AEX-Notify: Thwarting Precise Single-Stepping Attacks through Interrupt Awareness for Intel® SGX Enclaves

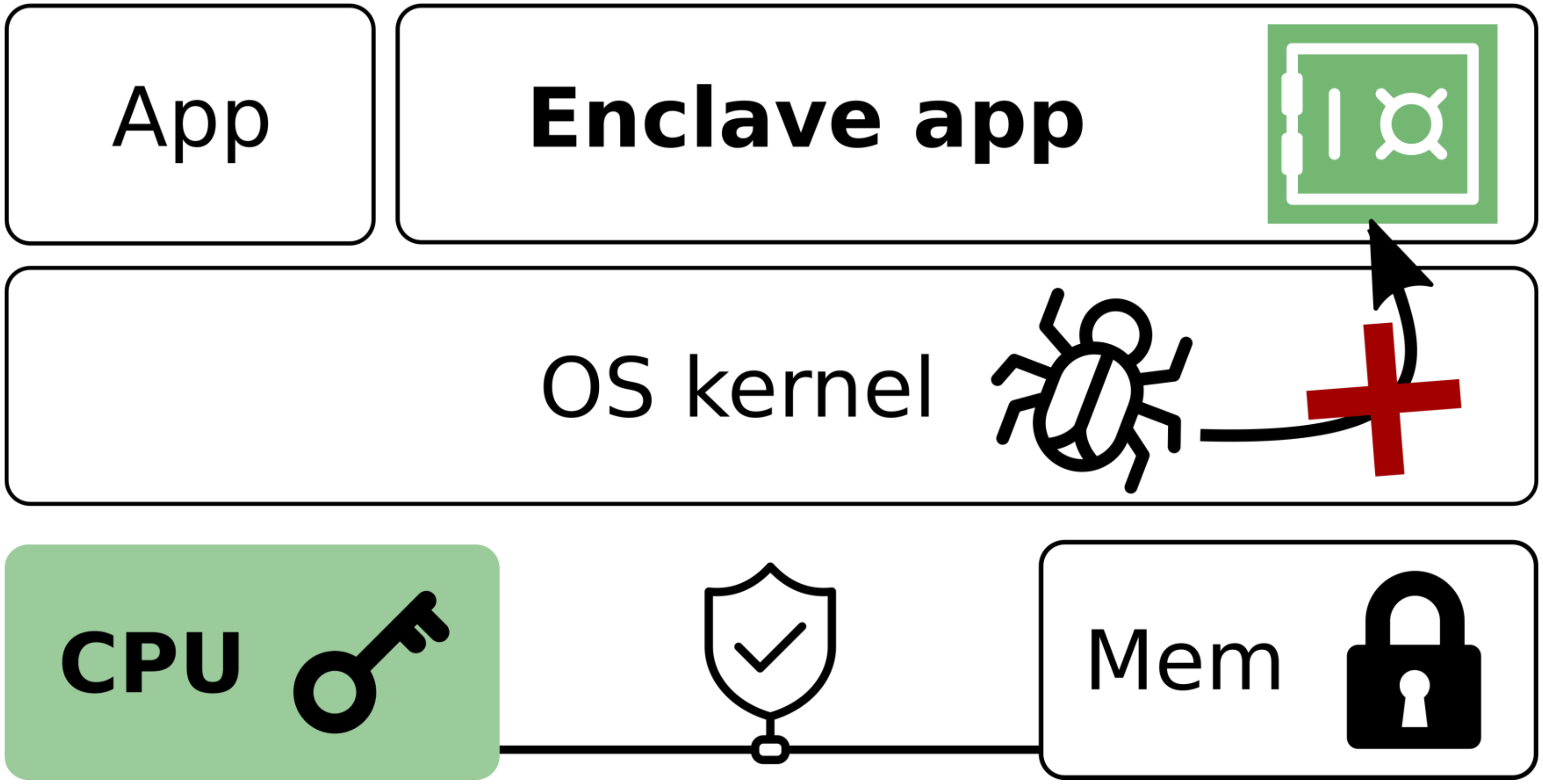
Scott Constable[1], Jo Van Bulck[2], Xiang Cheng[3], Yuan Xiao[1], Cedric Xing[1], Ilya Alexandrovich[1], Taesoo Kim[3], Frank Piessens[2], Mona Vij[1], Mark Silberstein[4]

[1]Intel Corporation [2]imec-DistriNet, KU Leuven [3]Georgia Institute of Technology [4]Technion

32nd USENIX Security Symposium, August 10, 2023

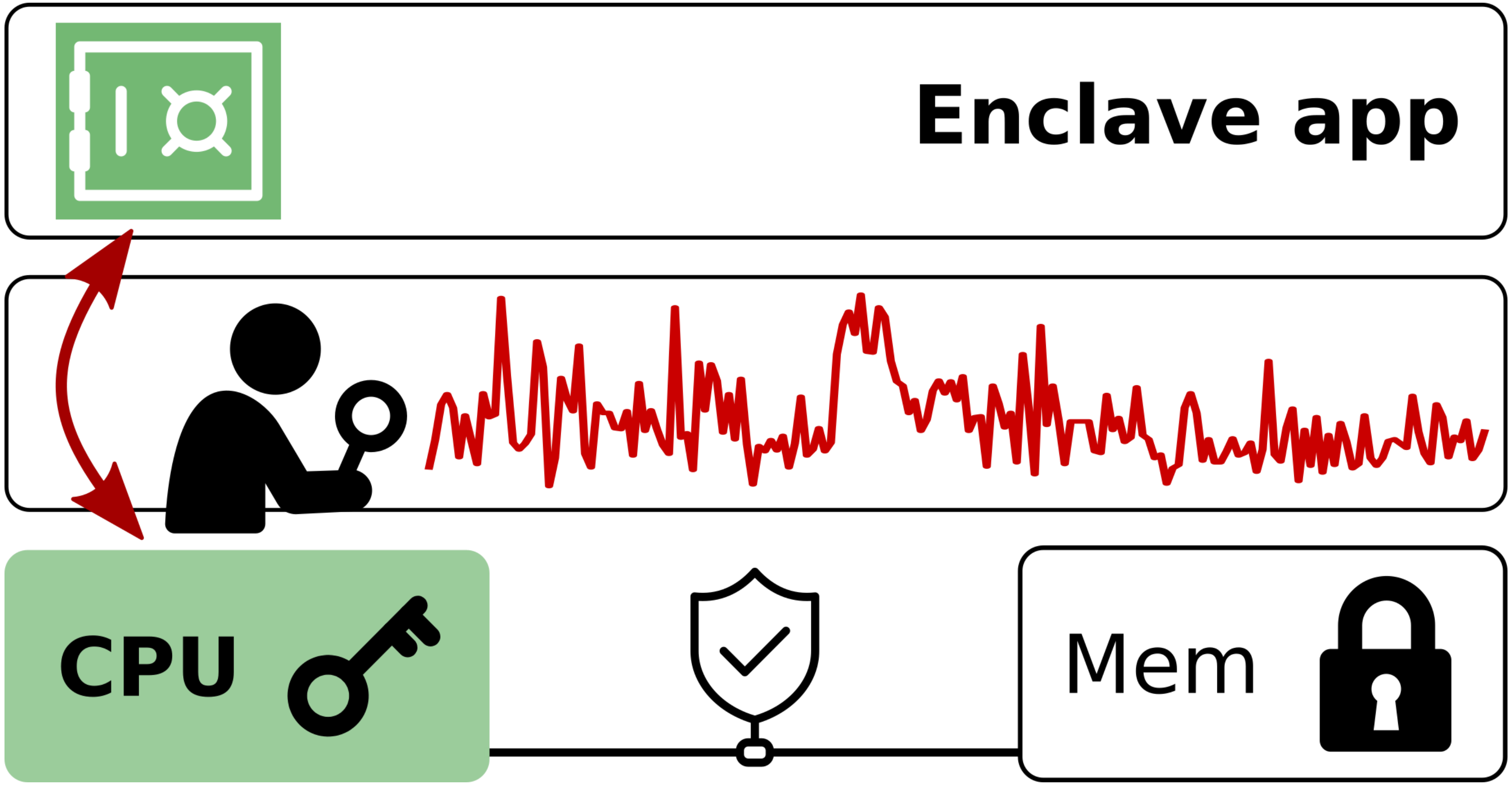
Part I: Problem statement

Enclaved execution: Reducing attack surface



Intel® SGX: Hardware-level **isolation and attestation**

Enclaved execution: Privileged side channels



Game changer: Untrusted OS → new class of powerful **side channels!**

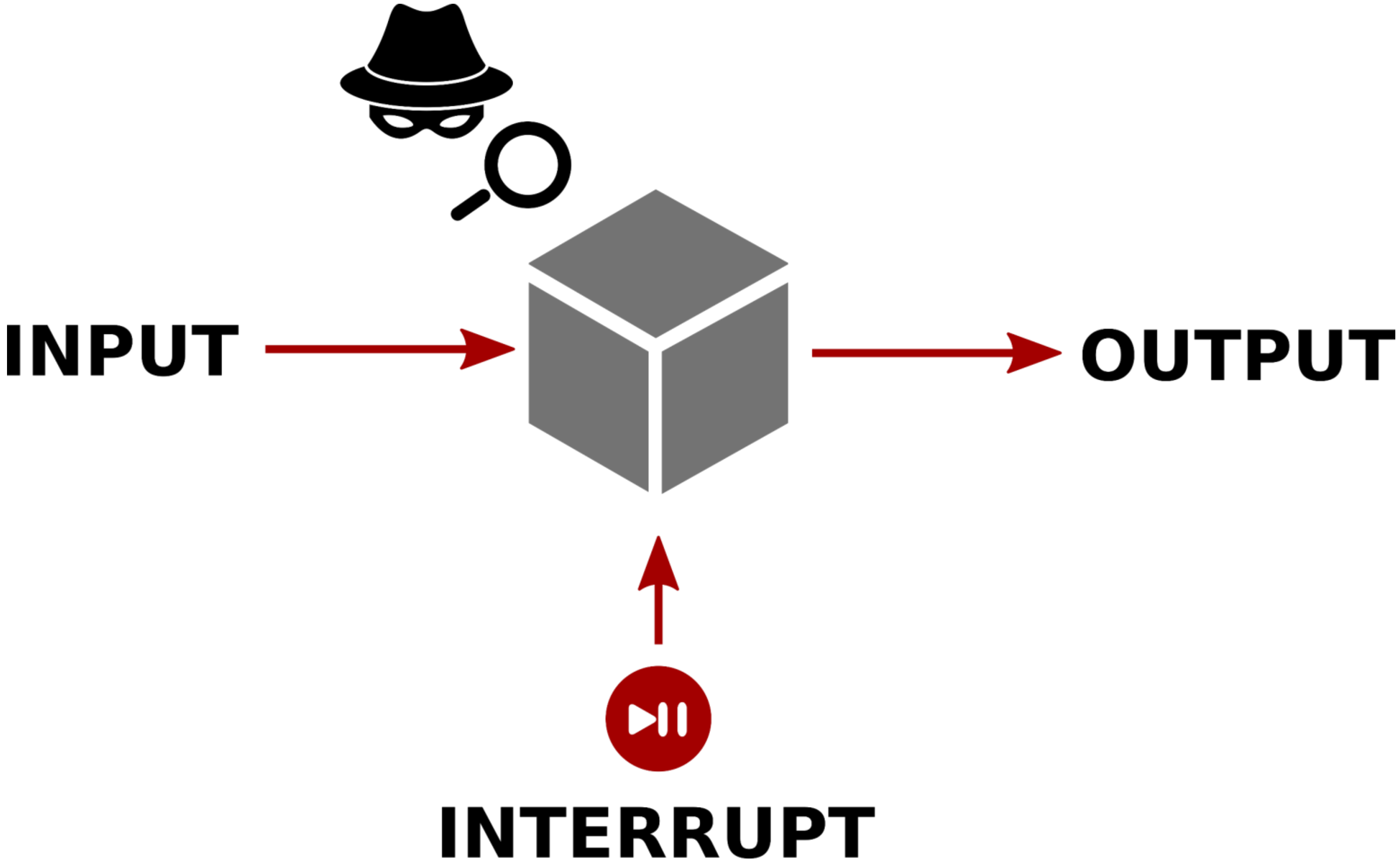
Challenge: Side-channel sampling rate

Fast
shutter speed

Medium
shutter speed

Slow
shutter speed

SGX-Step: Executing enclaves one instruction at a time

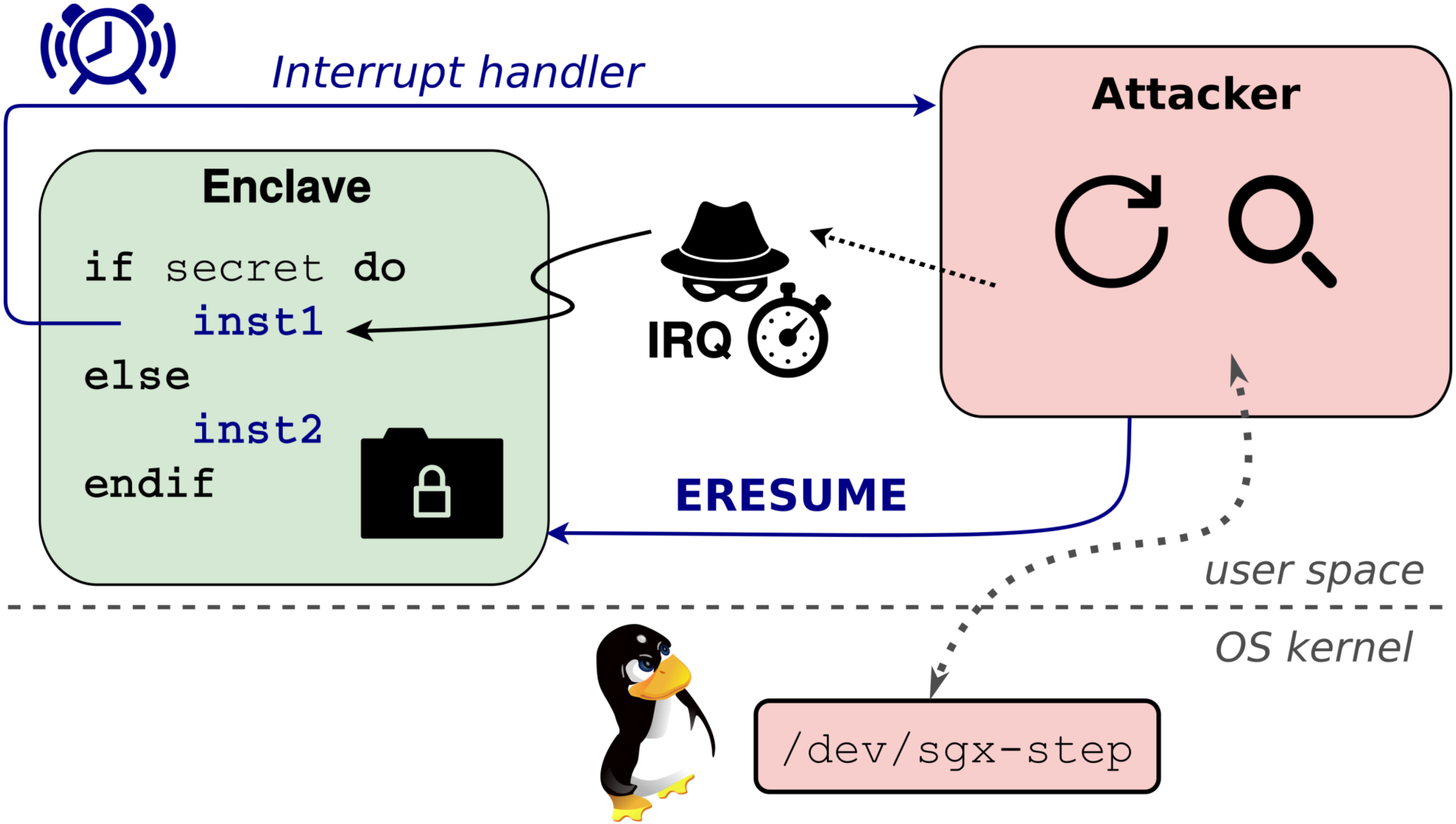


SGX-Step: Executing enclaves one instruction at a time

SGX-Step

https://github.com/jovanbulck/sgx-step

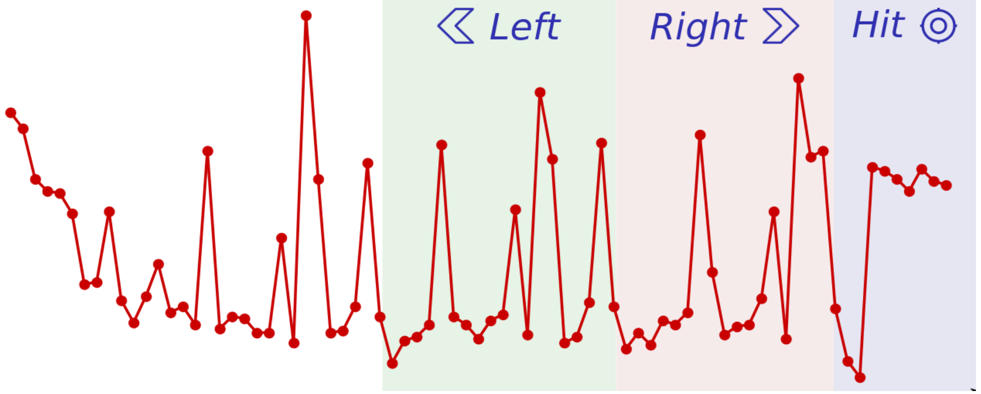
SGX-Step: Executing enclaves one instruction at a time



```
if secret do
    inst1
else
    inst2
endif
```

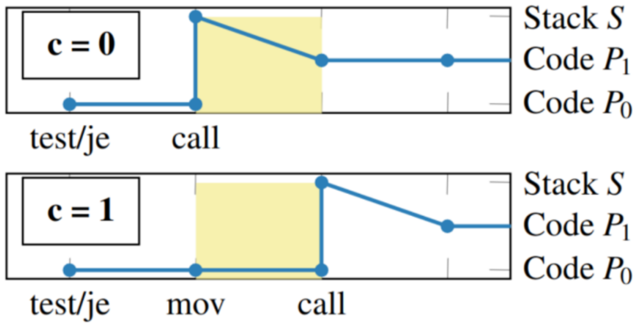
```
/dev/sgx-step
```

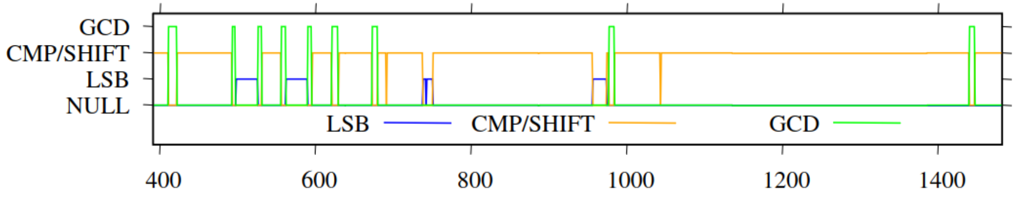
SGX-Step: Enabling a line of high-resolution attacks



1. Interrupt latency

[CCS'18, USENIX'21]





SGX-Step

2. Interrupt counting

[CCS'19, CHES'20-21, USENIX'20]

3. Zero-step replaying

[USENIX'18, CCS'19, ISCA'19, S&P'21]

ÆPIC LEAK

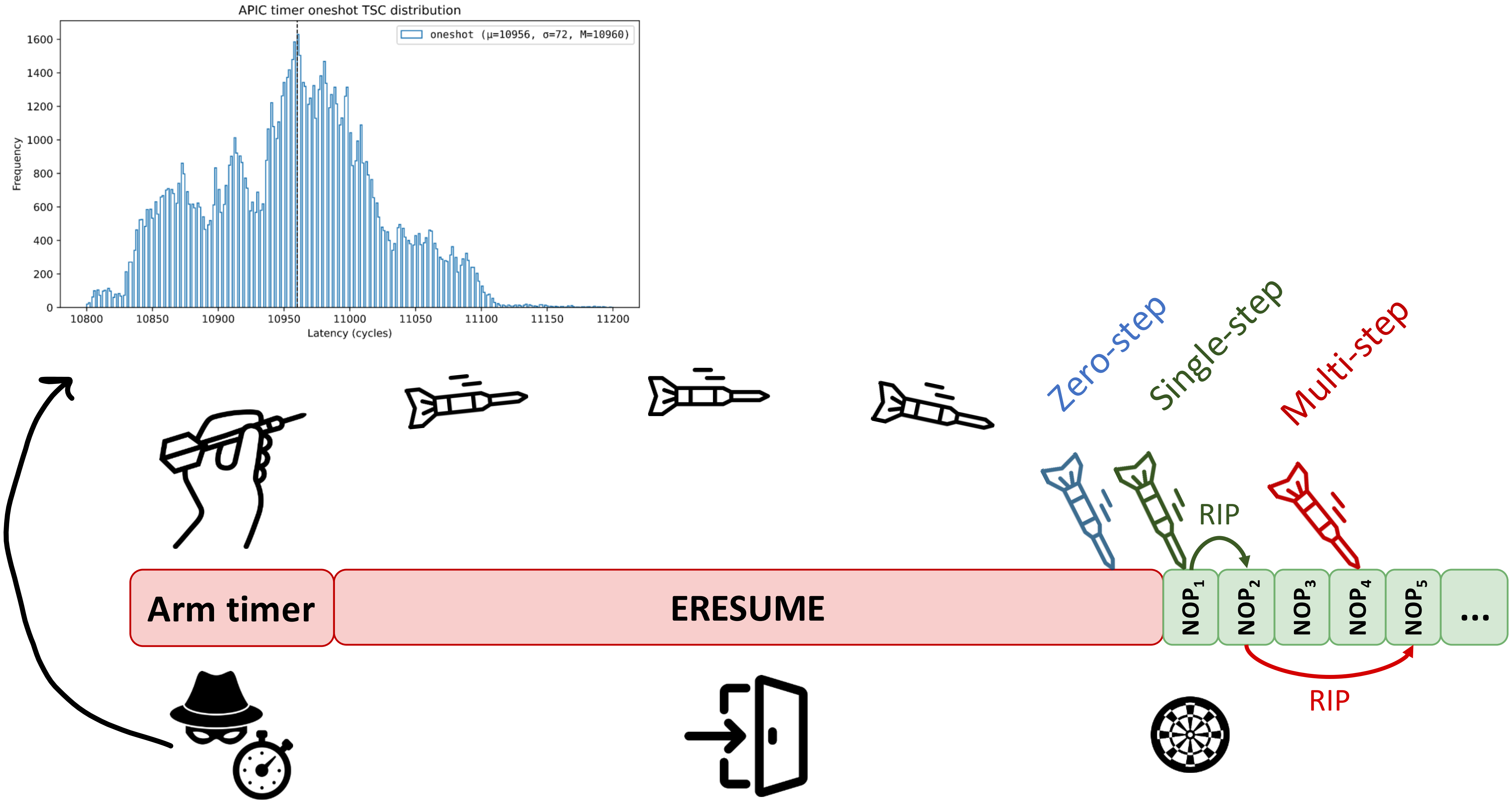
4. Amplification

[ATC'17, CCS'19/21, CHES'17-19, S&P'20-21, USENIX'17/18/22]

SGX-Step demo: Building a `memcmp` password oracle

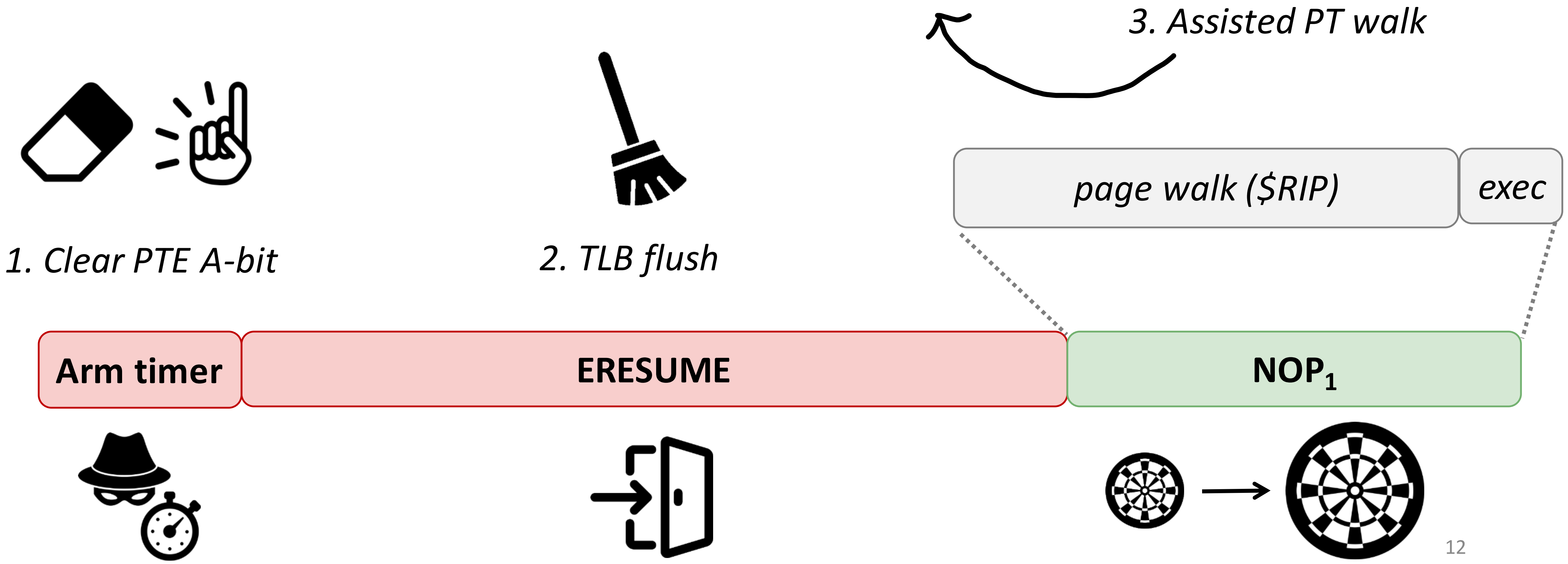
```
jo@breuer:~/sgx-step-demo$ sudo ./app
```

Root-causing SGX-Step: Aiming the timer interrupt



Root-causing SGX-Step: Microcode assists to the rescue!

PTE A-bit	Mean (cycles)	Stddev (cycles)
A=1	27	30
A=0	666	55



3. *Assisted PT walk*

page walk ($RIP)

exec

1. *Clear PTE A-bit*

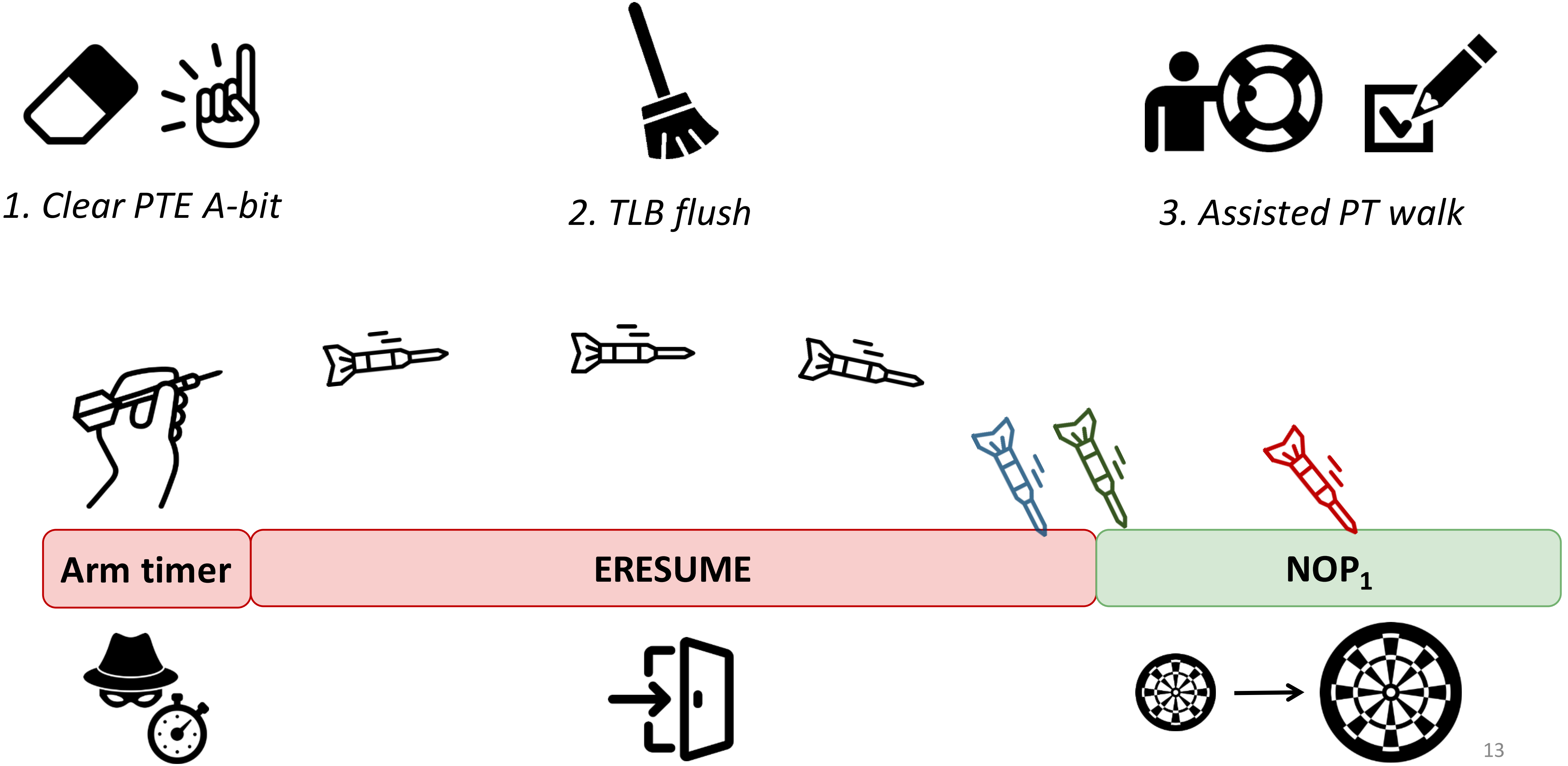
2. *TLB flush*

Arm timer

ERESUME

NOP$_1$

Root-causing SGX-Step: Microcode assists to the rescue!



1. Clear PTE A-bit

2. TLB flush

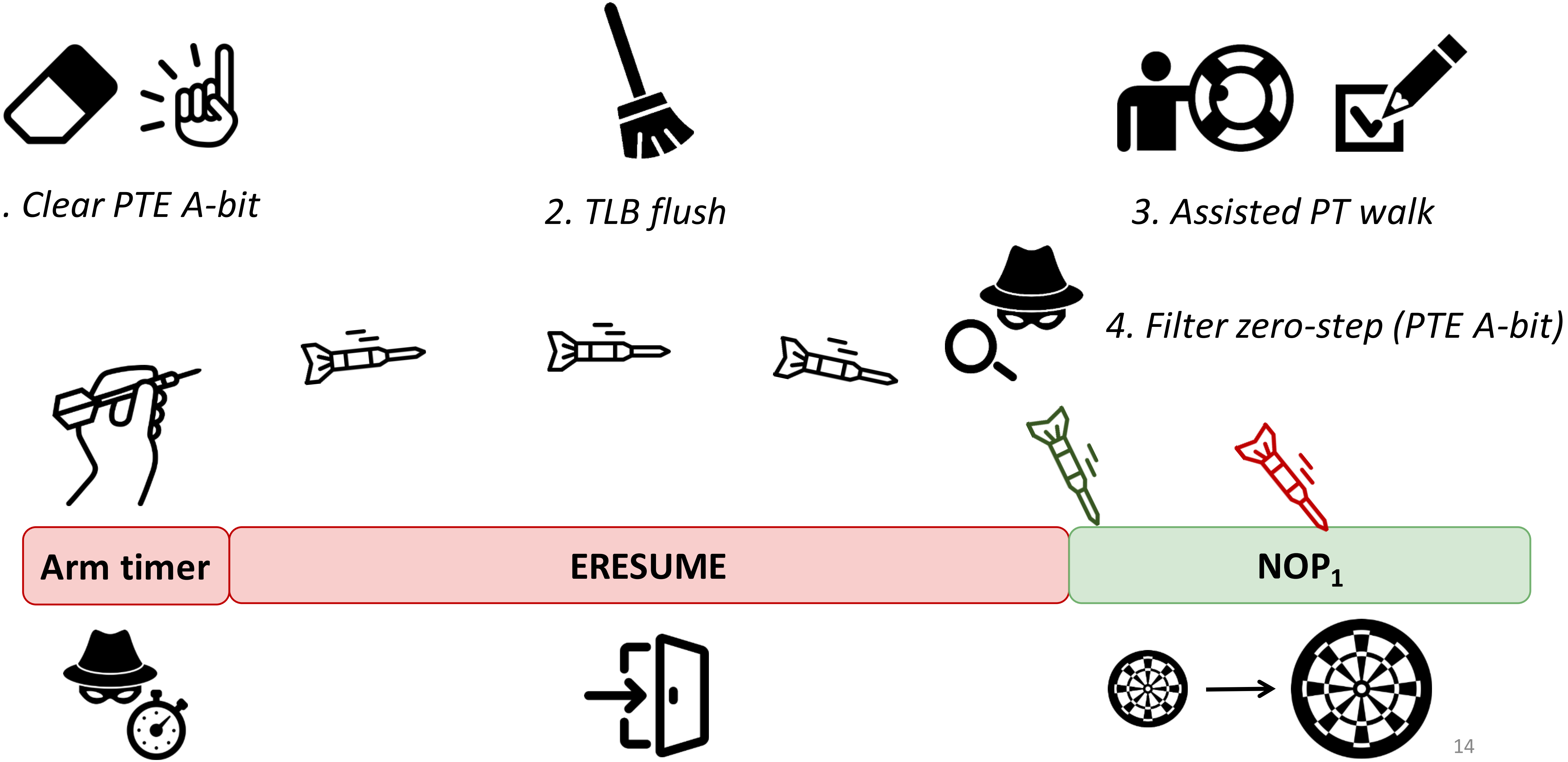
3. Assisted PT walk

Arm timer

ERESUME

NOP$_1$

Root-causing SGX-Step: Microcode assists to the rescue!



1. *Clear PTE A-bit*

2. *TLB flush*

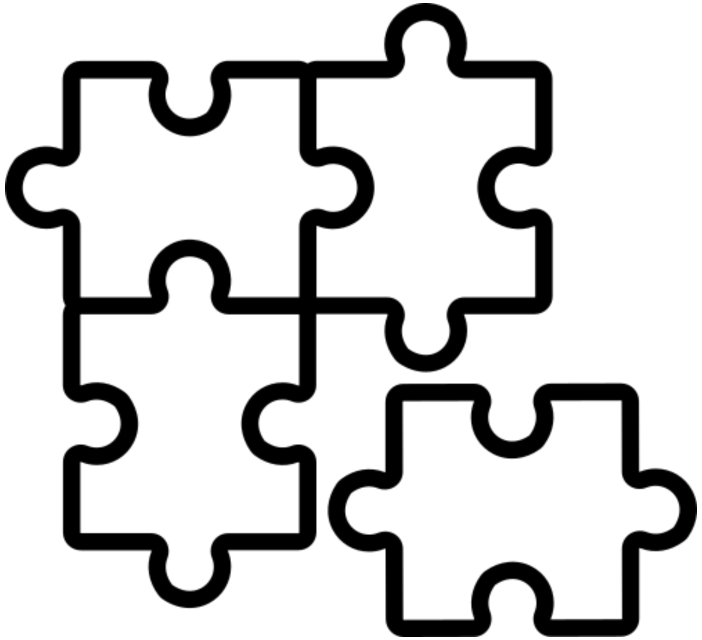
3. *Assisted PT walk*

4. *Filter zero-step (PTE A-bit)*

Arm timer

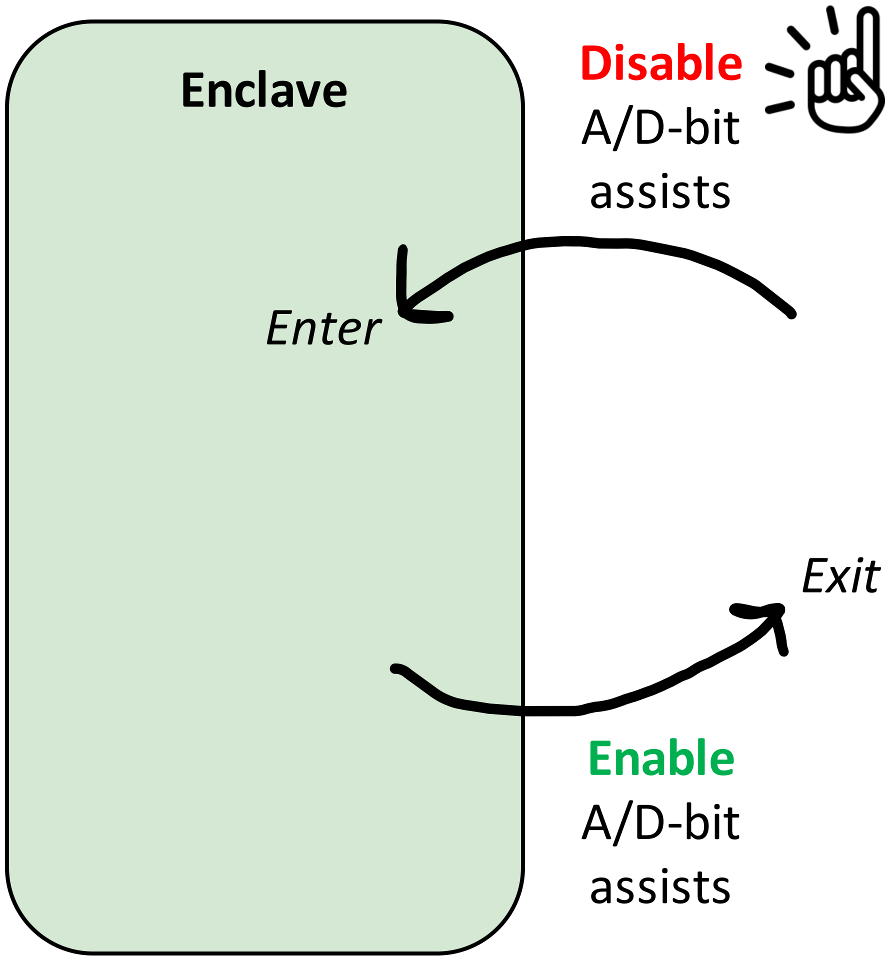
ERESUME

NOP$_1$



Part II: Solution overview

Ideas that were rejected (1)



Enclave

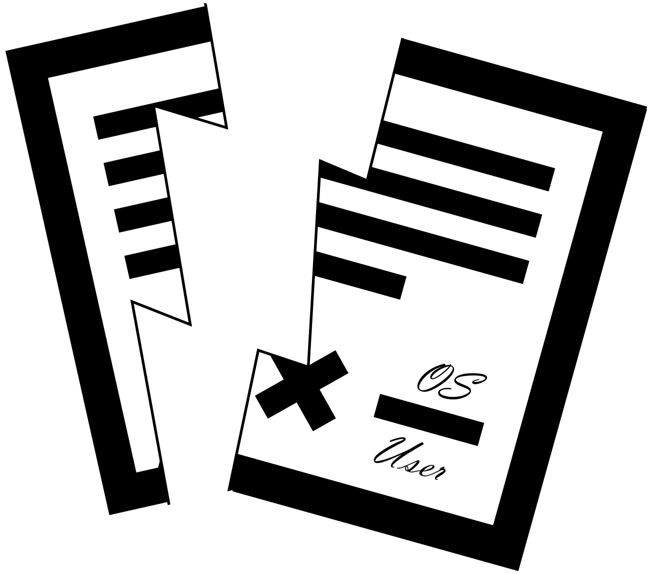
Disable A/D-bit assists

Enter

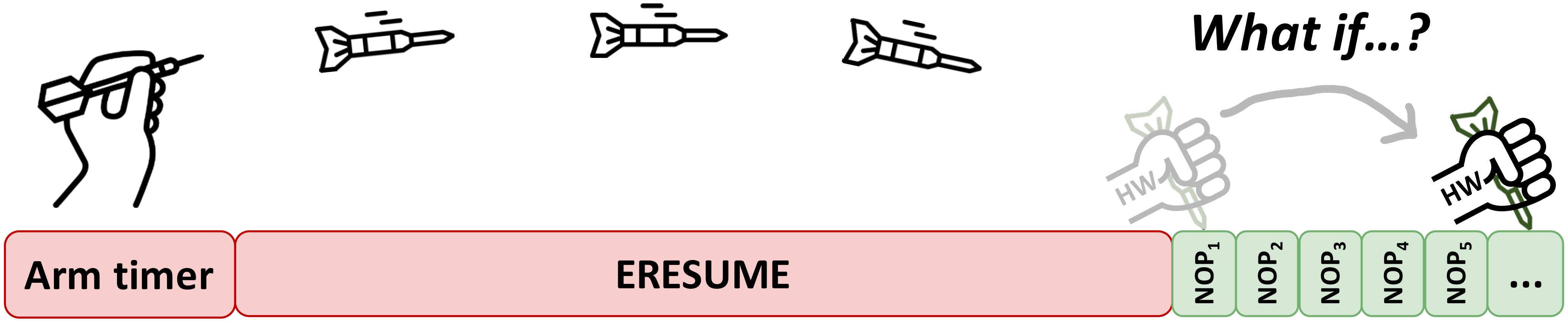
Exit

Enable A/D-bit assists

Breaks the OS/User contract



Ideas that were rejected (2)

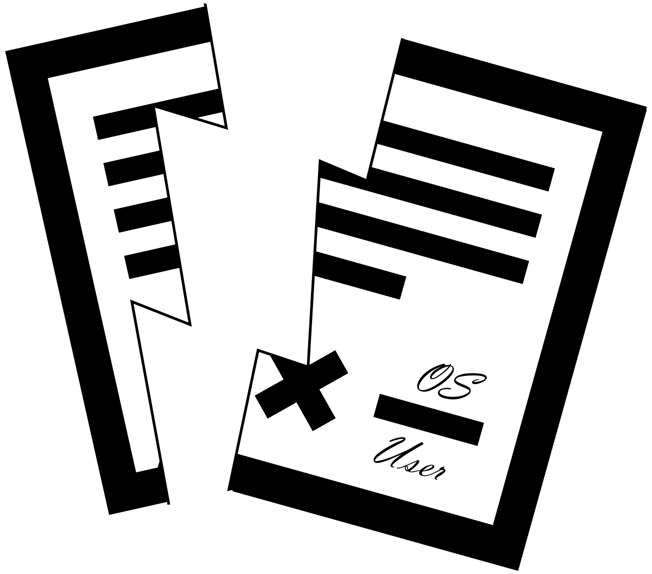


What if…?

Arm timer

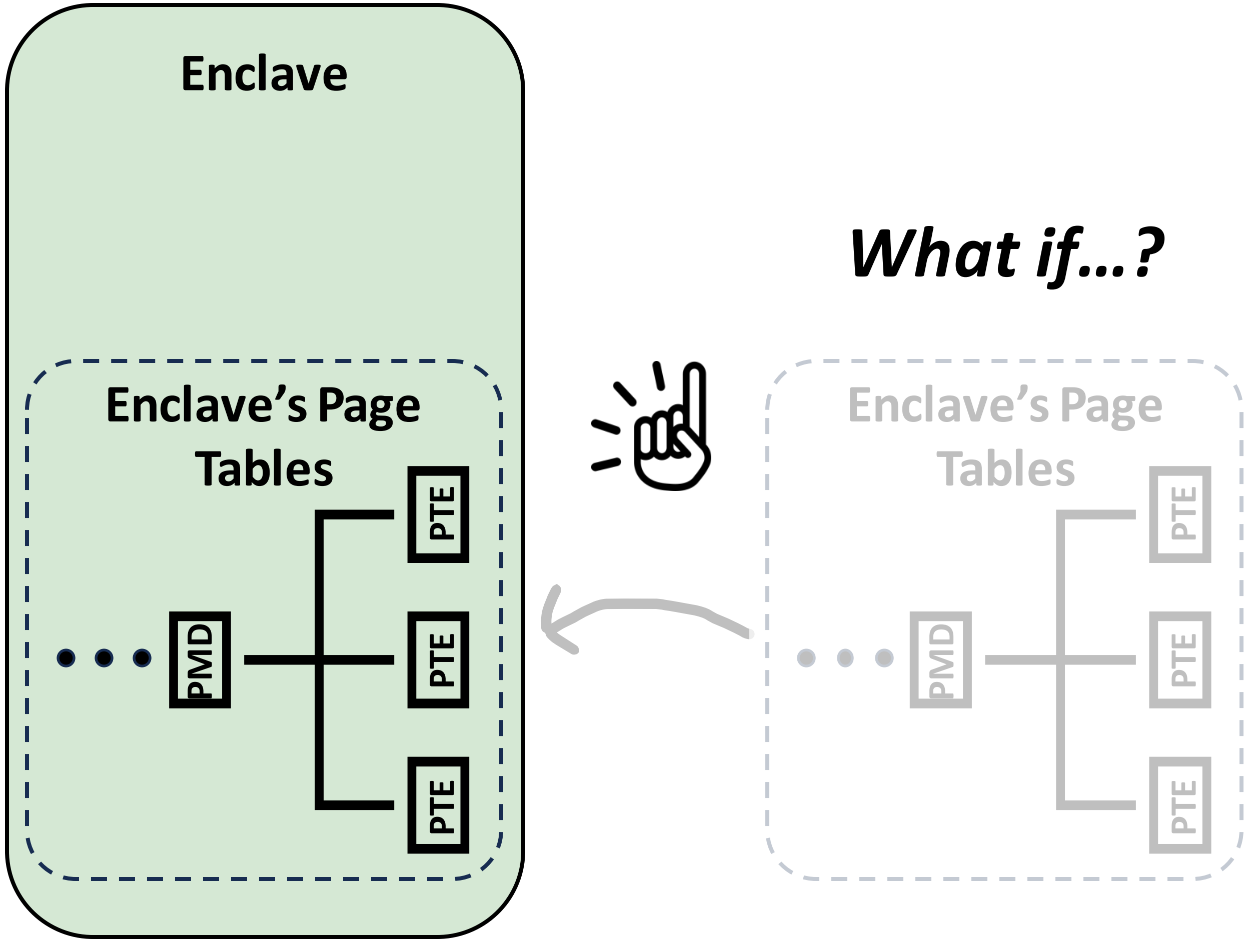
ERESUME

NOP$_1$ NOP$_2$ NOP$_3$ NOP$_4$ NOP$_5$ …



Highly complex

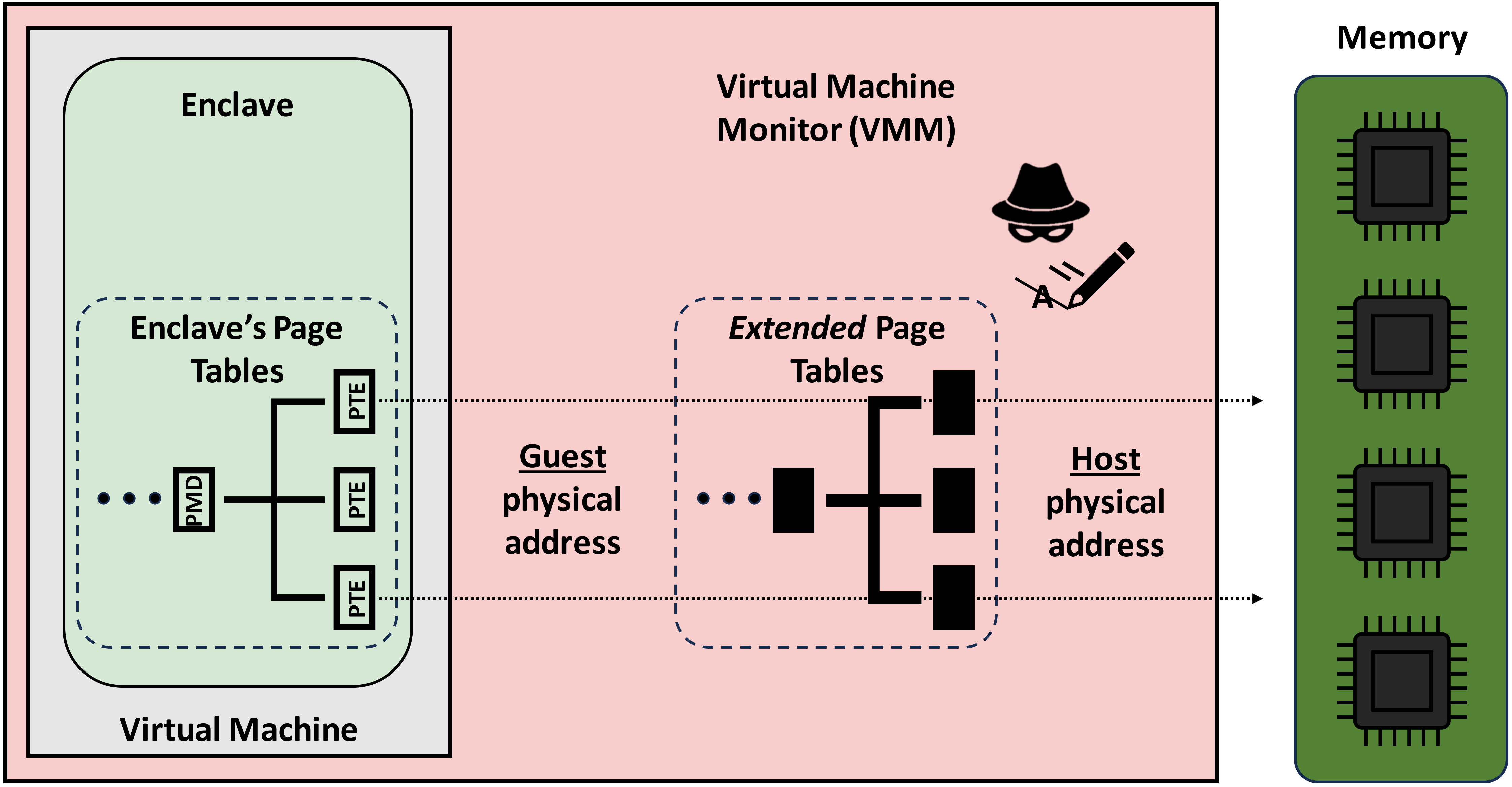
Ideas that were rejected (3)



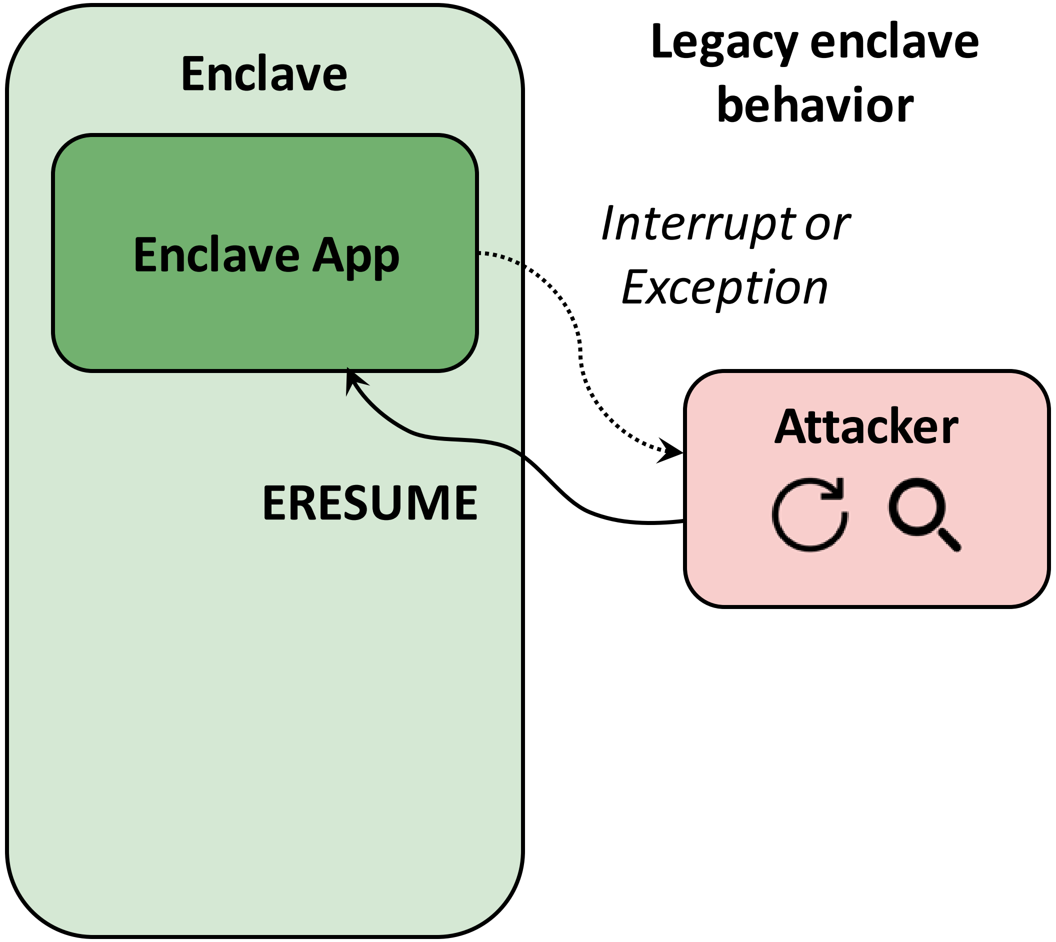
What if…?

Highly complex

Ideas that were rejected (3)



AEX-Notify solution overview



Legacy enclave behavior

Enclave

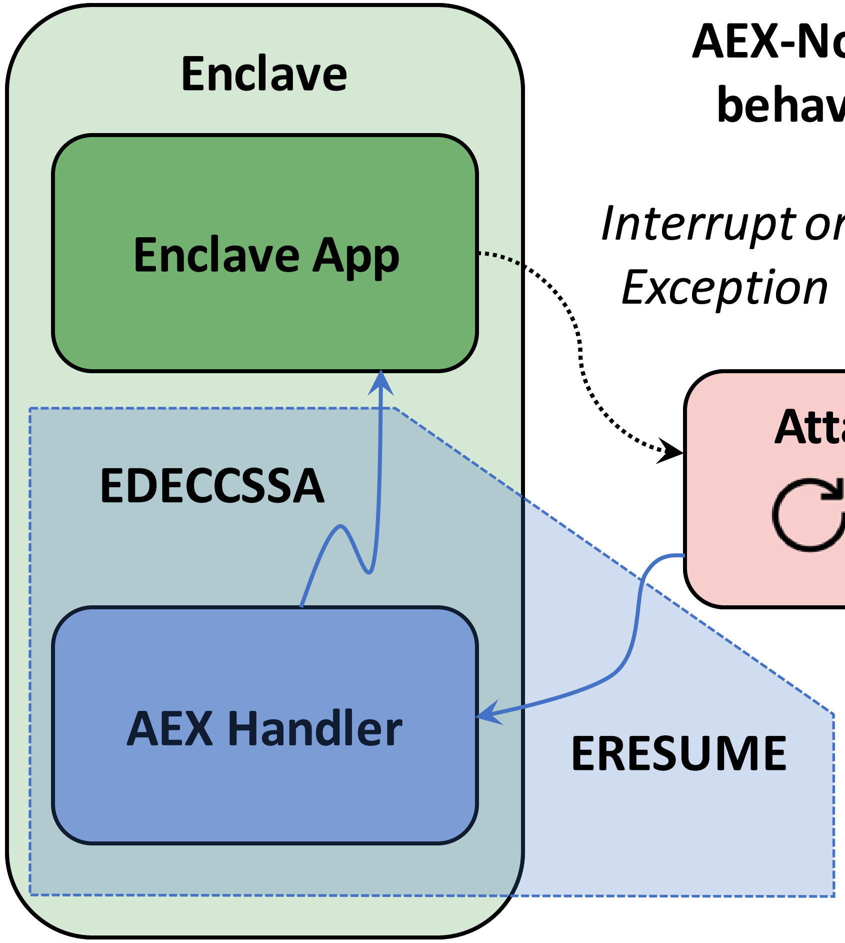
Enclave App

Interrupt or Exception

Attacker

ERESUME

AEX-Notify solution overview



Enclave

Enclave App

EDECCSSA

AEX Handler

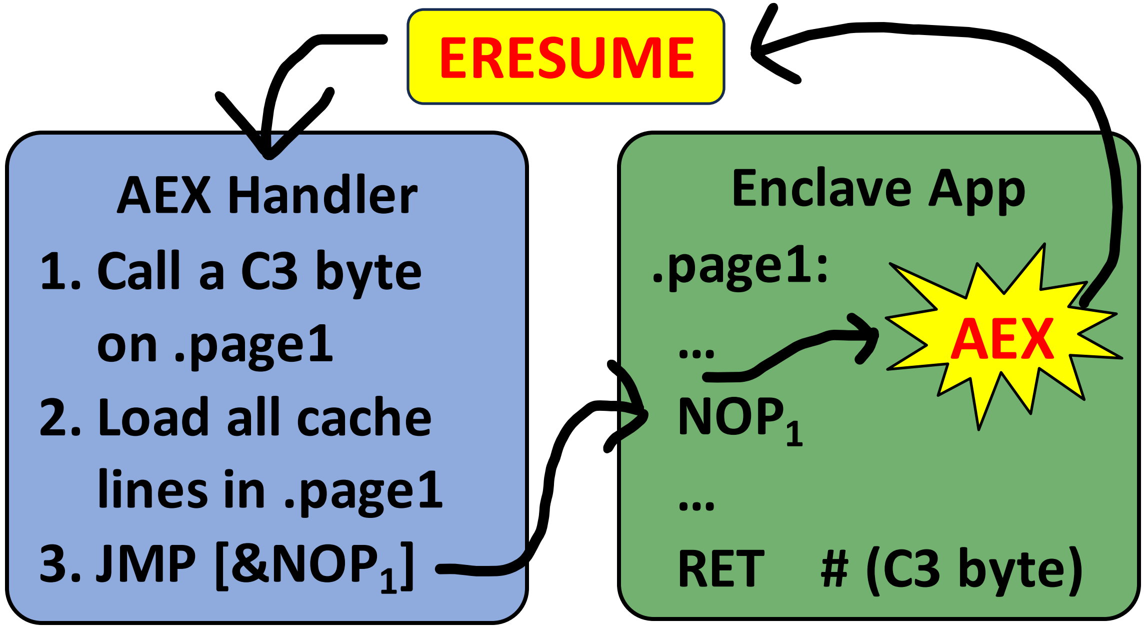
AEX-Notify behavior

Interrupt or Exception

Attacker

ERESUME

Legend: AEX-Notify ISA Extension



ERESUME

AEX Handler

1. Call a C3 byte on .page1
2. Load all cache lines in .page1
3. JMP [&NOP$_1$]

Enclave App

.page1:

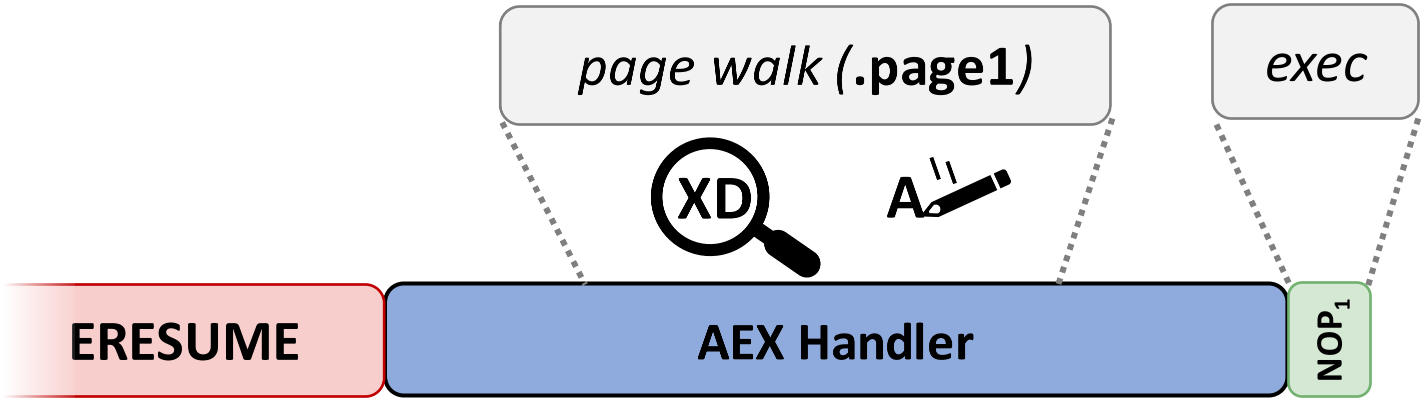
…

NOP$_1$

…

RET # (C3 byte)

AEX



page walk (.page1)

exec

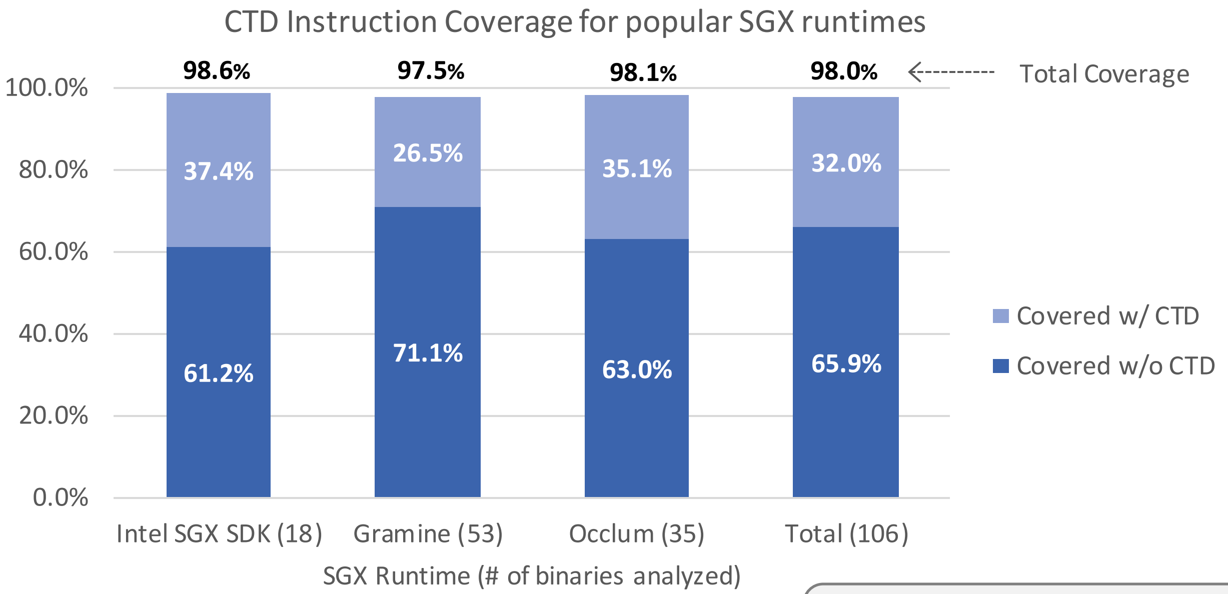
ERESUME

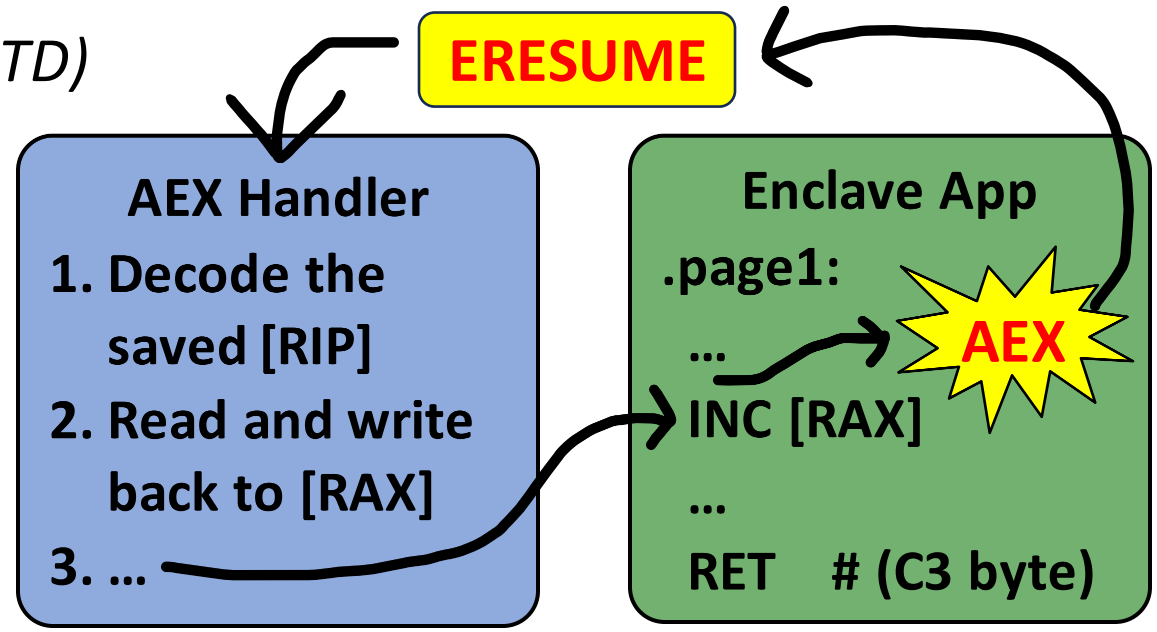
AEX Handler

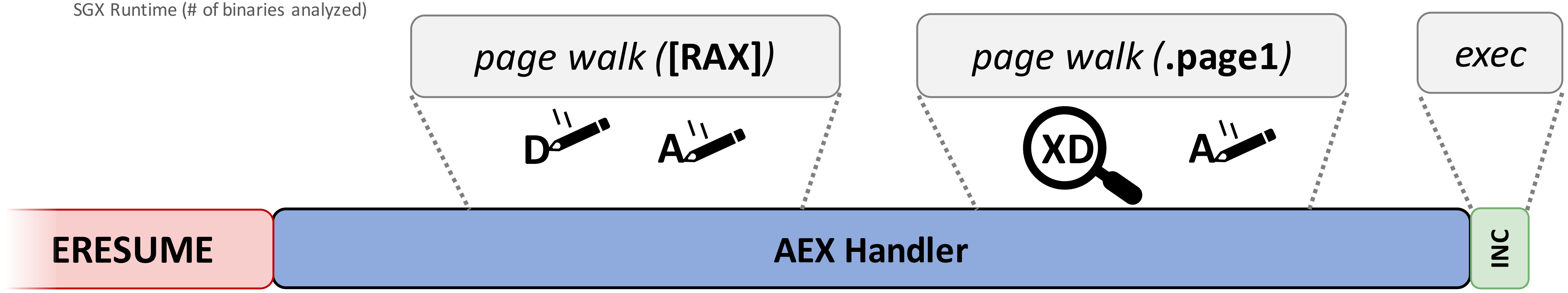
NOP$_1$

AEX-Notify solution overview

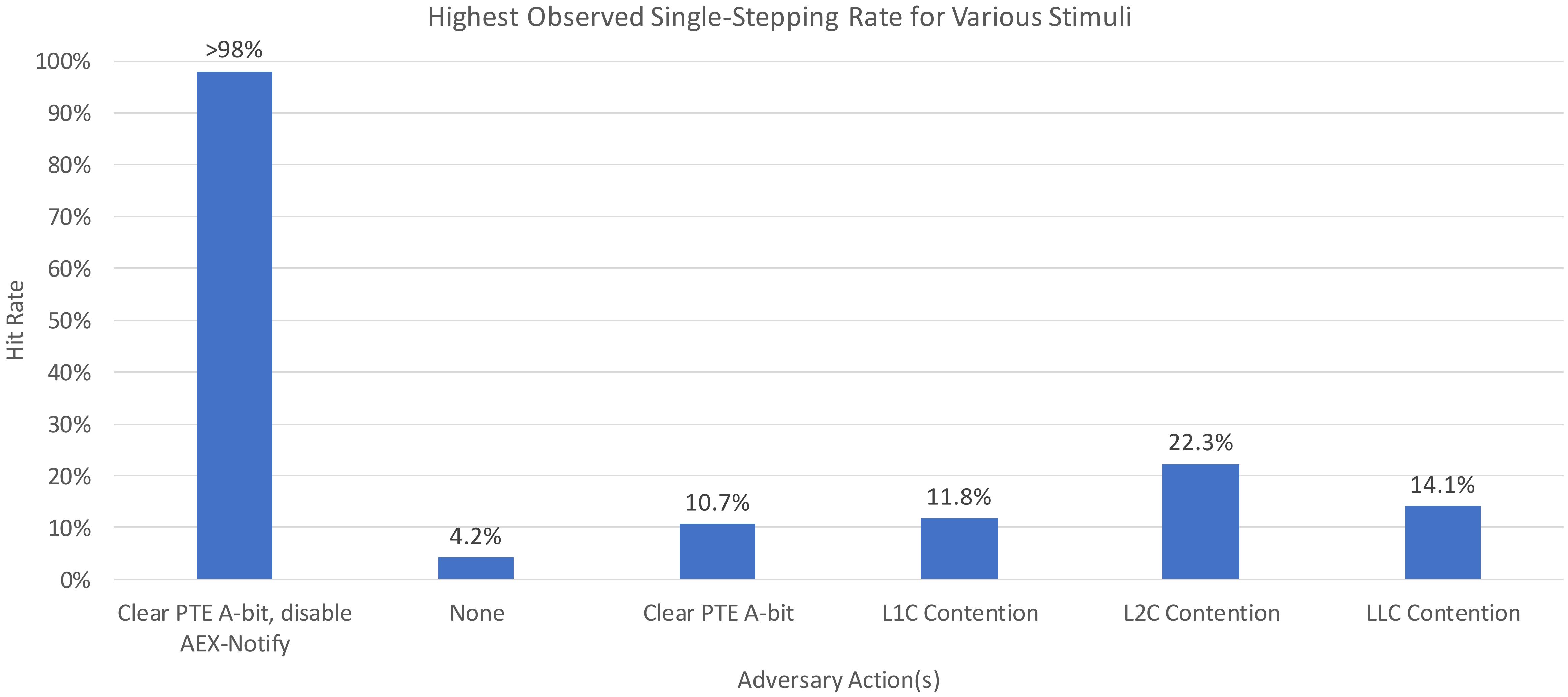
We implemented a fast, constant-time decoder (CTD)







Evaluation: Effectiveness



Experiments were conducted on an Intel Coffee Lake processor-based Xeon E3 platform

Evaluation: Performance

Real-world performance impact depends on **AEX frequency**…

	With Mitigation	Without Mitigation
Resuming an enclave thread	**58% slowdown** (6,500 → 10,300 cycles)	Performance unaffected
Handling an exception within an enclave	**76% speedup**	**88% speedup**

If the enclave is interrupted every 1 million cycles, the overhead is:

$$\frac{10{,}300 - 6{,}500}{1{,}000{,}000} = \mathbf{0.38}\%$$

Experiments were conducted on an Intel Ice Lake processor-based server platform

Conclusions

Extensible AEX-Notify **hardware-software** co-design

Eliminate root cause of perfect **single/zero stepping**

Minimal performance overhead and fast **CTD**

https://github.com/intel/linux-sgx/

Thank you! Questions?