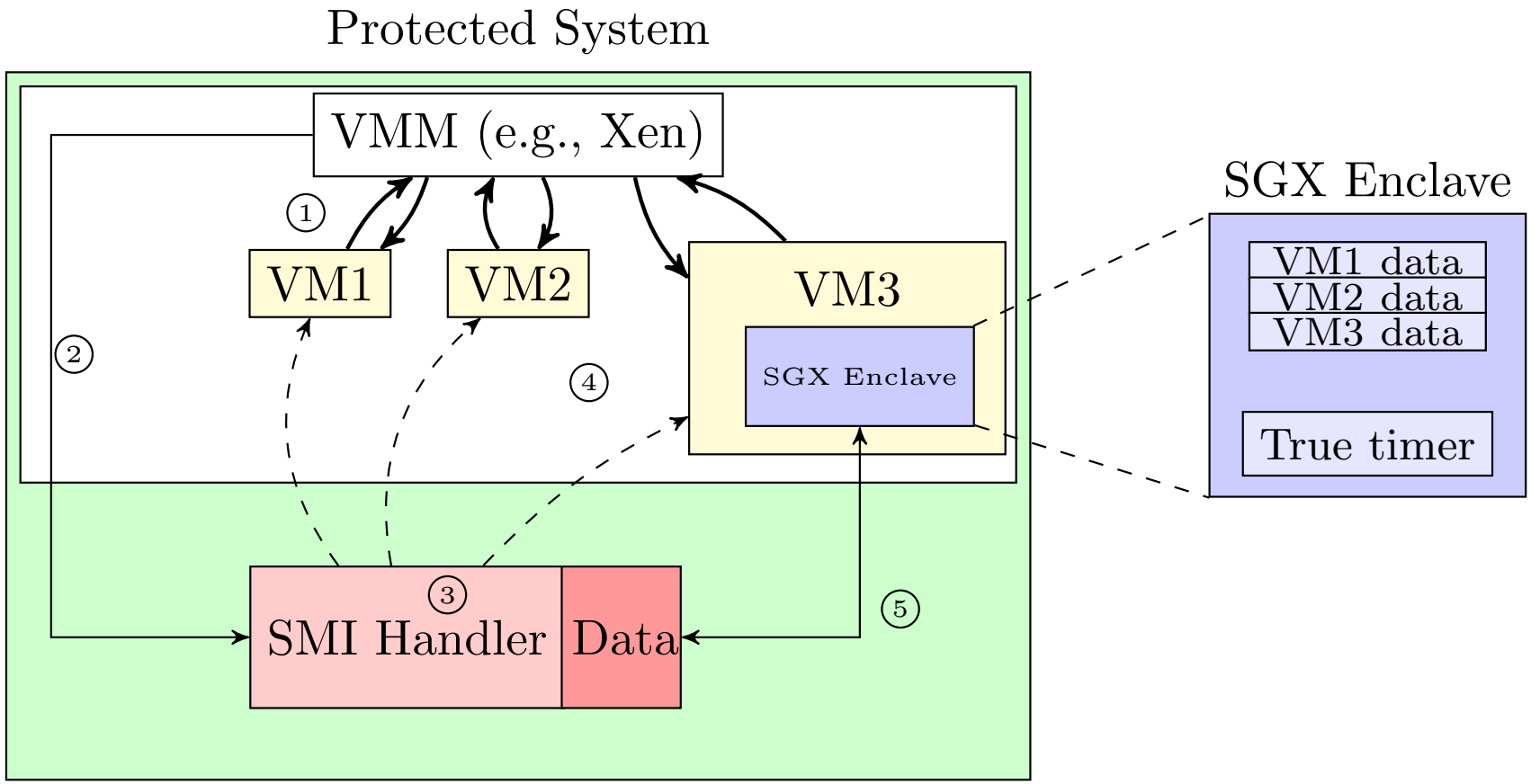
1. VMM decides to switch between VM guests

Scotch: Transparent Cloud Resource Accounting



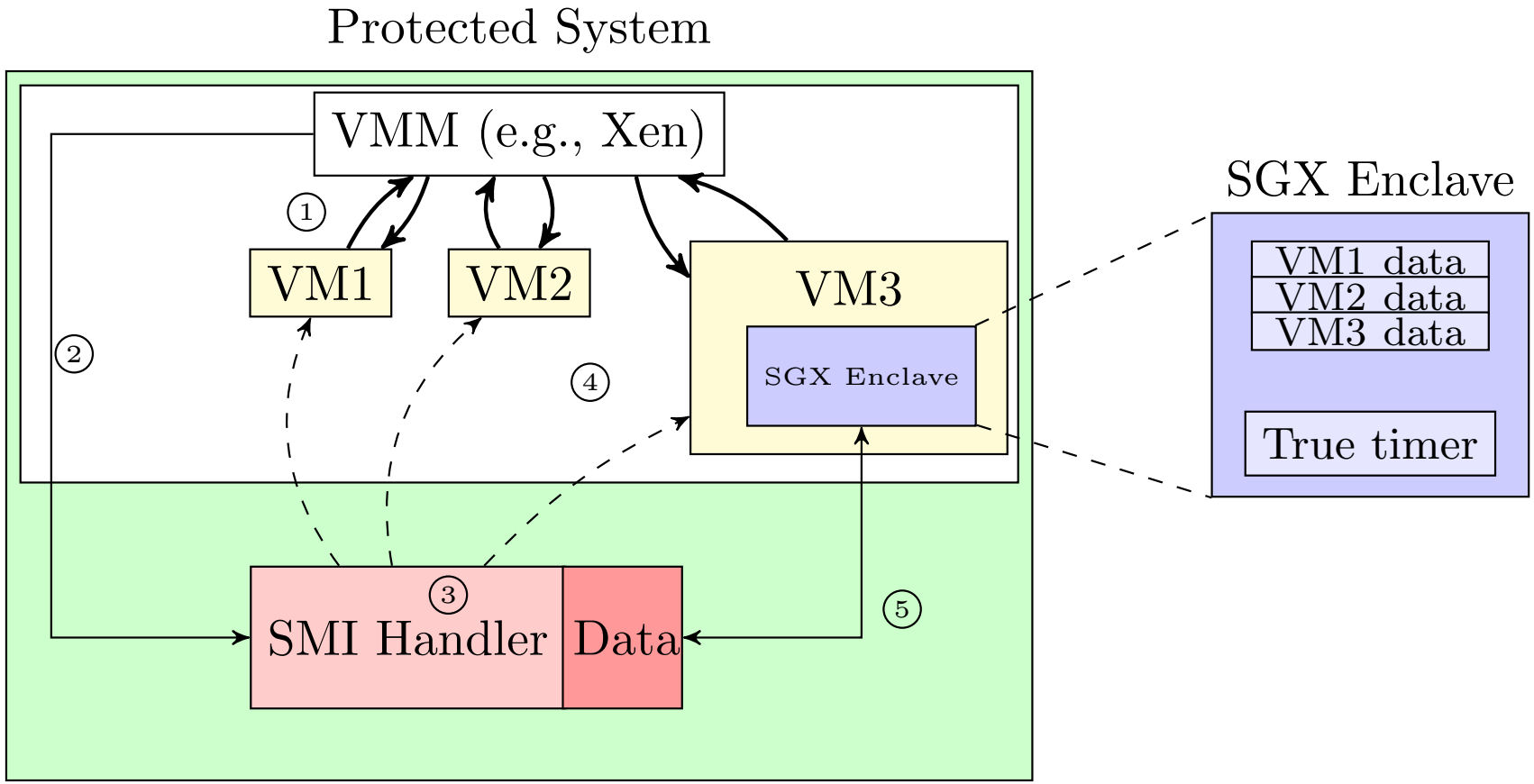
2. Scotch measures resource consumption by invoking SMM every context switch

Scotch: Transparent Cloud Resource Accounting



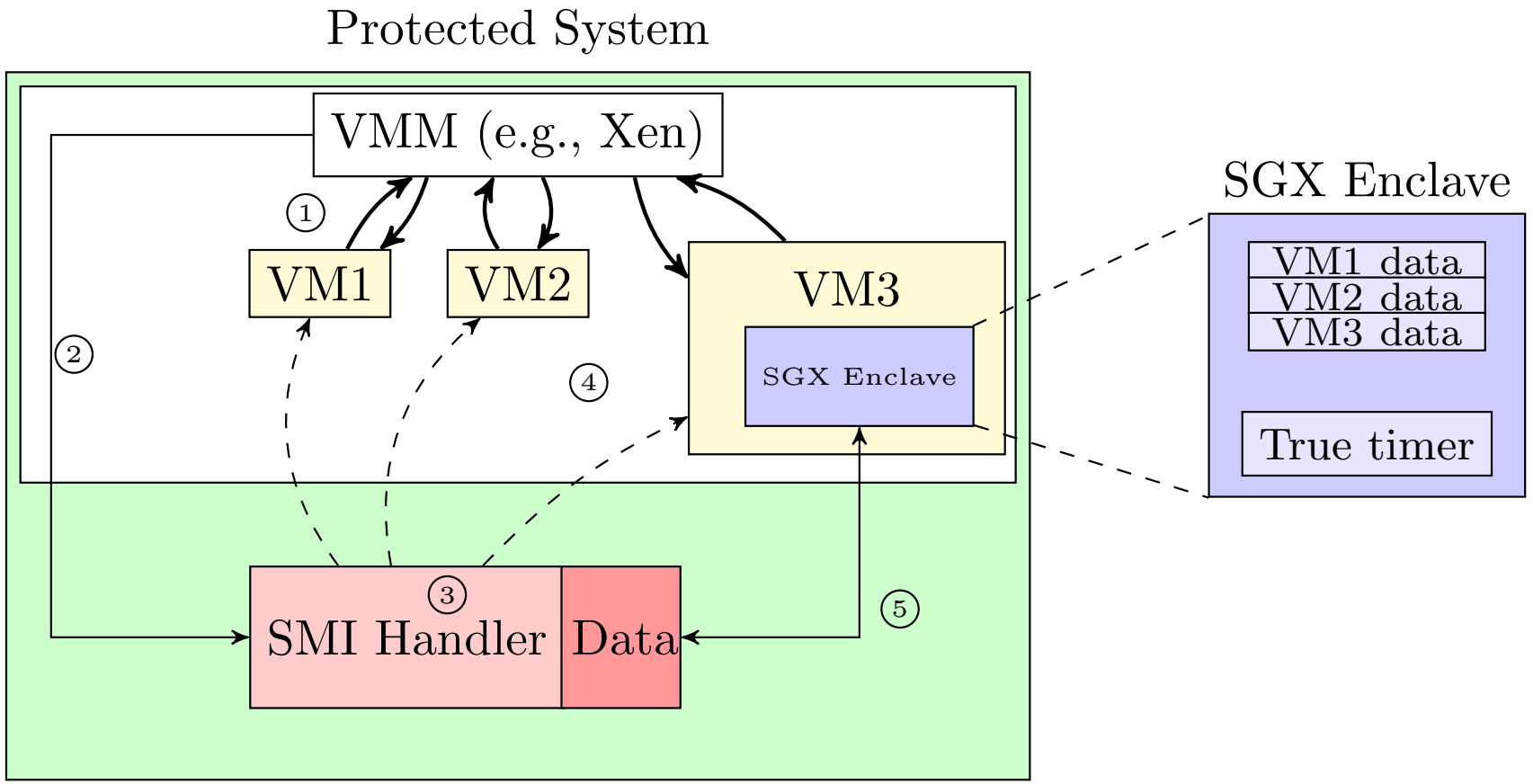
3. SMM handler executes resource accounting in isolation

Scotch: Transparent Cloud Resource Accounting



4. Data is marshalled to SGX enclave within VM

Scotch: Transparent Cloud Resource Accounting



5. Benign VM can monitor resource accounting data with high integrity

Evaluation

Research Questions

- RQ1: Can we maintain accurate accounting during scheduler attacks?
- RQ2: What is our overhead on benign workloads?
- RQ3: Can we maintain accurate accounting during resource interference attacks?
- RQ4: Can we maintain accurate accounting during VM escape attacks?

RQ1: Scheduler Attacks

- Implement controllable scheduler
  - Simulate attacker by altering the CPU time allocation by a varying degree

RQ1: Scheduler Attacks

- Implement controllable scheduler
  - Simulate attacker by altering the CPU time allocation by a varying degree
- Run two VMs, one simulated attacker and one benign
  - Both are computing indicative workloads: `pi`, `gzip`, and the PARSEC benchmarks

RQ1: Scheduler Attacks

- Implement controllable scheduler
  - Simulate attacker by altering the CPU time allocation by a varying degree
- Run two VMs, one simulated attacker and one benign
  - Both are computing indicative workloads: `pi`, `gzip`, and the PARSEC benchmarks
- Compare observed CPU time consumption presented by Xen vs. Scotch

RQ1: Scheduler Attacks

- Implement controllable scheduler
  - Simulate attacker by altering the CPU time allocation by a varying degree
- Run two VMs, one simulated attacker and one benign
  - Both are computing indicative workloads: `pi`, `gzip`, and the PARSEC benchmarks
- Compare observed CPU time consumption presented by Xen vs. Scotch
- **TL;DR** Scotch shows significant difference in allocated CPU time

RQ1: Scheduler Attacks

Table : Ratio of attacker VM CPU time to guest VM CPU time.

	Scheduler attack severity level						
	Benign	1	3	5	7	9	10
Scotch	1.00	1.04	1.10	1.17	1.26	1.36	1.41
ground truth	0.99	1.05	1.12	1.17	1.25	1.35	1.39

RQ1: Scheduler Attacks

Table : Ratio of attacker VM CPU time to guest VM CPU time.

	Scheduler attack severity level						
	Benign	1	3	5	7	9	10
Scotch	1.00	1.04	1.10	1.17	1.26	1.36	1.41
ground truth	0.99	1.05	1.12	1.17	1.25	1.35	1.39

The attacker receives disproportionate CPU time. Ground truth obtained with Xentrace.

RQ2: Overhead

- Invoking SMIs to run accounting code can be costly

RQ2: Overhead

- Invoking SMIs to run accounting code can be costly
- Accounting code takes 2248 ± 69 cycles to execute
  - Roughly 1μs incurred every context switch

RQ2: Overhead

- Invoking SMIs to run accounting code can be costly
- Accounting code takes 2248 ± 69 cycles to execute
  - Roughly 1μs incurred every context switch
- Xen takes roughly 20,000 cycles (7.1μs) per context switch

RQ2: Overhead

- Invoking SMIs to run accounting code can be costly
- Accounting code takes 2248 ± 69 cycles to execute
  - Roughly 1μs incurred every context switch
- Xen takes roughly 20,000 cycles (7.1μs) per context switch
- **Scotch adds 14% overhead per context switch**

RQ2: Overhead

- Invoking SMIs to run accounting code can be costly
- Accounting code takes 2248 ± 69 cycles to execute
  - Roughly 1μs incurred every context switch
- Xen takes roughly 20,000 cycles (7.1μs) per context switch
- **Scotch adds 14% overhead per context switch**
- **.0033% system overhead on CPU-bound workloads**

RQ3: Resource Interference Attacks

- By construction, Scotch provides accurate accounting information

RQ3: Resource Interference Attacks

- By construction, Scotch provides accurate accounting information
- Scotch does not automatically detect Resource Interference Attacks

RQ3: Resource Interference Attacks

- By construction, Scotch provides accurate accounting information
- Scotch does not automatically detect Resource Interference Attacks
- However, SGX allows userspace access to reliable accounting information
  - Client can monitor their resource usage and perform their own analysis for their case

RQ4: VM Escape Attacks

- Accounting code stored in isolated SMRAM and SGX enclave

RQ4: VM Escape Attacks

- Accounting code stored in isolated SMRAM and SGX enclave
- Even if attacker roots hypervisor, they cannot change the accounting code

RQ4: VM Escape Attacks

- Accounting code stored in isolated SMRAM and SGX enclave
- Even if attacker roots hypervisor, they cannot change the accounting code
- BIOS locks SMRAM, so no opportunity for attacker to infiltrate SMM if BIOS is trusted

Conclusion

- Scotch is an SMM- and SGX-based framework for accurately accounting resource consumption in cloud infrastructure

Conclusion

- Scotch is an SMM- and SGX-based framework for accurately accounting resource consumption in cloud infrastructure
- Scotch accounts for resource usage every context switch, introducing minimal overhead on indicative workloads

Conclusion

- Scotch is an SMM- and SGX-based framework for accurately accounting resource consumption in cloud infrastructure
- Scotch accounts for resource usage every context switch, introducing minimal overhead on indicative workloads
- Scotch accurately accounts for CPU time consumption in the presence of scheduler attack
  - Porting drivers to SMM would readily admit incorporating accounting for additional types of resources, such as network usage

Conclusion

- Scotch is an SMM- and SGX-based framework for accurately accounting resource consumption in cloud infrastructure
- Scotch accounts for resource usage every context switch, introducing minimal overhead on indicative workloads
- Scotch accurately accounts for CPU time consumption in the presence of scheduler attack
  - Porting drivers to SMM would readily admit incorporating accounting for additional types of resources, such as network usage
- By construction, Scotch protects the hypervisor from VM escape and other control hijacking attacks