Eleos: Exit-Less OS Services for SGX Enclaves

Meni Orenbach
Marina Minkin
Pavel Lifshits
Mark Silberstein

Accelerated Computing Systems Lab

EuroSys
ACM Sigops in Europe

Technion
Israel Institute
of Technology

What do we do?

Improve performance: I/O intensive & memory demanding SGX enclaves

Why?

Cost of SGX execution for these applications is high

How?

In-enclave System Calls & User Managed Virtual Memory

Results

Eleos vs vanilla SGX

2x ⬆ Throughput: memcached & face verification servers

Even for 5x ⬆ available enclave memory
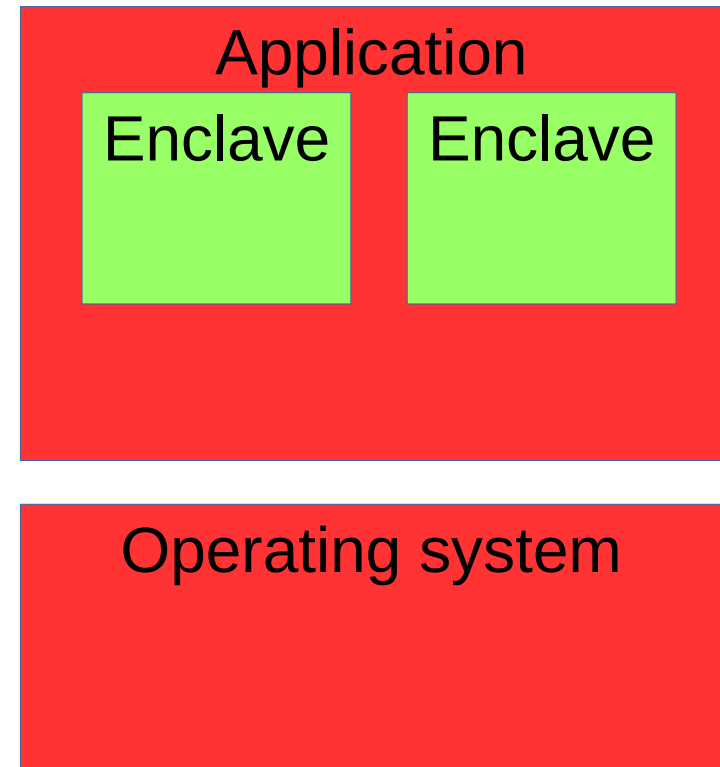
Available for Linux, Windows*

(*) Without Eleos, these applications crash in Windows enclaves

- **Background**
- Motivation
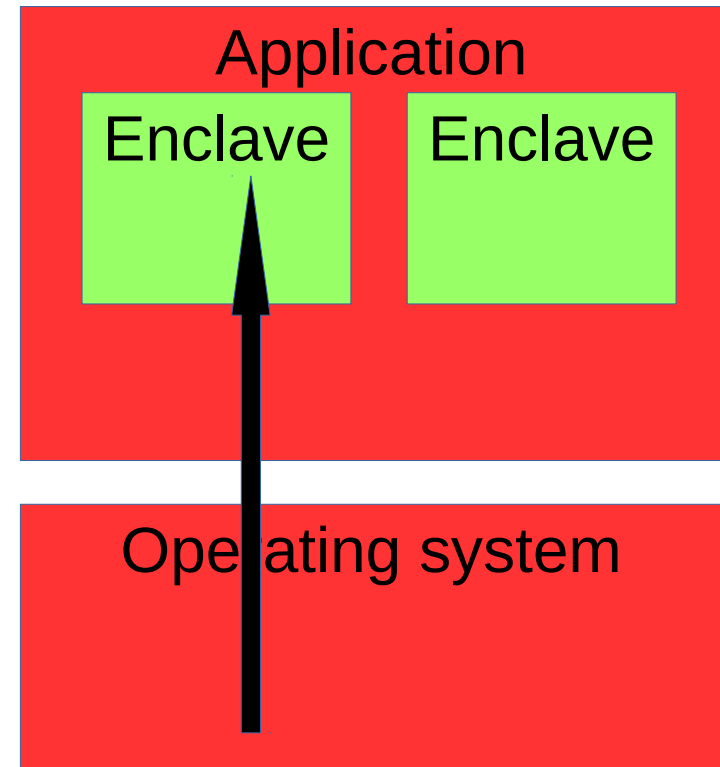- Overhead analysis
- Eleos design
- Evaluation

SGX enclaves are already here!

- Secured execution environment

- Reversed sandbox

- Small TCB

- Private code & data

  – Confidentiality

  – Integrity

  – Freshness

- Only CPU is trusted

Application

Enclave   Enclave

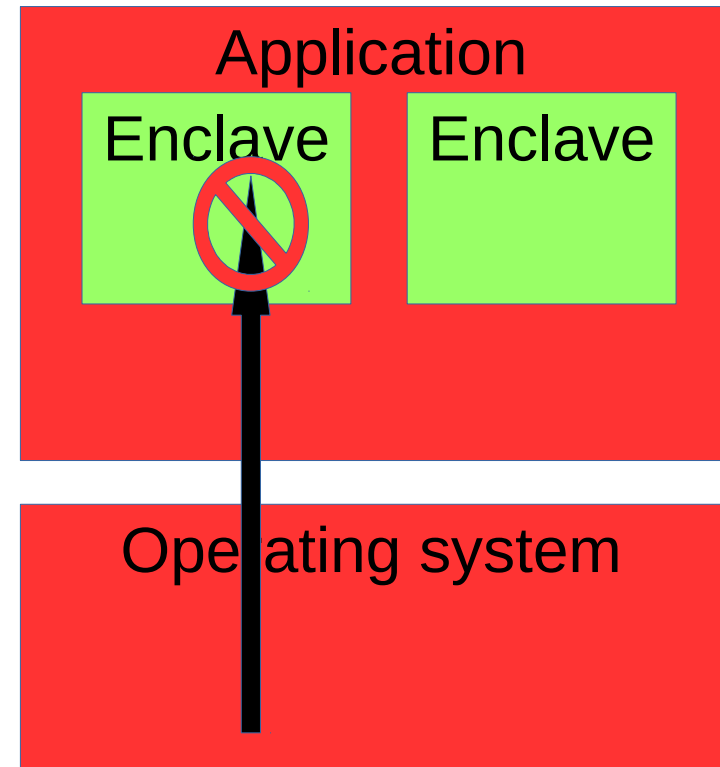Operating system

SGX enclaves are already here!

- Secured execution environment

- Reversed sandbox

- Small TCB

- Private code & data

  – Confidentiality

  – Integrity

  – Freshness

- Only CPU is trusted

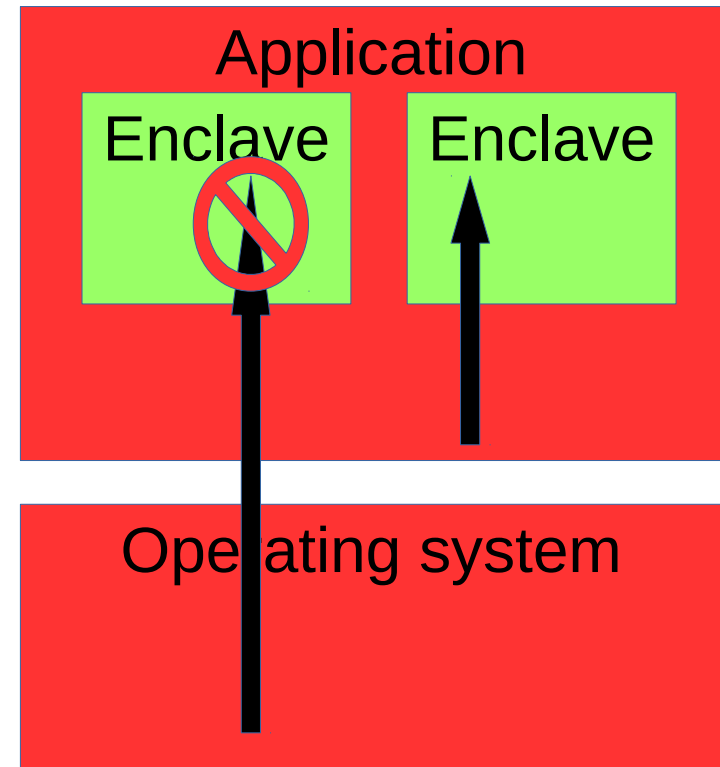Meni Orenbach, Technion

SGX enclaves are already here!

- Secured execution environment

- Reversed sandbox

- Small TCB

- Private code & data

  - Confidentiality

  - Integrity

  - Freshness

- Only CPU is trusted

SGX enclaves are already here!

- Secured execution environment

- Reversed sandbox

- Small TCB

- Private code & data

  – Confidentiality

  – Integrity
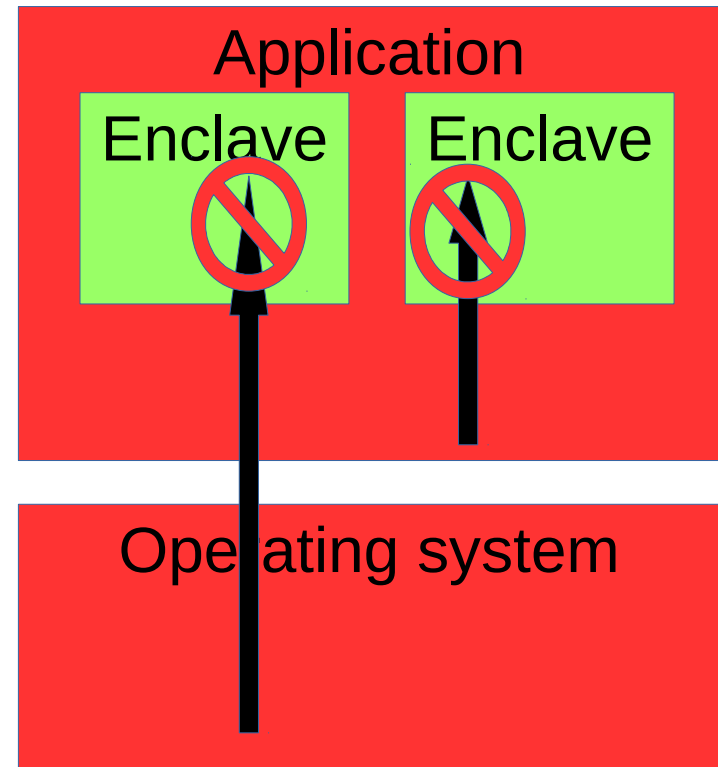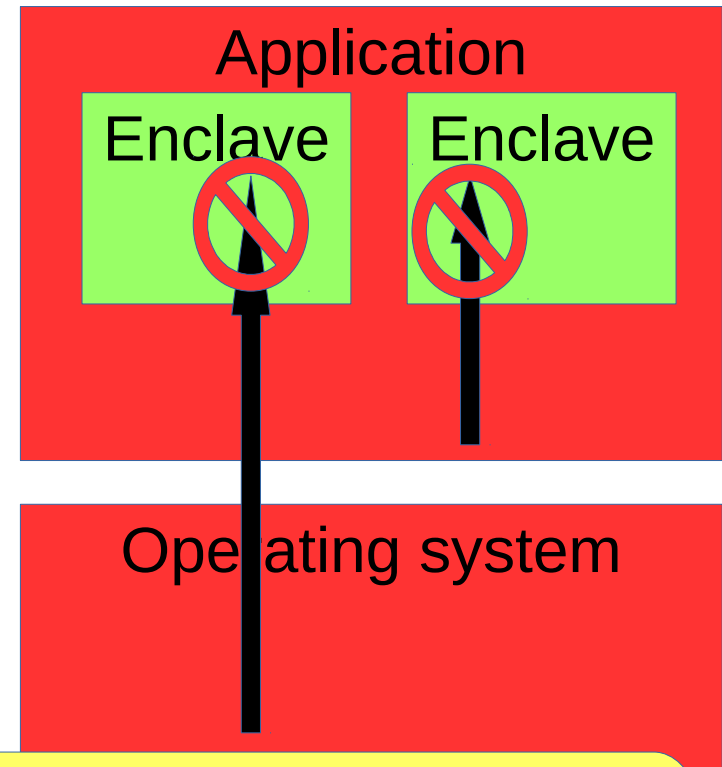
  – Freshness

- Only CPU is trusted

SGX enclaves are already here!

- Secured execution environment

- Reversed sandbox

- Small TCB

- Private code & data

  - Confidentiality

  - Integrity

  - Freshness

- Only CPU is trusted

SGX enclaves are already here!

- Secured execution environment

- Reversed sandbox

- Small TCB

- Private code & data

  - Confidentiality
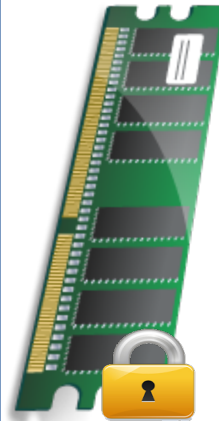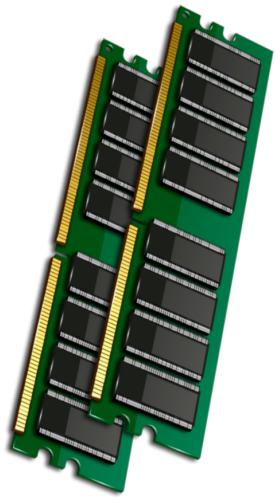
  - Integrity

  - Freshness

- O

Application

Enclave   Enclave

Operating system

Lets look at
How to secure server applications with enclaves

Background:
Lifetime of a secured server

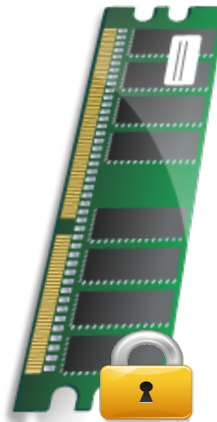Untrusted (Host & OS)

Trusted (Enclave)

Background:
Lifetime of a secured server

Untrusted (Host & OS)

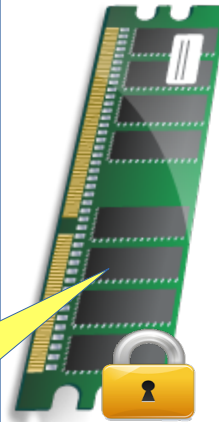Trusted (Enclave)

Untrusted memory
Unsecured access

Background:
Lifetime of a secured server

Untrusted (Host & OS)

Trusted (Enclave)

Untrusted memory
Unsecured access

Dedicated SGX mem
Limited to: 128 MB
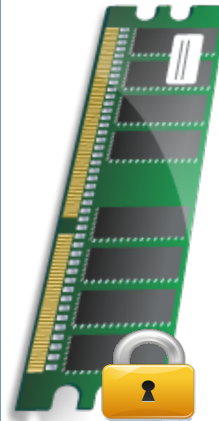Secured access
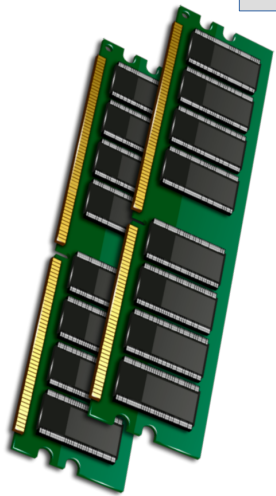
Meni Orenbach, Technion

Background:
Lifetime of a secured server

Untrusted (Host & OS)

Trusted (Enclave)

Host app

Wait for network requests

Background:
Lifetime of a secured server

Host app

Wait for network requests

Background:
Lifetime of a secured server

Untrusted (Host & OS) | Trusted (Enclave)

Host app

Wait for network requests

Enter enclave

Decrypt requests

Background:
Lifetime of a secured server

Host app

Wait for network requests

Enter enclave

Decrypt requests

Process requests

Background:
Lifetime of a secured server

Untrusted (Host & OS)

Trusted (Enclave)

Host app

Wait for network requests

Enter enclave

Decrypt requests

Process requests

Encrypt responses

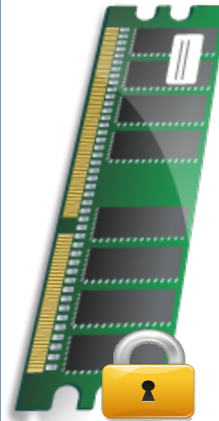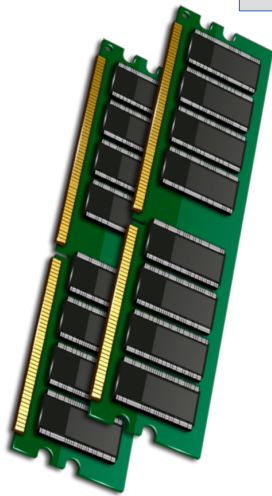Meni Orenbach, Techni... 17

Background:
Lifetime of a secured server

Untrusted (Host & OS)   Trusted (Enclave)

Host app

Wait for network requests

Enter enclave

Decrypt requests

Process requests

Exit enclave

Send responses

Encrypt responses

eni Orenbach, Technion   18

SGX enclaves should be fast

- ISA extensions

- Implemented in HW & Firmware

- Same CPU HW

- In-cache execution suffers no overheads

Meni Orenbach, Technion
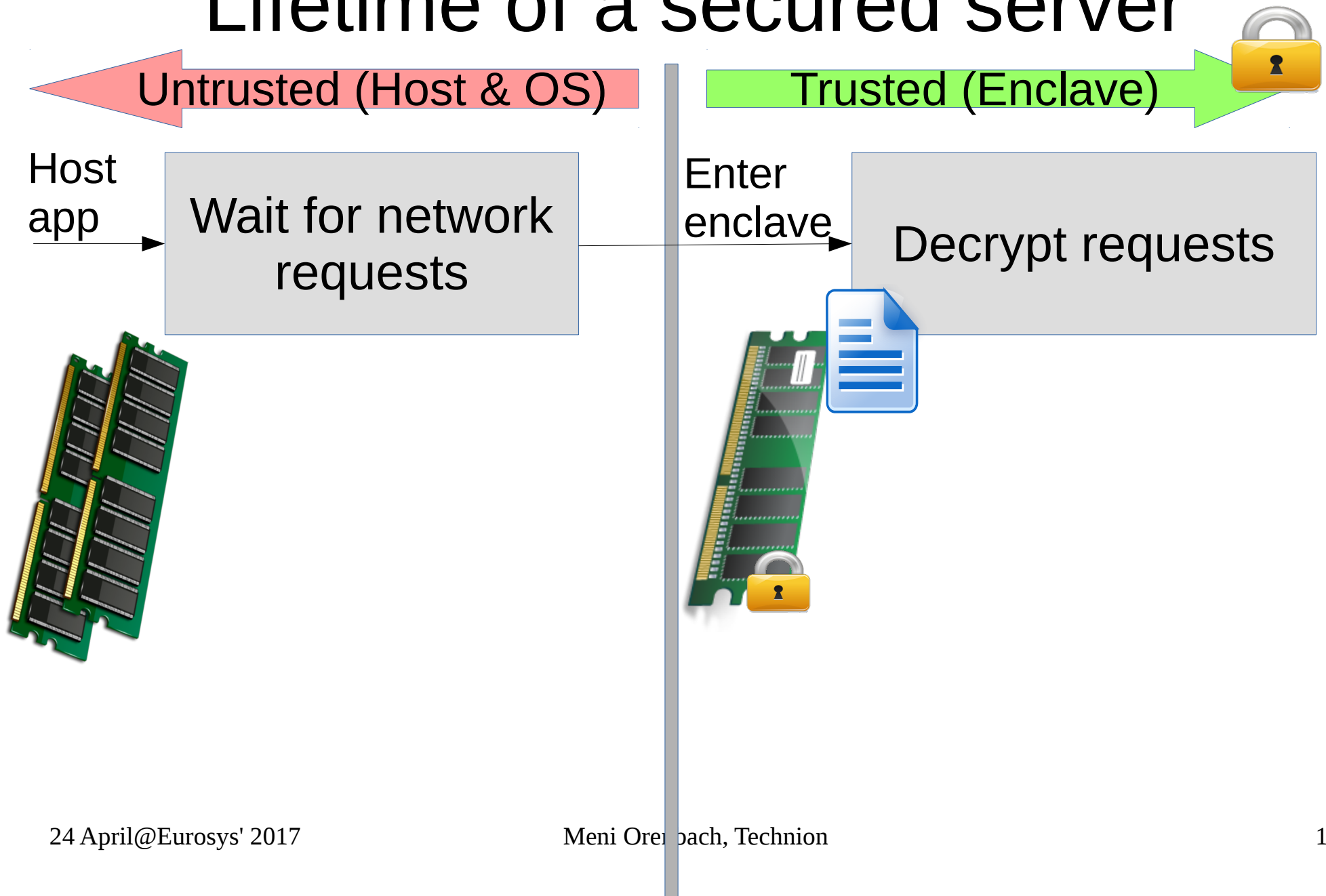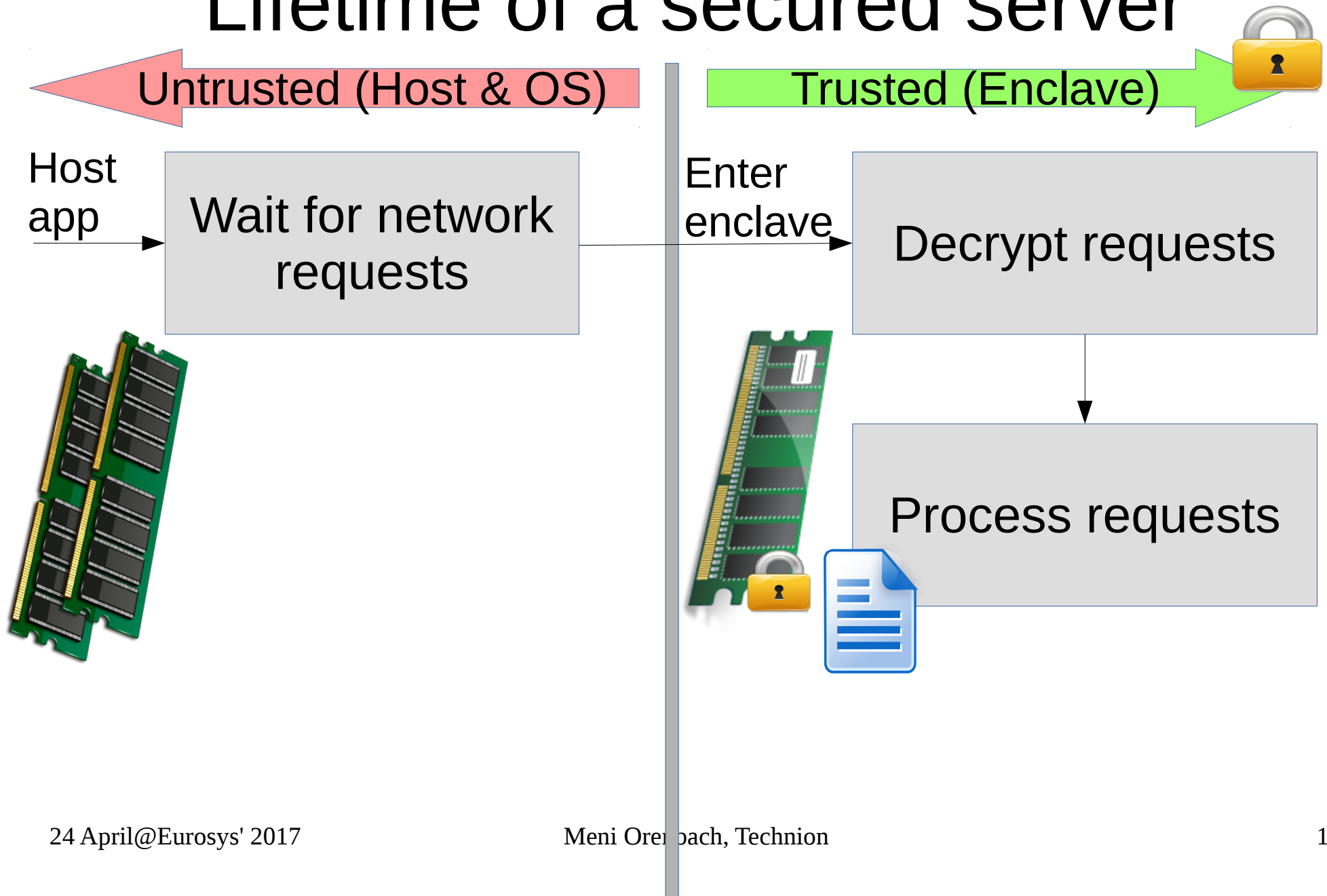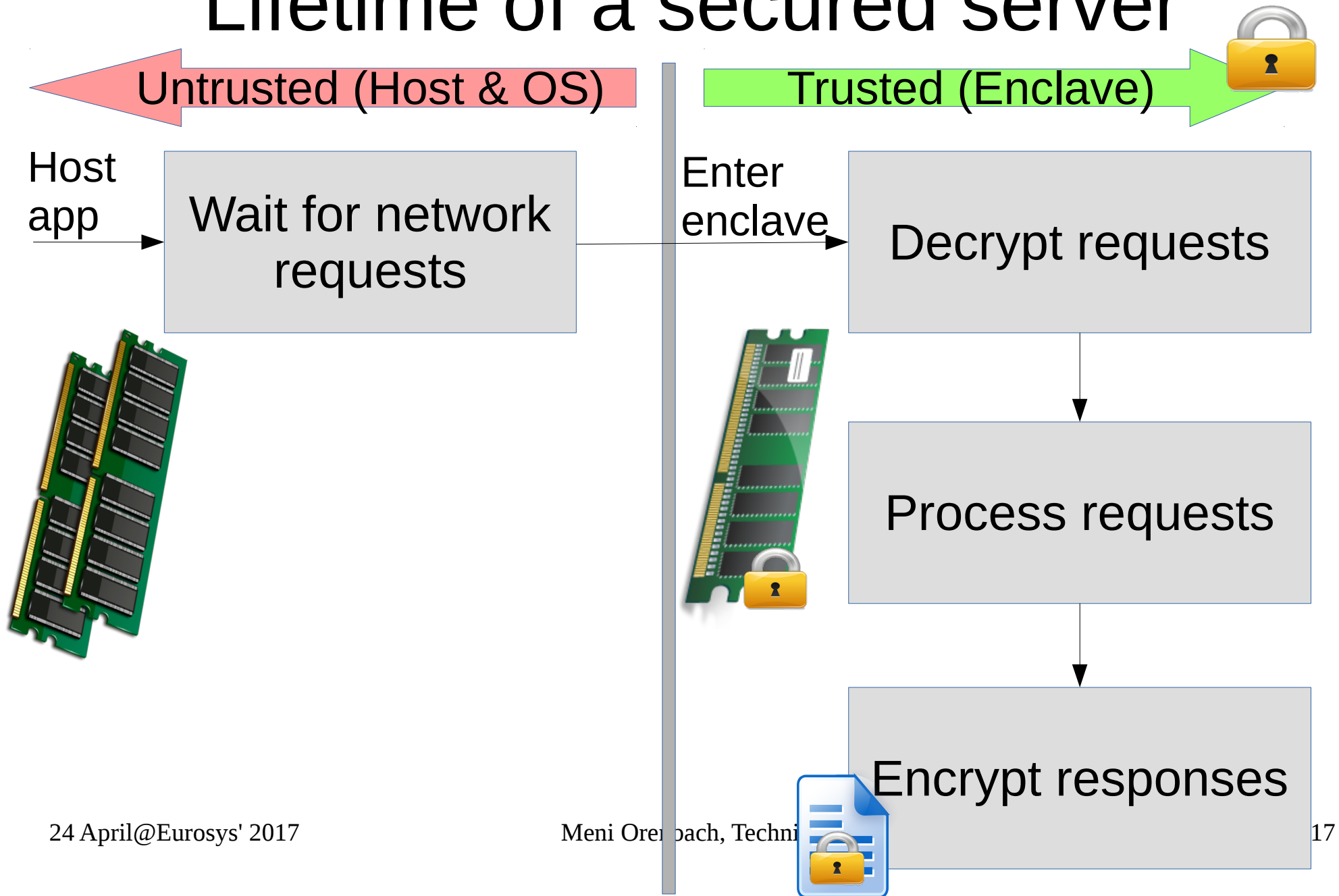
SGX enclaves should be fast

- ISA extensions

- Implemented in HW & Firmware

- Same CPU HW

- In-cache execution suffers no overheads

However...

Meni Orenbach, Technion
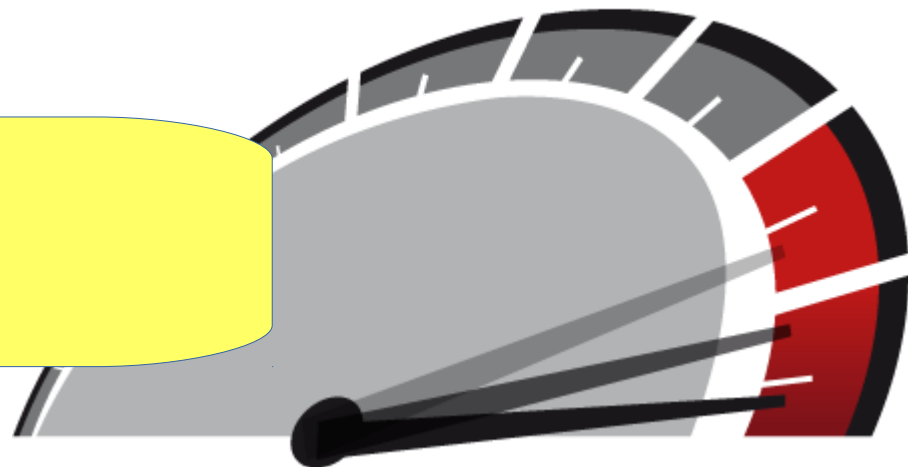
Executing a Key-Value Store in enclave is <u>slower</u>

Executing a Key-Value Store in enclave is <u>slower</u>

Throughput: Slowdown factor

2
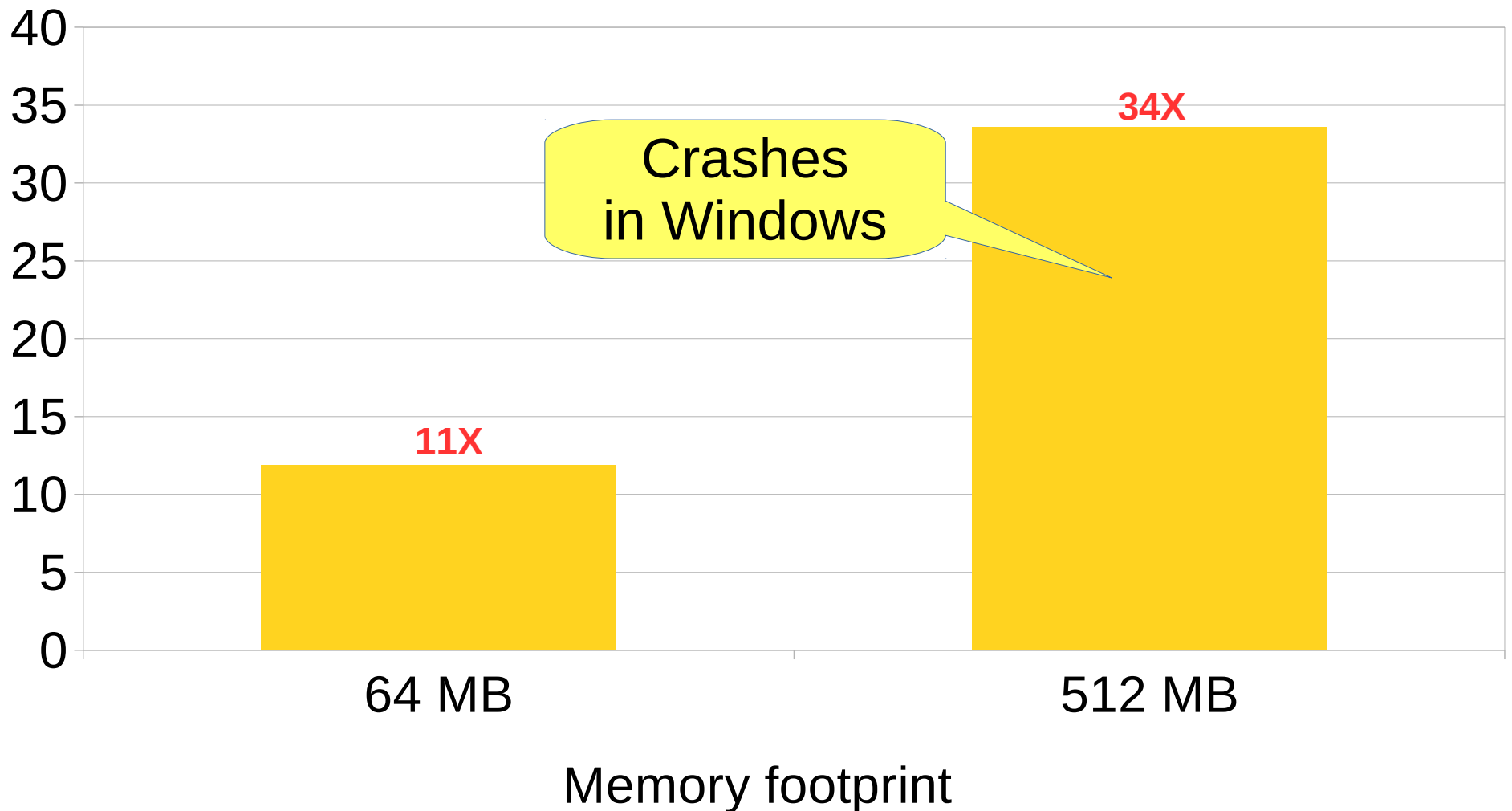
- Background
- Motivation
- **Overhead analysis**
- Eleos design
- Evaluation

Overhead analysis
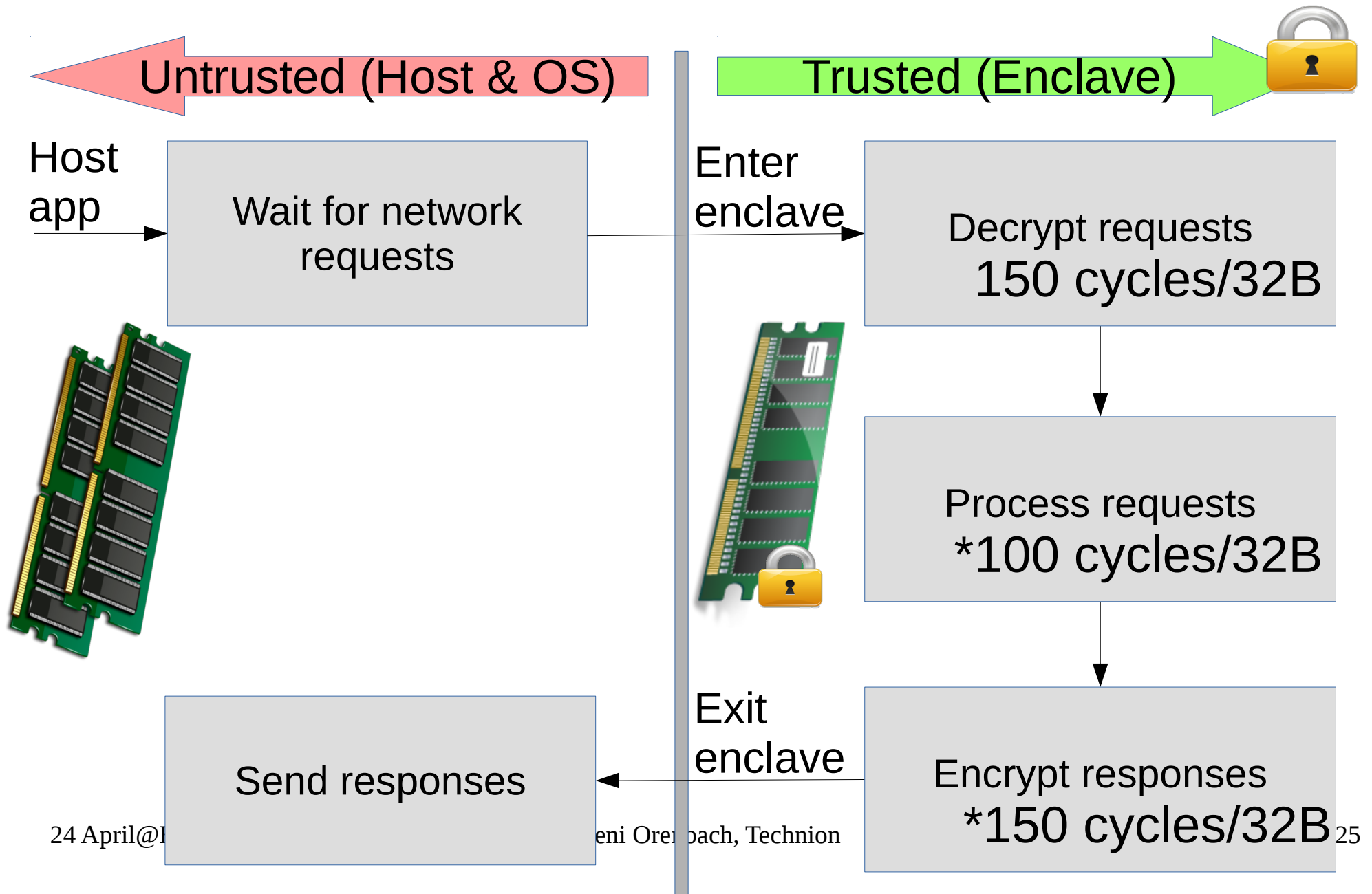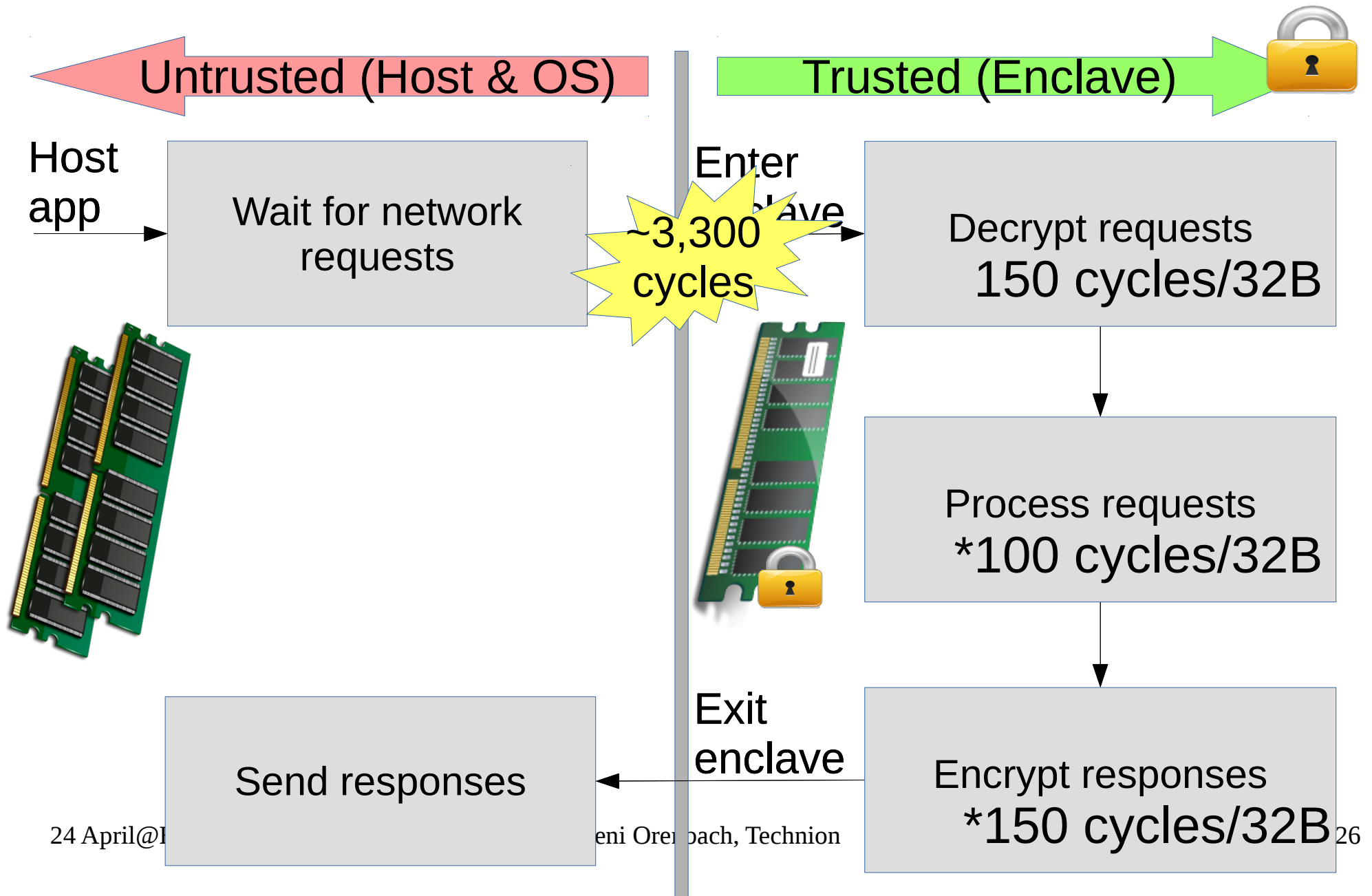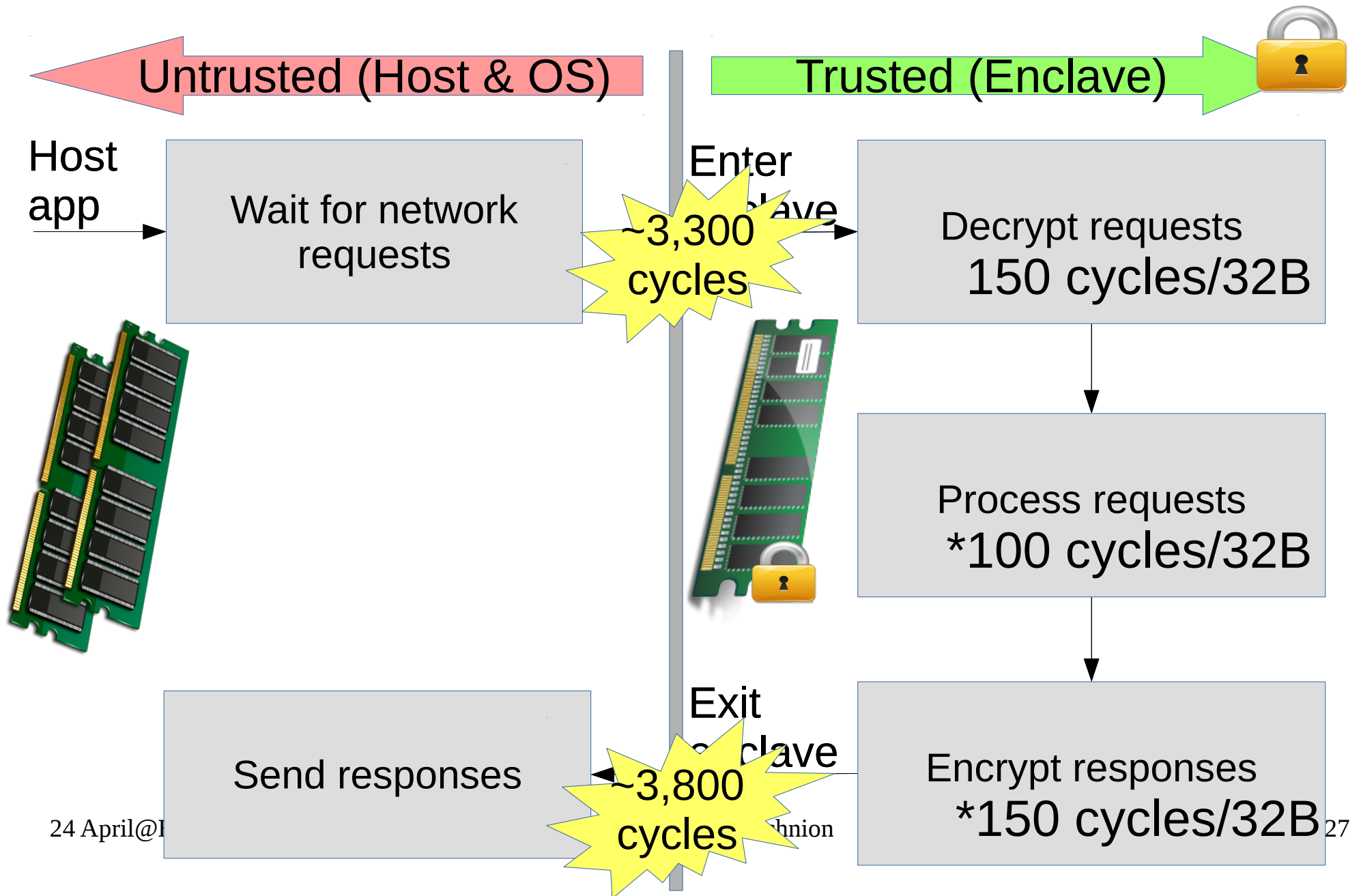
Trusted (Enclave)

Host app

Wait for network requests

Enter enclave

Decrypt requests
150 cycles/32B

Process requests
*100 cycles/32B

Exit enclave

Send responses

Encrypt responses
*150 cycles/32B

Overhead analysis

Trusted (Enclave)

Host app

Wait for network requests

Enter enclave
~3,300 cycles

Decrypt requests
150 cycles/32B

Process requests
*100 cycles/32B

Exit enclave

Send responses

Encrypt responses
*150 cycles/32B

Overhead analysis

Untrusted (Host & OS)   **Trusted (Enclave)**

Host app

| Wait for network requests |

Enter enclave
~3,300 cycles

| Decrypt requests 150 cycles/32B |

| Process requests *100 cycles/32B |

| Encrypt responses *150 cycles/32B |

Exit enclave
~3,800 cycles

| Send responses |

24 April@Technion
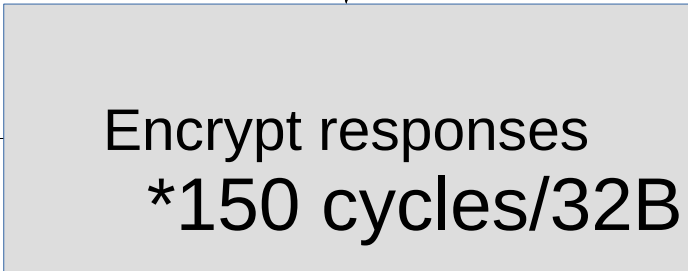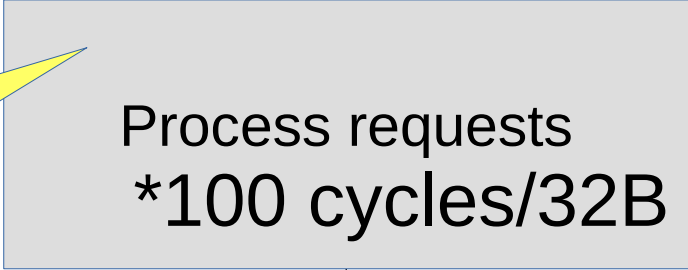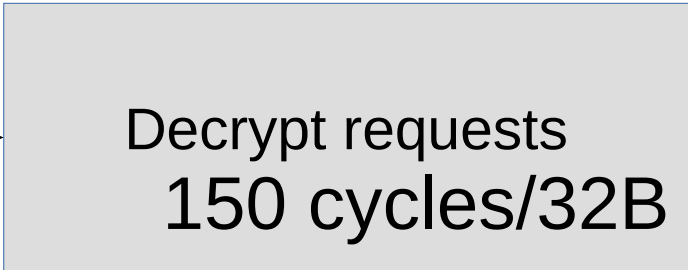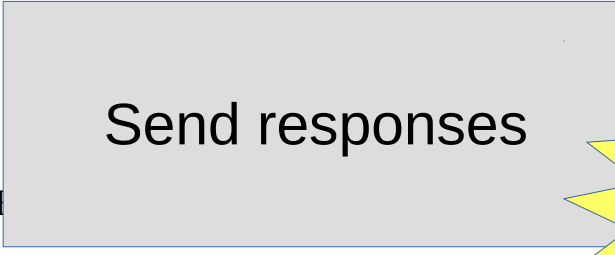
Overhead analysis

Untrusted (Host & OS) → **Trusted (Enclave)**

Host app → Wait for network requests

Enter enclave
~3,300 cycles

Decrypt requests
150 cycles/32B

Process requests
*100 cycles/32B

Exits causes indirect costs:
1.5X – 5X slower execution
FlexSC [OSDI'10]
syscall analysis

Exit enclave
~3,800 cycles

Send responses

Encrypt responses
*150 cycles/32B

Overhead analysis

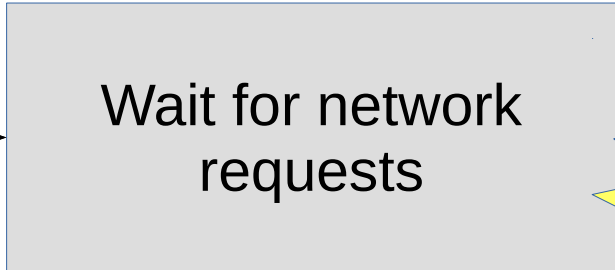**Host app** → | Wait for network requests |

Enter enclave ~3,300 cycles →

| Decrypt requests **150 cycles/32B** |

↓

| Process requests ***100 cycles/32B** |

↓

| Encrypt responses ***150 cycles/32B** |

Exit enclave ~3,800 cycles →

| Send responses |

> **Exits causes indirect costs:**
> **1.5X – 5X slower execution**
> **FlexSC [OSDI'10]**
> **syscall analysis**

Eleos does better!

Throughput: Slowdown factor
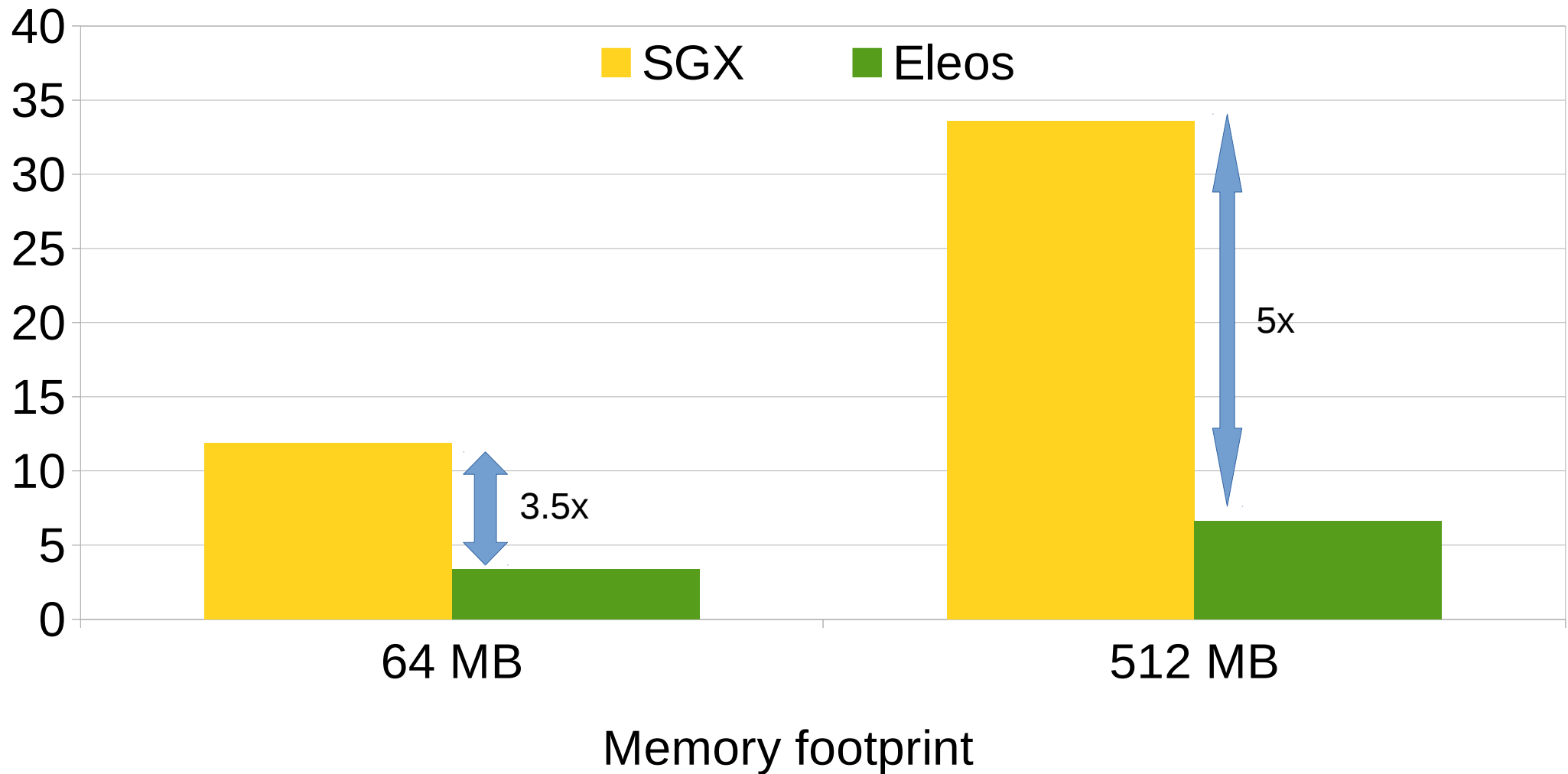
Eleos does better!

Throughput: Slowdown factor
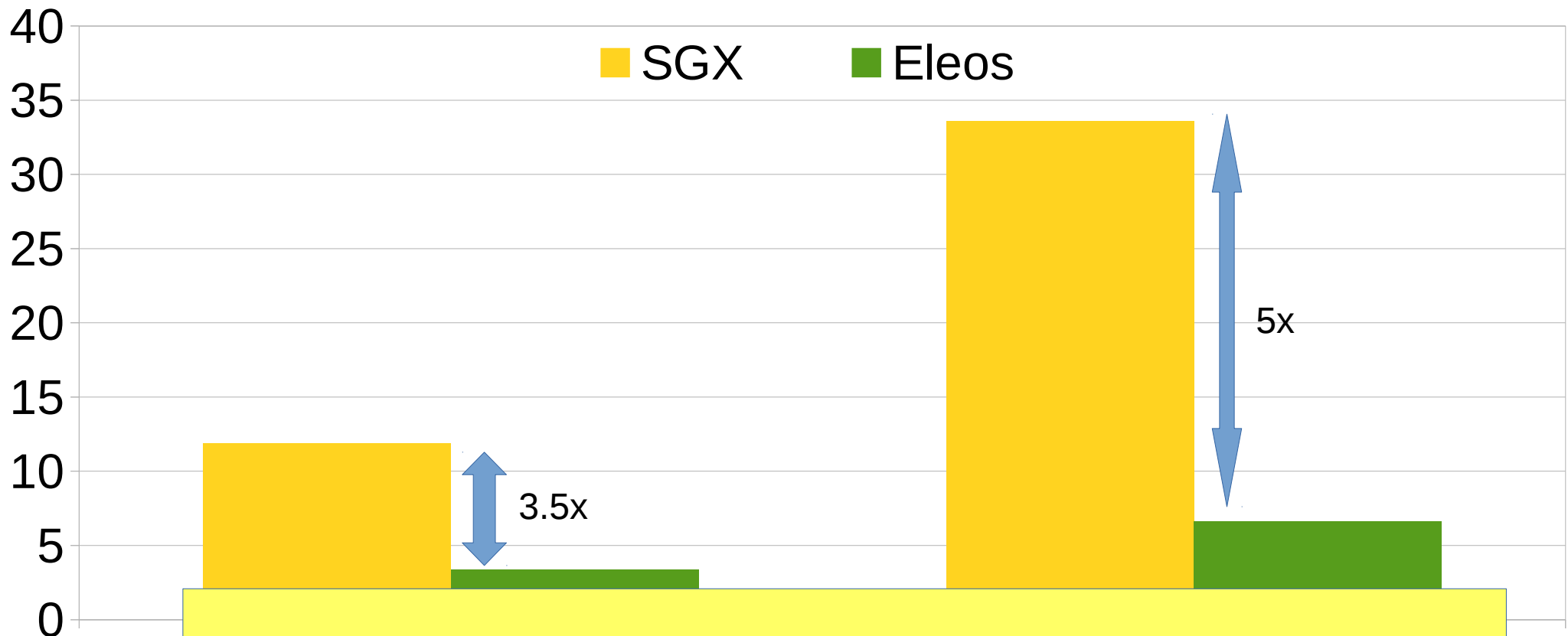
How does Eleos achieve this?

Eleos: **Exit-less** services

Exit-less system calls with RPC infrastructure

Exit-less SGX paging

Eleos: **Exit-less** services

Exit-less system calls with RPC infrastructure

Exit-less SGX paging

Background: SGX paging

System mem

SGX mem

Dedicated memory
Enclave code & data
Limited to 128 MB

Background: SGX paging

Enclave
Trusted

System mem

secret_foo():

...
*p = 1;

SGX mem

Untrusted

Background: SGX paging

Enclave
Trusted

System mem

secret_foo():

...

*p = 1;

Hardware
Address translation

SGX mem

Untrusted

Meni Orenbach, Technion

Background: SGX paging

Enclave
Trusted

System mem

secret_foo():

...

*p = 1;

SGX mem

Hardware
Address translation

Page table

Encrypted

Untrusted

Meni Orenbach, Technion

Background: SGX paging

Enclave
Trusted

System mem

secret_foo():

...

*p = 1;

Hardware
Address translation

SGX mem

Page table

Encrypted

Untrusted

Swapped-out

Meni Orenbach, Technion

Background: SGX paging

Enclave
Trusted

System mem

secret_foo():

...

*p = 1;

SGX mem

Hardware
Address translation

Page table

Encrypted

SGX-driver
Untrusted

Fault
handler

Swapped-out

Meni Orenbach, Technion

Background: SGX paging

Enclave
Trusted

System mem

secret_foo():

...

*p = 1;

Hardware
Address translation

SGX mem

Decrypted

Page table

Integrity
validation

SGX-driver
Untrusted

Fault
handler

Encrypted

Swapped-out

Meni Orenbach, Technion

Background: SGX paging

Enclave
Trusted

SGX driver
Untrusted

System mem

```
secret_foo():

...
*p = 1;
*(++p) = 2;
```

Hardware
Address translation

Page table

Fault
handler

SGX mem

Decrypted

Encrypted

Background: SGX paging

Enclave
Trusted

System mem

secret_foo():

...
*p = 1;
*(++p) = 2;

SGX mem

Hardware
Address translation

Decrypted

Fast path

Page table

Encrypted

Fault
handler

SGX driver
Untrusted

Background: SGX paging

Background: SGX paging

Enclave
Trusted

System mem

secret_foo():

...
*p = 1;
*(++p) = 2;

Hardware
Address translation

Page table

SGX mem

Decrypted

Fault
handler

Encrypted

SGX driver
Untrusted

SGX paging overheads

Enclave
Trusted

System mem

secret_foo():

...
*p = 1;
*(++p) = 2;

Hardware
Address translation

SGX mem

Decrypted

Page table

Enclave
exit

Enclave
resume

SGX driver
Untrusted

Fault
handler

Encrypted

Meni Orenbach, Technion

SGX paging overheads

Enclave
Trusted

Indirect costs

secret_foo():

...
*p = 1;
*(++p) = 2;

Hardware
Address translation

System mem

SGX mem

Decrypted

Page table

Enclave
exit

Enclave
resume

Fault
handler

Encrypted

SGX driver
Untrusted

SGX paging overheads

Enclave
Trusted

System mem

secret_foo():

...
*p = 1;
*(++p) = 2;

SGX mem

Decrypted

Indirect costs

Hardware
Address translation

Page table

Enclave
exit

Enclave
resume

Encrypted

SGX driver
Untrusted

Fault
handler

Overaheads: Untrusted software
manages enclave memory

Wanted: In-enclave virtual memory management

No more exits!

Ideal in-enclave VM management

Enclave Trusted

SGX driver Untrusted

secret_foo():

...
*p = 1;
*(++p) = 2;

Hardware Address translation

Page table

Fault handler

System mem

SGX mem

Ideal in-enclave VM management

Enclave Trusted

secret_foo():

...
*p = 1;
*(++p) = 2;

Hardware
Address translation

Page table

Fault
handler

System mem

SGX mem

Ideal in-enclave VM management

Enclave
Trusted

secret_foo():

...
*p = 1;
*(++p) = 2;

Hardware
Address translation

No available
hardware

Page table

Fault
handler

System mem

SGX mem

Ideal in-enclave VM management

System mem

secret_foo():

...
*p = 1;
*(++p) = 2;

**Software
Address translation**

Page table

Fault
handler

SGX mem

Meni Orenbach, Technion

SUVM: Secured user-space VM

Enclave
Trusted

secret_foo():
s_ptr<int> p =
suvm_malloc(1024);

...
*p = 1;

Software
Address translation

Page table

Fault
handler
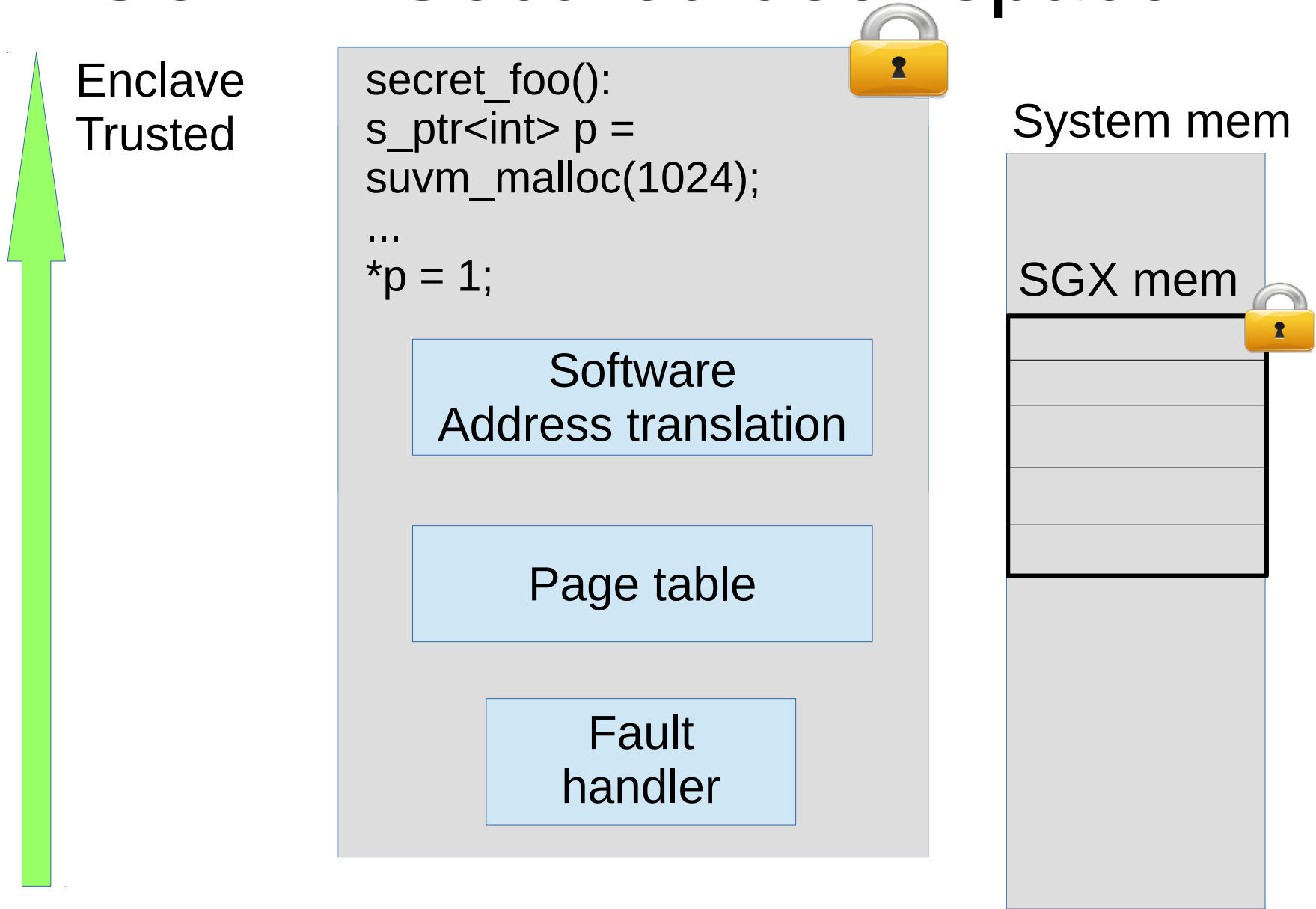
System mem

SGX mem

SUVM: Secured user-space VM

Enclave
Trusted

System mem

secret_foo():
s_ptr<int> p =
suvm_malloc(1024);
...
*p = 1;

Software
Address translation

Page table

Fault
handler

SGX mem

Template class:
SecuredPointer.

SUVM: Secured user-space VM

Enclave
Trusted

secret_foo():
s_ptr<int> p =
suvm_malloc(1024);
...
*p = 1;

System mem

SGX mem

Template class:
SecuredPointer.

Software
Address translation

Page table

Fault
handler

Encrypted

SUVM: Secured user-space VM

Enclave
Trusted

```
secret_foo():
s_ptr<int> p =
suvm_malloc(1024);
...
*p = 1;
```

System mem

SGX mem

Template class:
SecuredPointer.

Software
Address translation

Page table

Fault
handler

Encrypted

Swapped-out

Meni Orenbach, Technion

SUVM: Secured user-space VM

Enclave
Trusted

System mem

```
secret_foo():
s_ptr<int> p =
suvm_malloc(1024);
...
*p = 1;
```

SGX mem

**Template class:
SecuredPointer.**

Software
Address translation

Page table

Fault
handler

Encrypted

Swapped-out

Meni Orenbach, Technion

SUVM: Secured user-space VM

Enclave
Trusted

```
secret_foo():
s_ptr<int> p =
suvm_malloc(1024);
...
*p = 1;
```

System mem

SGX mem

Software
Address translation

Decrypted

Page table

Integrity
validation

Fault
handler

Encrypted

**Template class:
SecuredPointer.**

Swapped-out

SUVM: Secured user-space VM

Enclave
Trusted

```
secret_foo():
s_ptr<int> p =
suvm_malloc(1024);
...
*p = 1;
```

System mem

SGX mem

Software Address translation

Decrypted

**Template class:
SecuredPointer.**

Page table

Integrity
validation

Fault handler

Encrypted

**Control path
in-enclave**

Swapped-out

SUVM: Secured user-space VM

Enclave
Trusted

secret_foo():
s_ptr<int> p =
suvm_malloc(1024);

...
*p = 1;
*(++p) = 2;

Software
Address translation

Page table

Fault
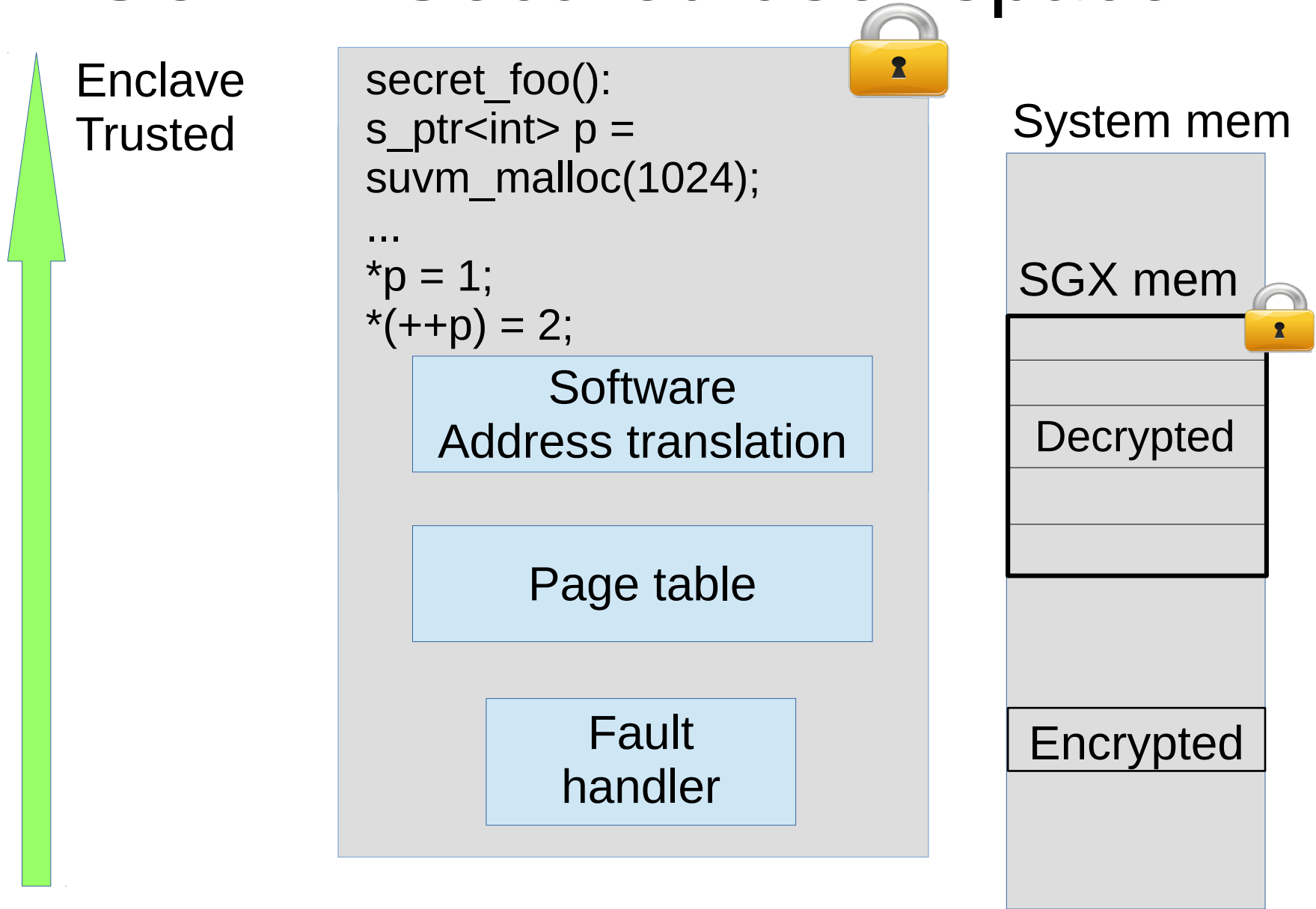handler

System mem

SGX mem

Decrypted

Encrypted

SUVM: Secured user-space VM

Enclave
Trusted

secret_foo():
s_ptr<int> p =
suvm_malloc(1024);

...
*p = 1;
*(++p) = 2;

Software
Address translation

Page table

Fault
handler

System mem

SGX mem

Decrypted

Encrypted

SUVM: Secured user-space VM

Enclave
Trusted

secret_foo():
s_ptr<int> p =
suvm_malloc(1024);

...
*p = 1;
*(++p) = 2;

Software
Address translation

Fast path
No page table
Lookup!

Page table

Fault
handler

System mem

SGX mem

Decrypted

Encrypted

Wait...Software based VM management?

Based on software address translation on GPUs, ActivePointers [ISCA'2016]

SUVM key contributions

- Multi-threaded

<u>Compared to SGX:</u>

Fast path: up to 20% overheads

Slow path: Eliminates costs of exits

	1 Thread	4 Threads
READ	5.5x	7x
WRITE	3.5x	5.9x

Throughput speedup

Software address translation offers new optimizations

- Customized page size

- Customized eviction policy

- Multi-enclave memory coordination

- Write-back only dirty pages

- Sub-page direct access to backing store

Software address translation offers new optimizations

- Customized page size

- Customized eviction policy

  Virtual Machine ballooning

- Multi-enclave memory coordination

- Write-back only dirty pages

- Sub-page direct access to backing store

<u>Software</u> address translation offers new optimizations

- Customized page size

- Customized eviction policy

- Multi-enclave memory coordination

- Write-back only dirty pages

- Sub-page direct access to backing store

Virtual Machine ballooning

- Background
- Motivation
- Overhead analysis
- Eleos design
- **Evaluation**

Meni Orenbach, Technion

Biometric Identity checking server

Workload
generator

Face
verification
server

+
ID

10Gb NIC

? =
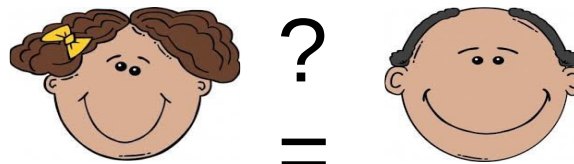
450MB DB
(5X SGX mem)

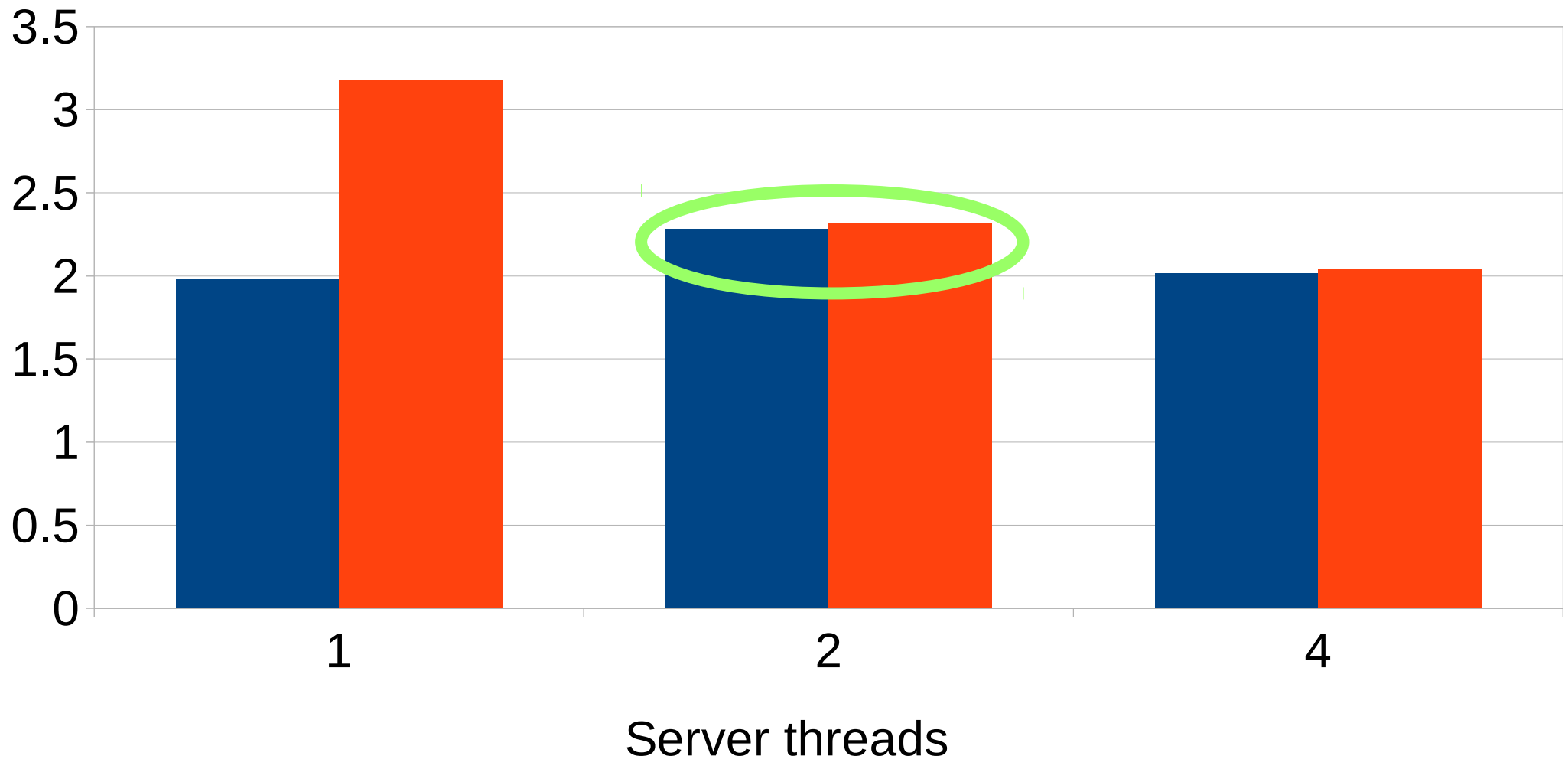# Biometric Identity validating server
Speedup compared to vanilla SGX
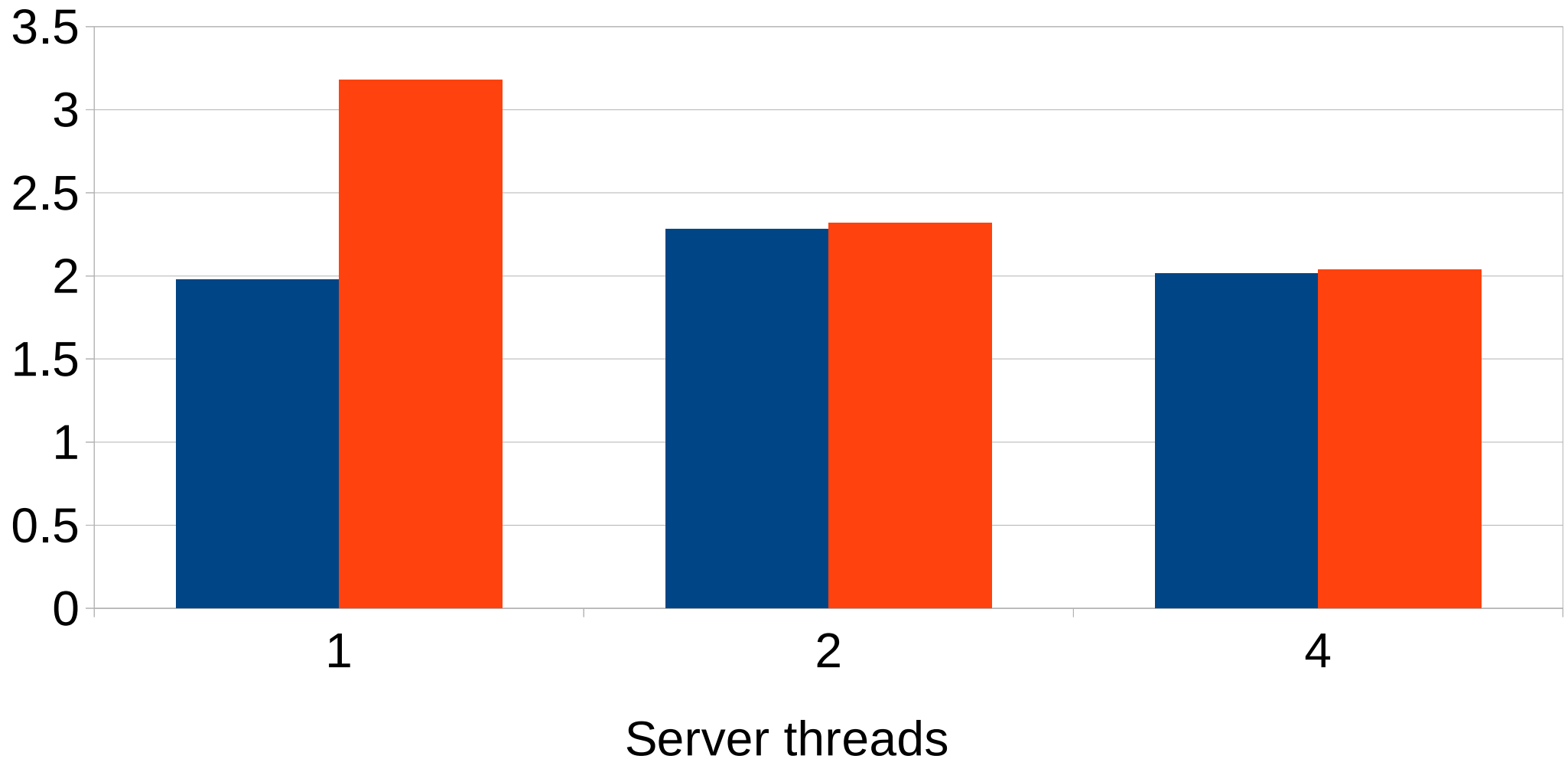
■ Eleos ■ Native

Server threads

Biometric Identity validating server

Speedup compared to vanilla SGX

■ Eleos   ■ Native

Server threads

Biometric Identity validating server

Speedup compared to vanilla SGX

Eleos scales better than vanilla-SGX:
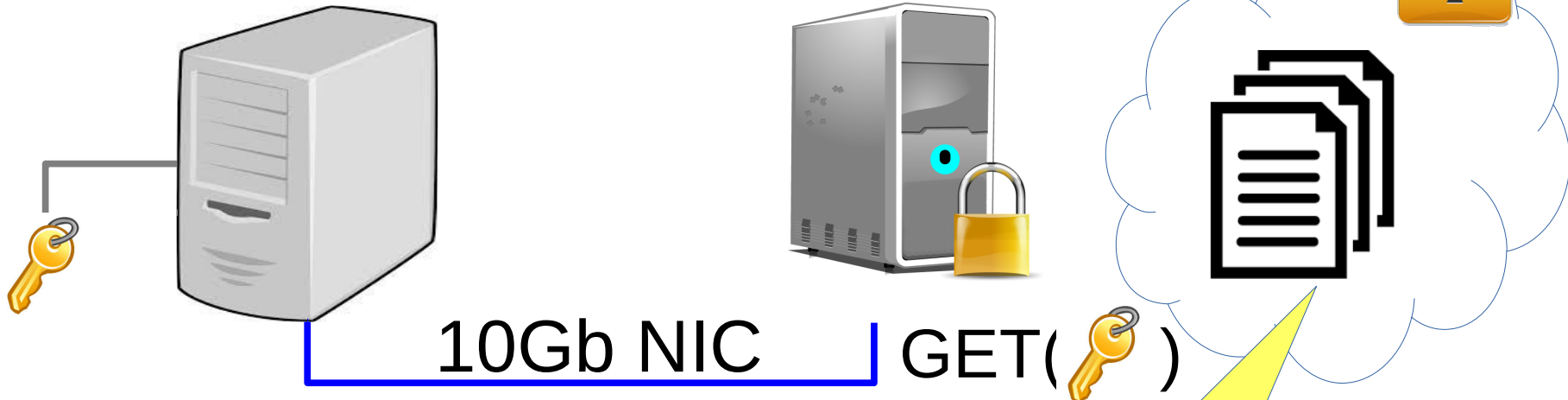Saves inter-processor-interrupts

Server threads
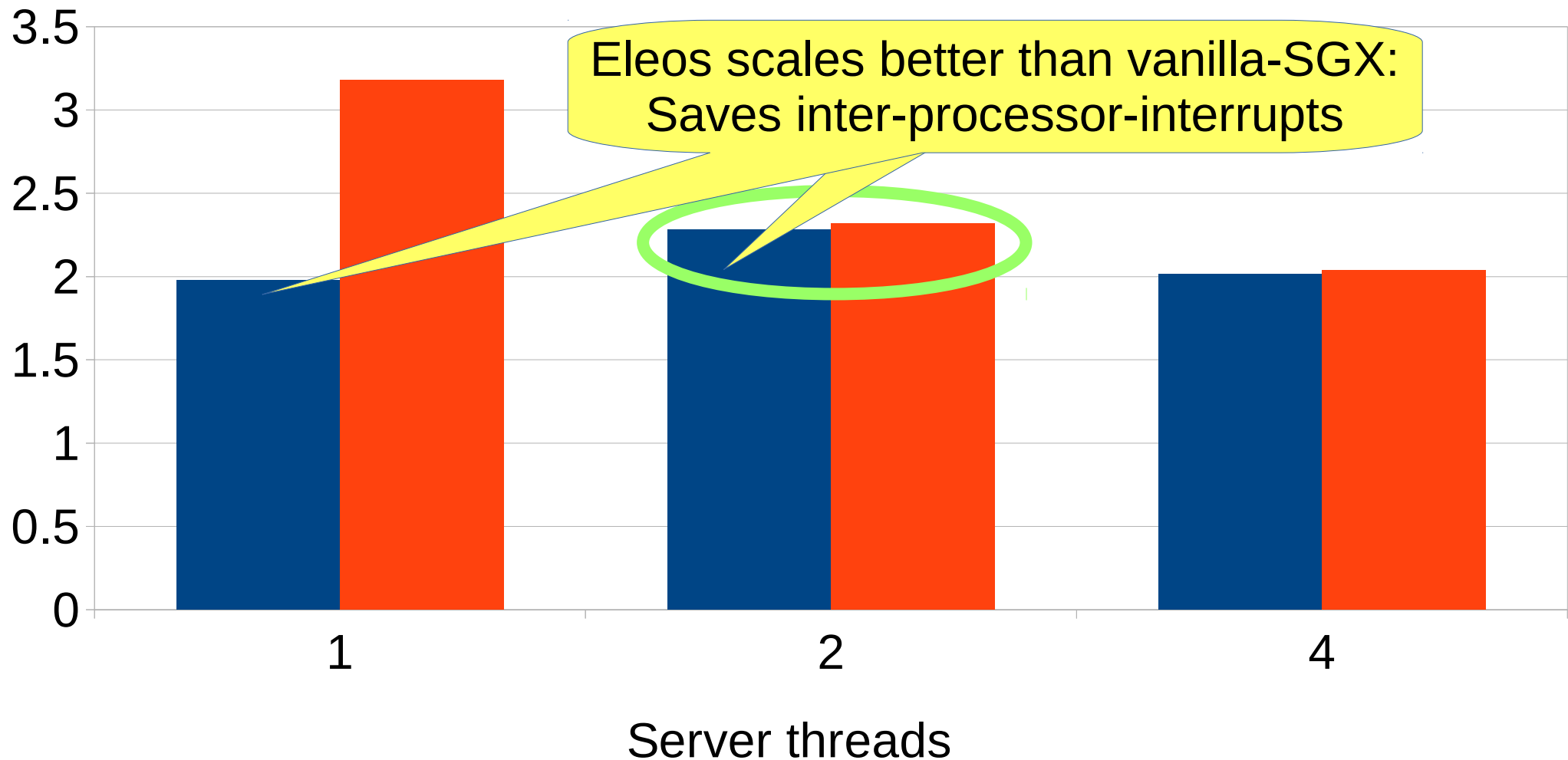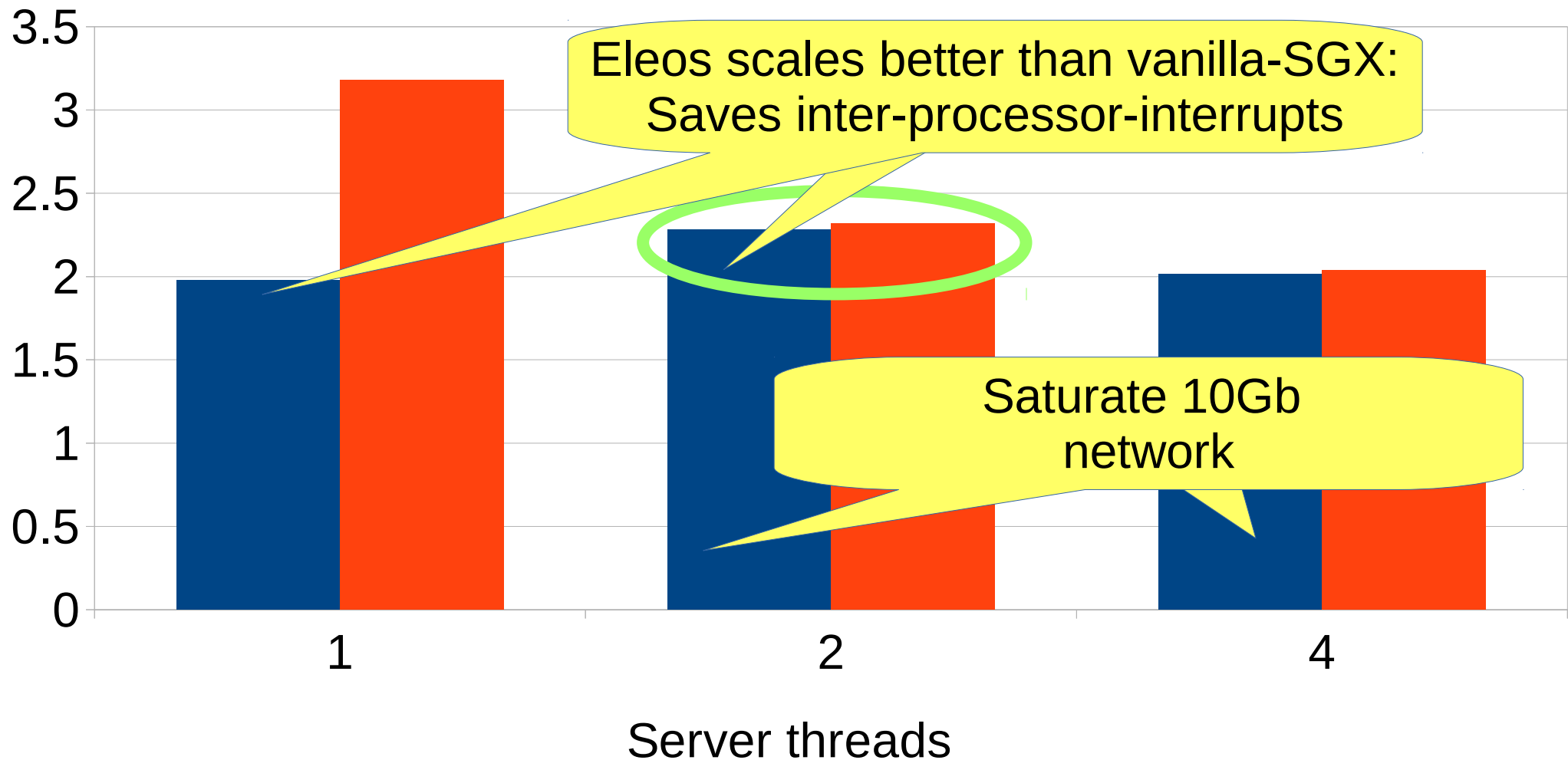
Memcached

Workload
Generator
(memaslap)

Memcached
Graphene LibOS
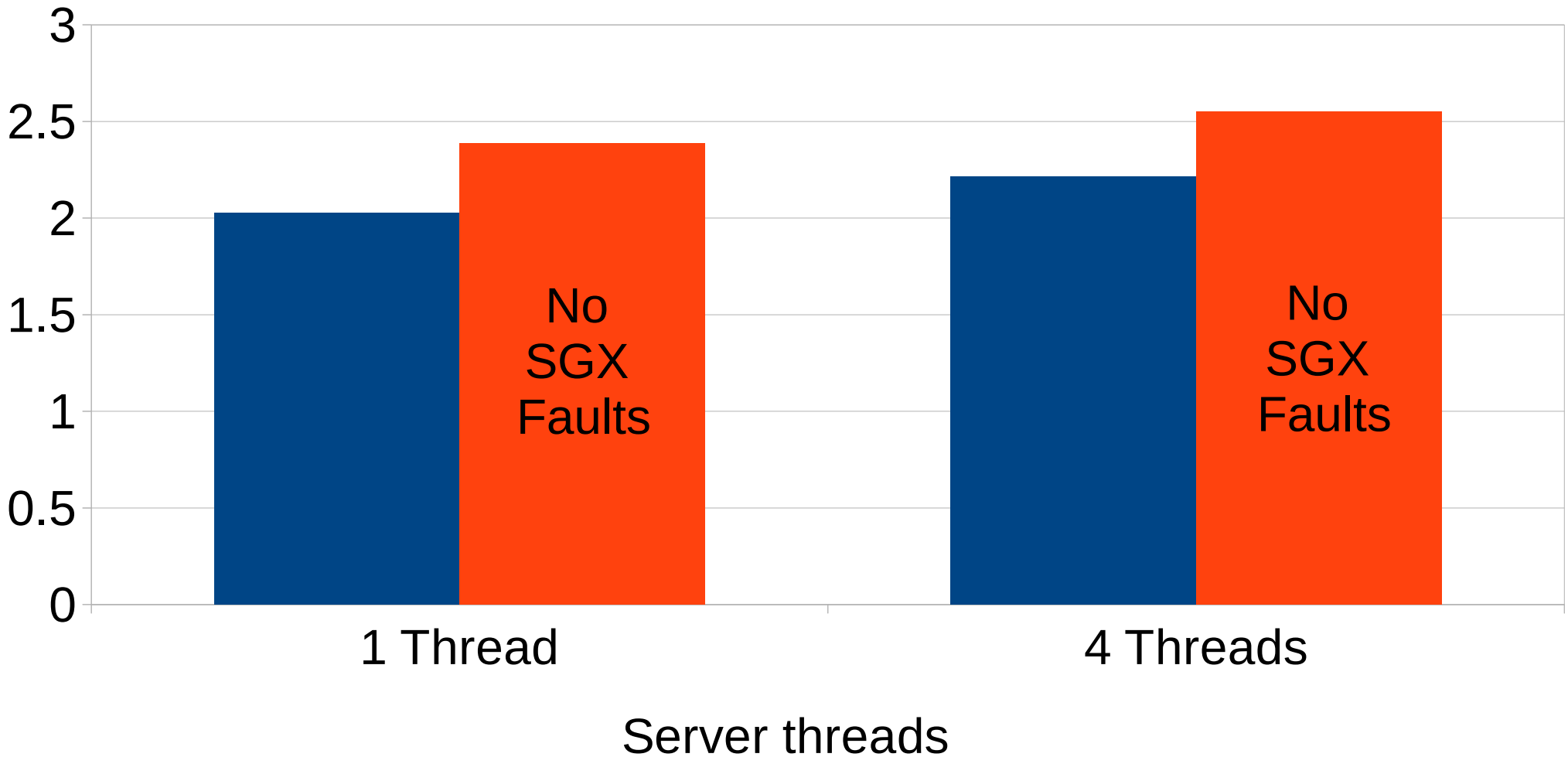[Eurosys'2014]

10Gb NIC   GET( )

~75 LOC
modification
for SUVM

500MB DB
(5.5X SGX mem)

Memcached

Speedup compared to vanilla SGX (500 MB)
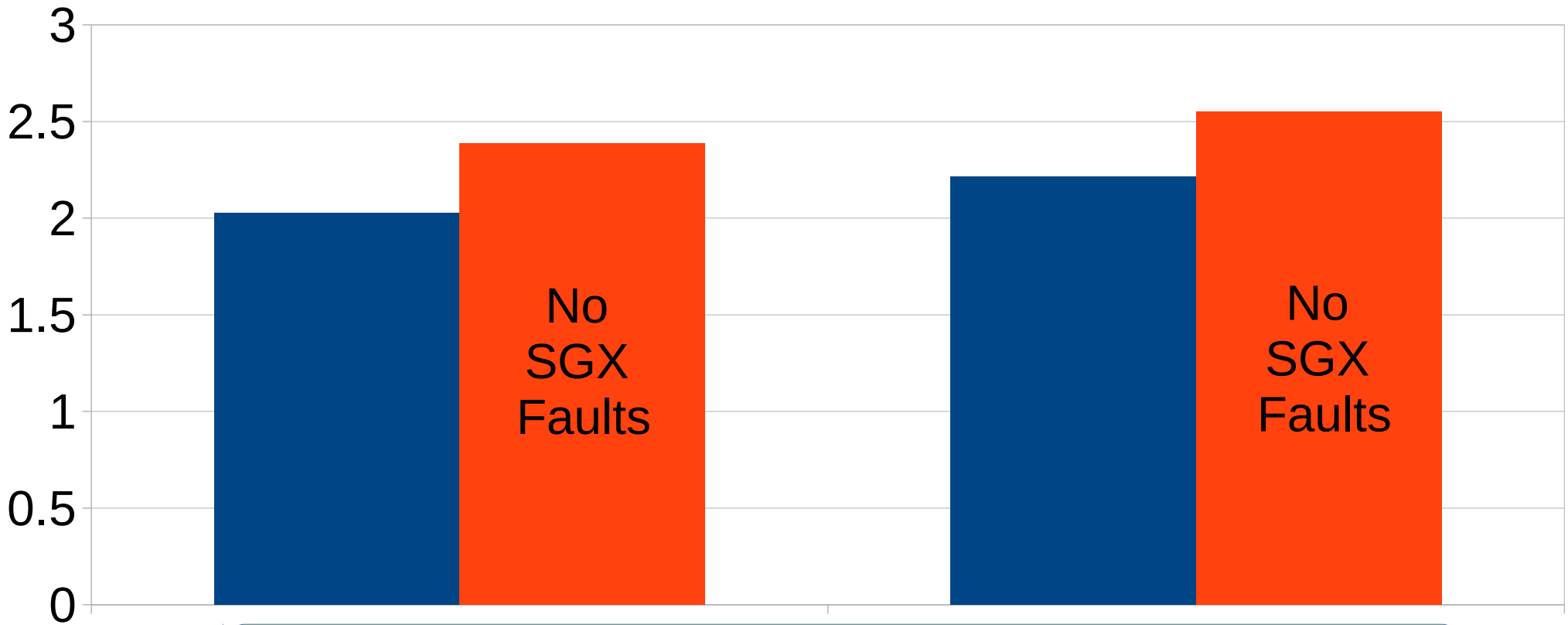
■ Eleos (500MB DB)   ■ vanilla SGX (20MB DB)

Memcached

Speedup compared to vanilla SGX (500 MB)

■ Eleos (500MB DB)   ■ vanilla SGX (20MB DB)
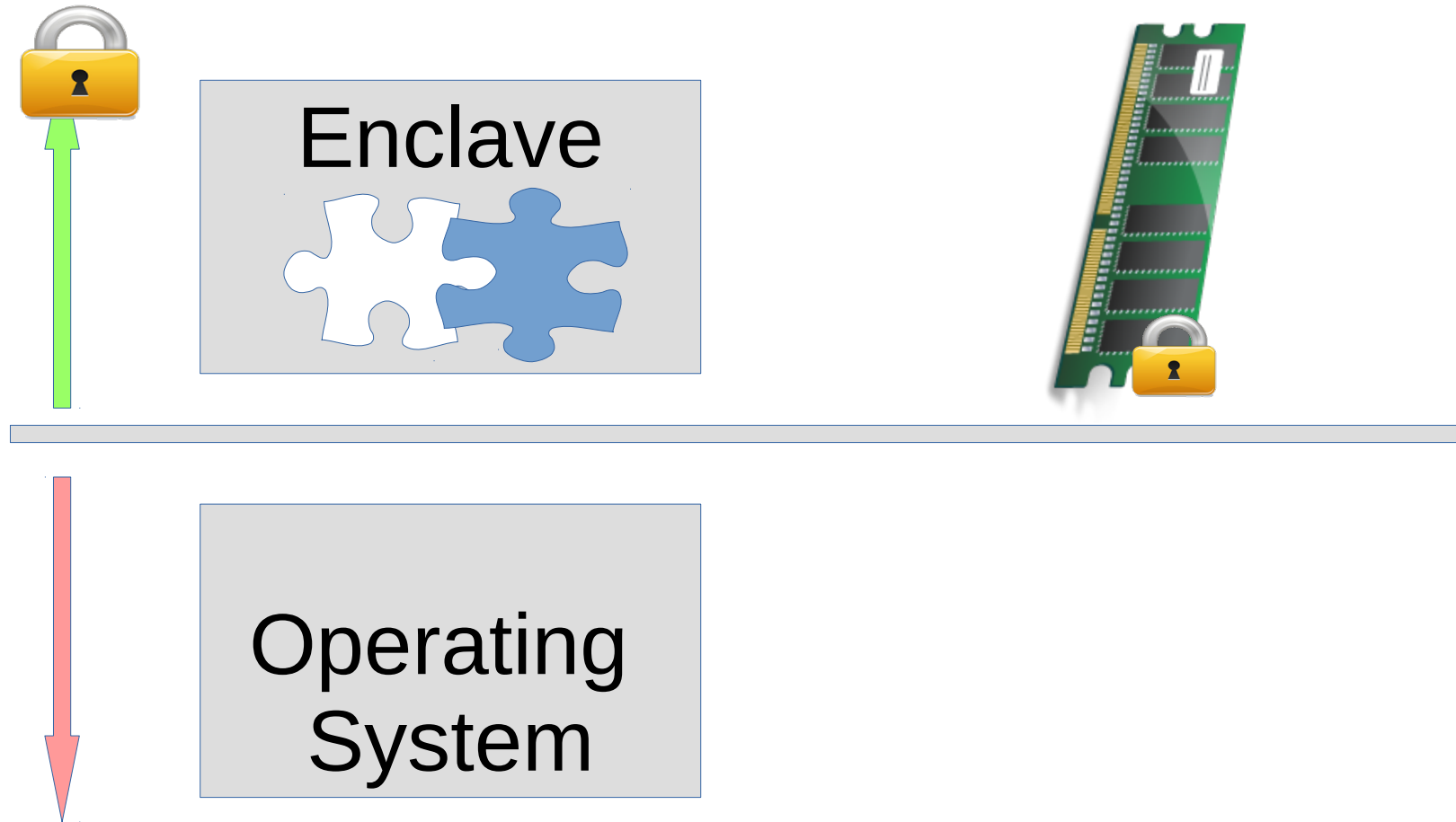

No SGX Faults

No SGX Faults

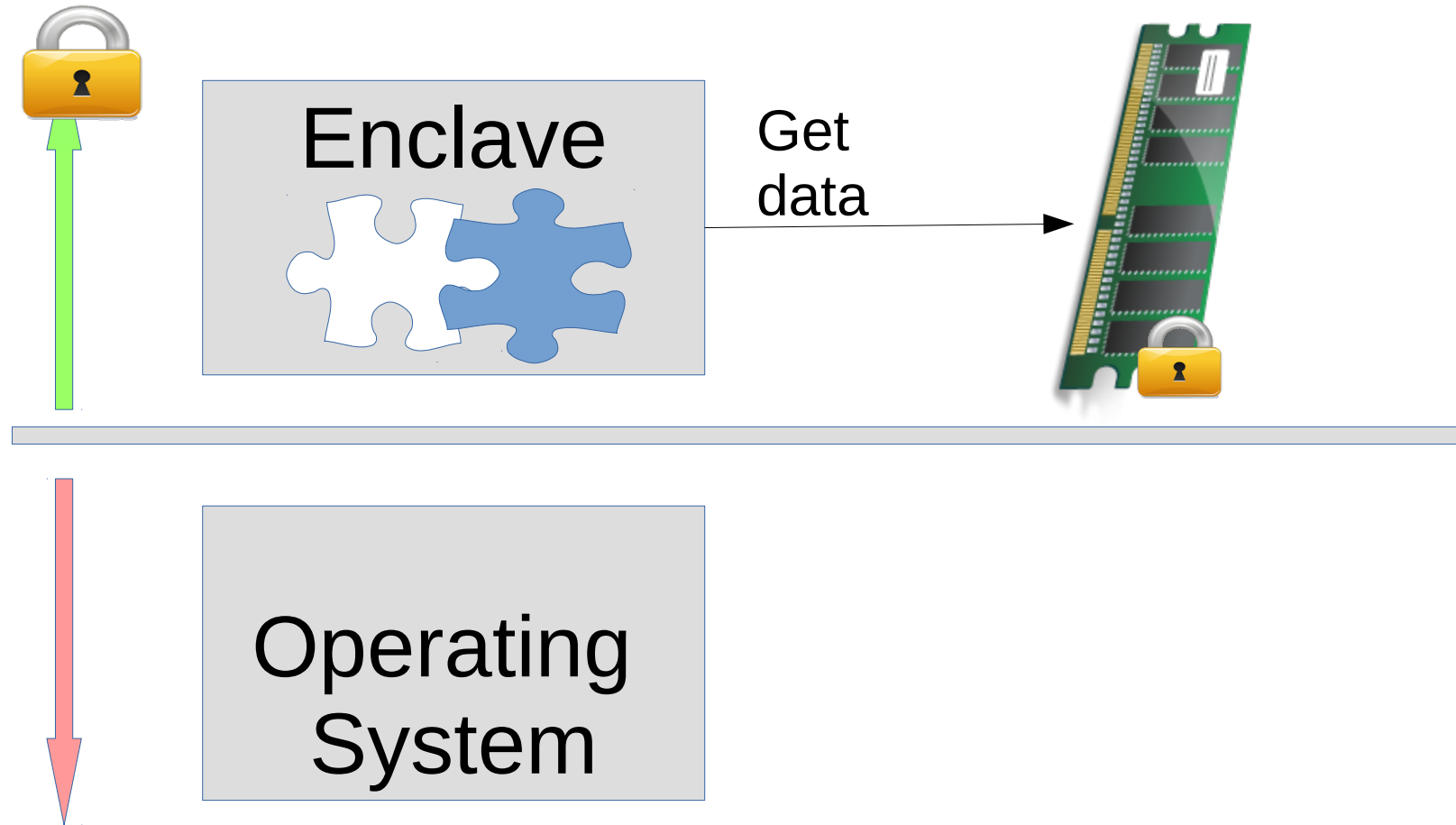Disclaimer: Eleos+Graphene is 3x slower than native

Take aways

- Eleos eliminates enclave exits costs
- Eleos available for Windows and Linux
  - Makes memory demanding applications available on Windows today
- Eleos takes a modularize approach
  - Memory demanding app? Link to SUVM
  - I/O intensive app? Link to RPC
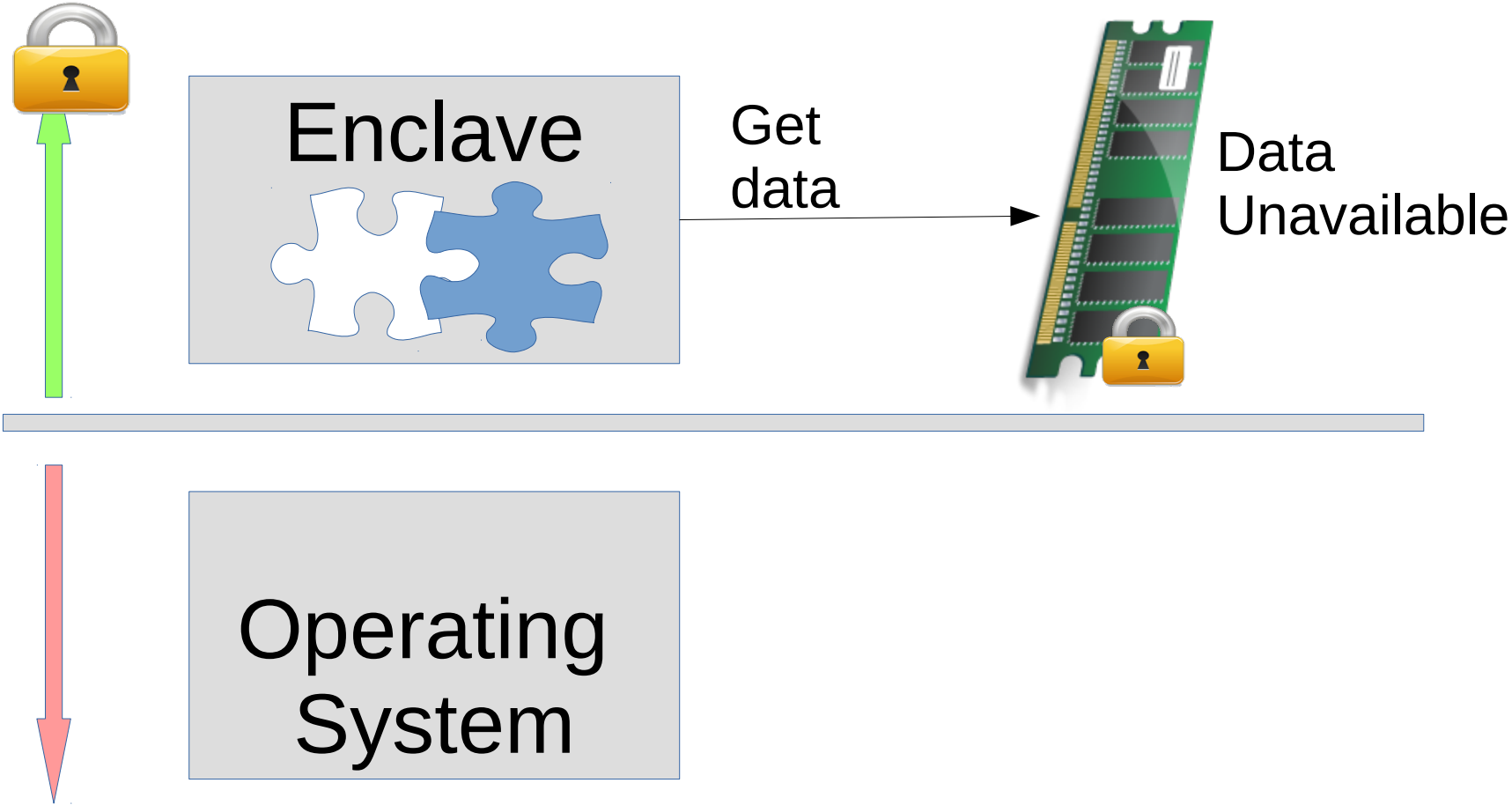 - Maintaining <u>small TCB</u>
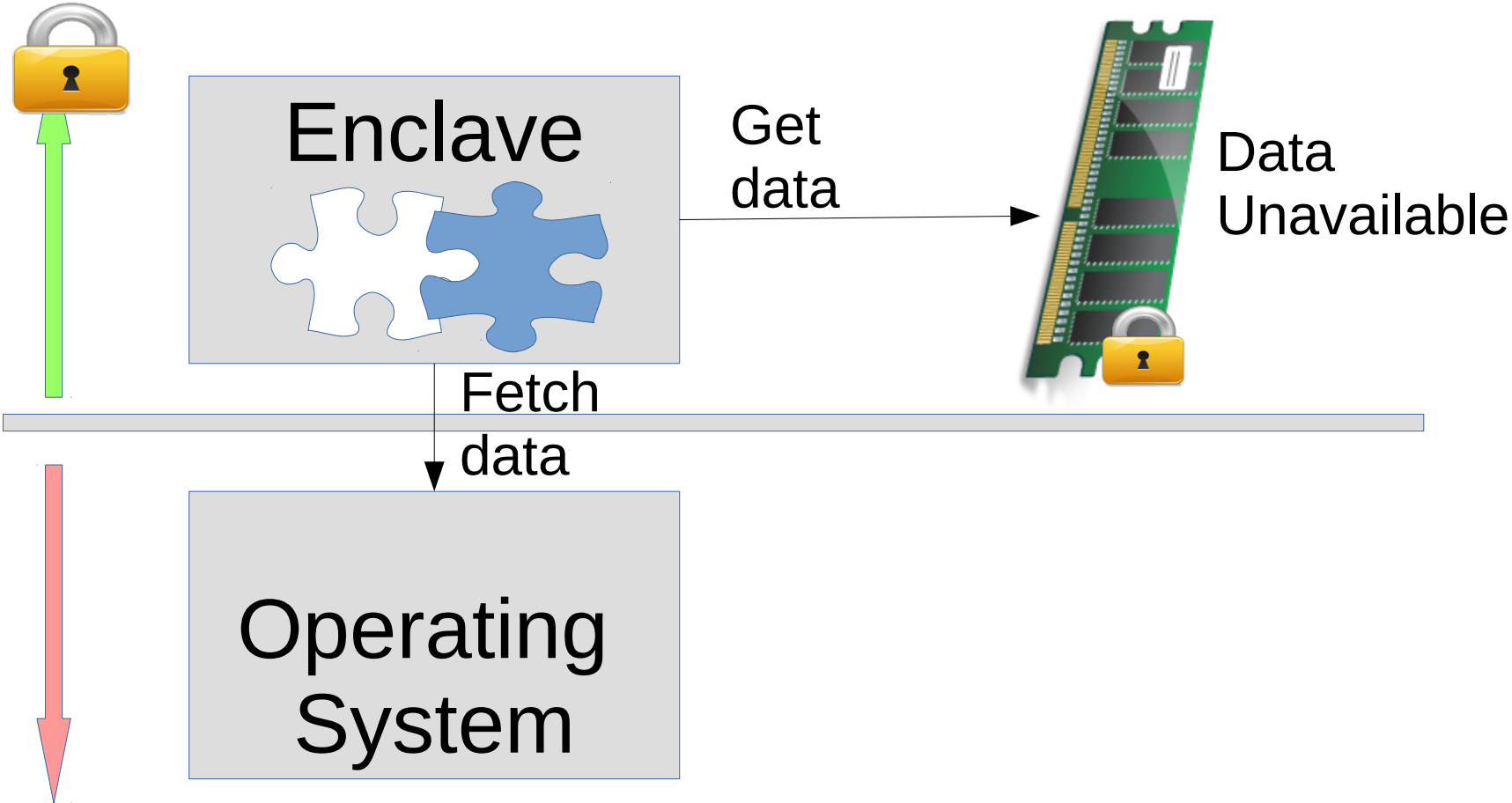
Traditional SGX:
Host-centric OS services

Enclave

Operating
System

Traditional SGX:
Host-centric OS services

Enclave

Get data

Operating System

Traditional SGX:
Host-centric OS services

Enclave

Get data

Data Unavailable

Operating System

Traditional SGX: Host-centric OS services

Enclave

Get data

Data Unavailable

Fetch data

Operating System

Traditional SGX:
Host-centric OS services

Enclave

Get data

Data Unavailable

Fetch data

Operating System

Loading
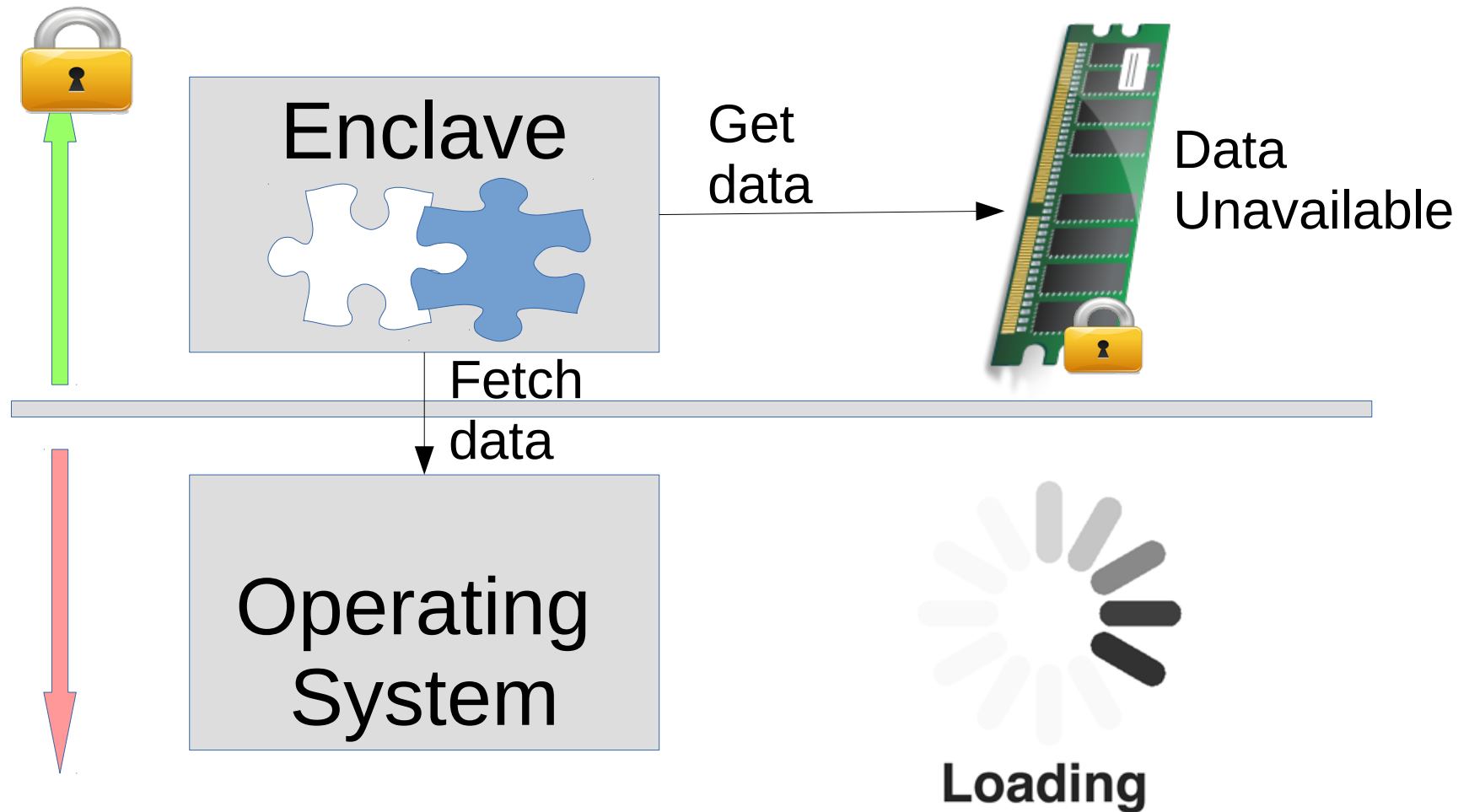
Eleos Insight:
Enclave-centric OS services

Enclave

In-enclave
Services

Get
data

Fetch data
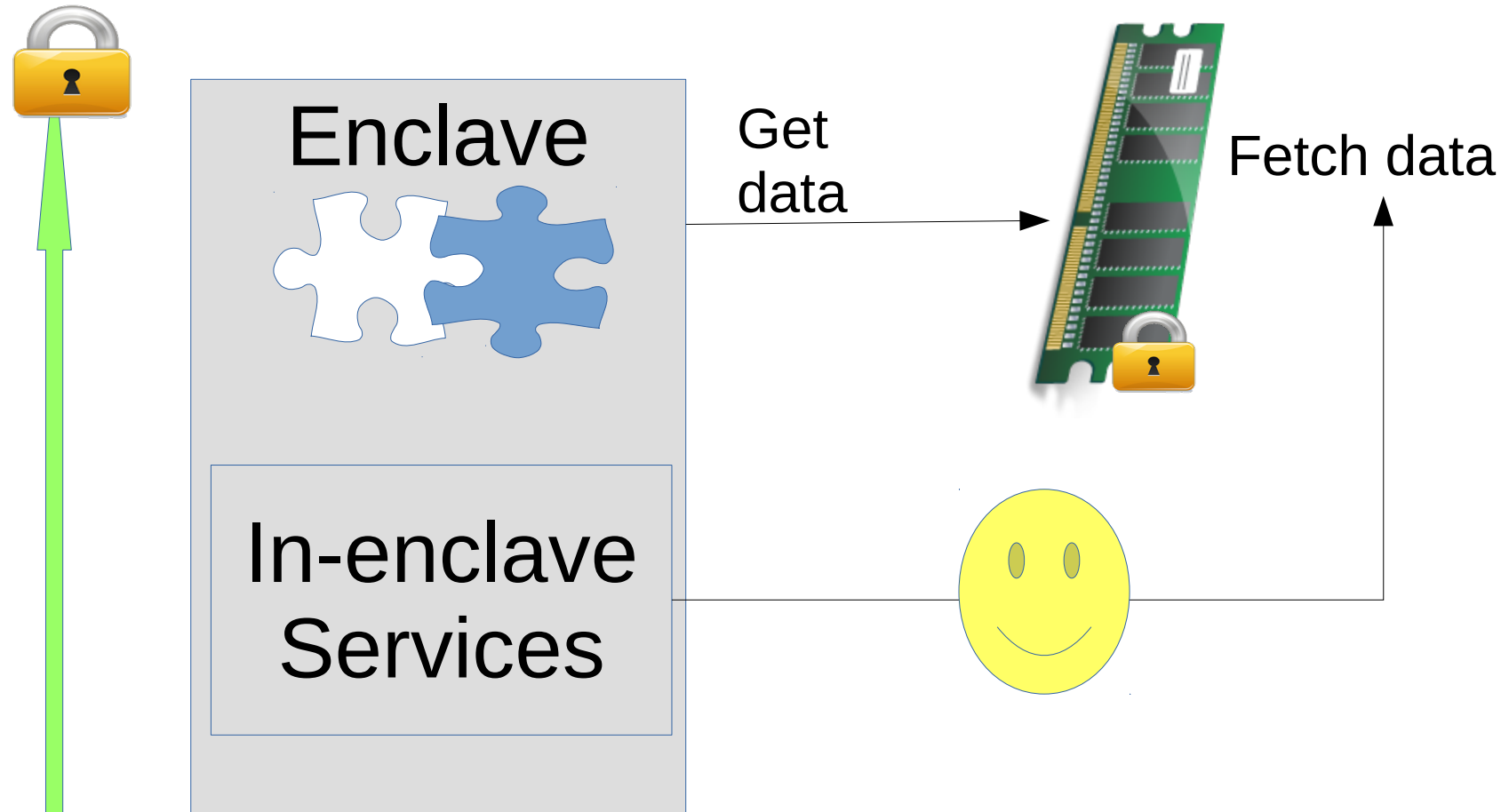
Take aways (2)

- **Eleos adapts 'accelerator-centric management'**
  - System calls: GPUfs [ASPLOS'13], GPUnet [OSDI'14]
  - Virtual memory: ActivePointers [ISCA'16]

- **We can do more!**

  - Asynchronous DMA host copies
  - Non-blocking enclave launches

  More information at:

  "SGX Enclaves as Accelerators" [Systex'16]

Thank you

Code is available at:
https://github.com/acsl-technion/eleos

shmeni@tx.technion.ac.il

Backup slides