Eleos: Exit-Less OS Services for SGX Enclaves

Meni Orenbach
Marina Minkin
Pavel Lifshits
Mark Silberstein

Accelerated Computing Systems Lab

Haifa, Israel

Systor2017 May 22 – 24 Haifa, Israel

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

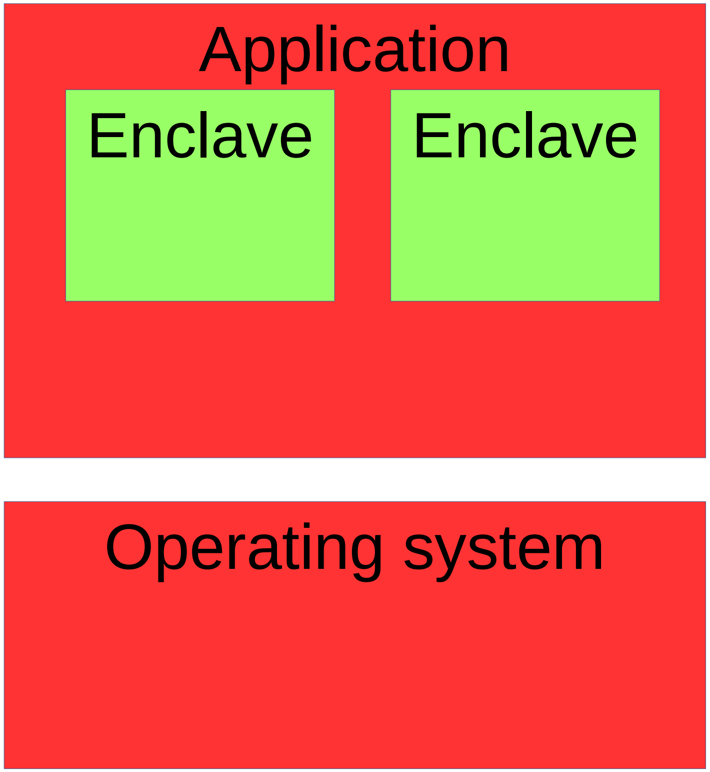
Available for Linux, Windows*

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

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

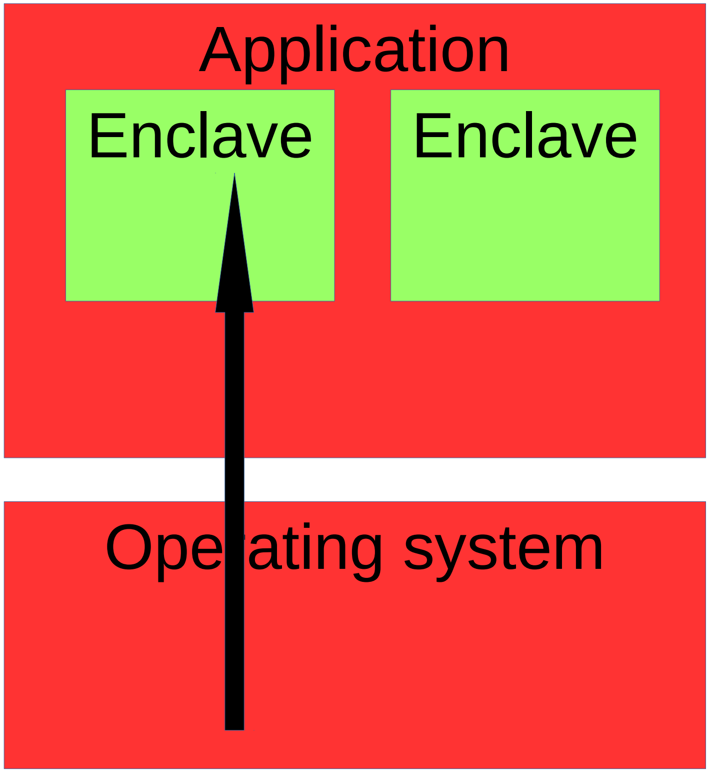
SGX enclaves are already here!

- Secured execution environment
- Reversed sandbox
- Small TCB
- Private code & data
  - Confidentiality
  - Integrity
  - Freshness
- Only CPU is trusted



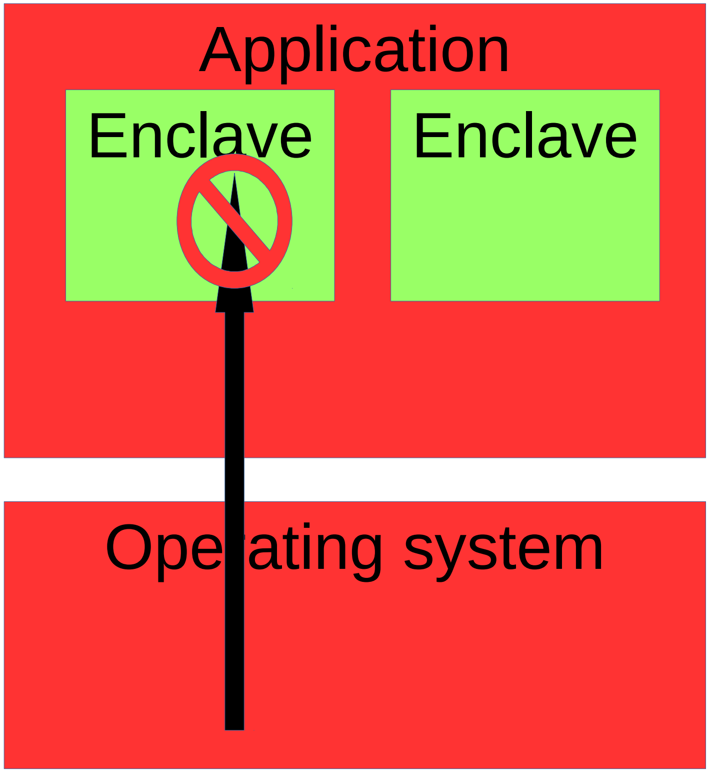
SGX enclaves are already here!

- Secured execution environment
- Reversed sandbox
- Small TCB
- Private code & data
  - Confidentiality
  - Integrity
  - Freshness
- Only CPU is trusted



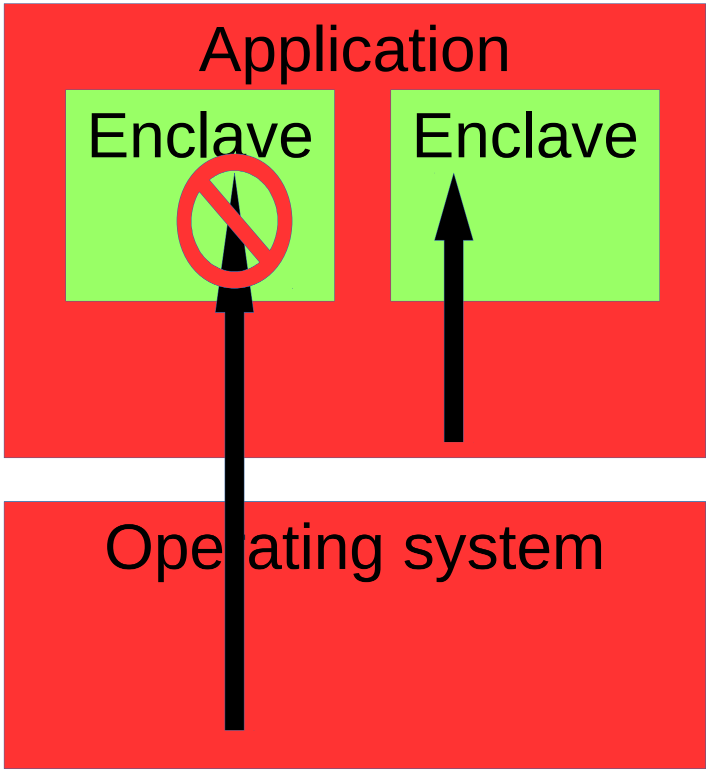
SGX enclaves are already here!

- Secured execution environment
- Reversed sandbox
- Small TCB
- Private code & data
  - Confidentiality
  - Integrity
  - Freshness
- Only CPU is trusted



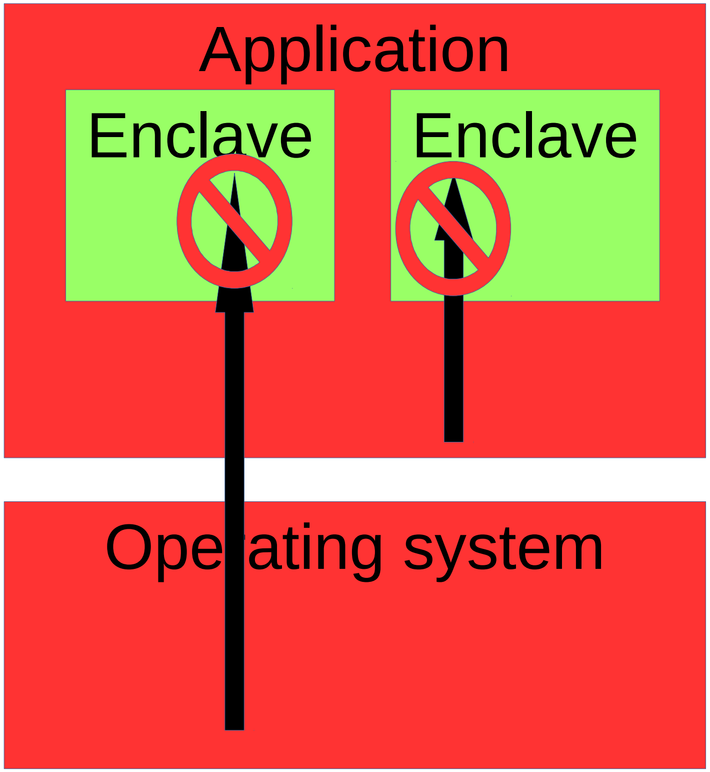
SGX enclaves are already here!

- Secured execution environment
- Reversed sandbox
- Small TCB
- Private code & data
  - Confidentiality
  - Integrity
  - Freshness
- Only CPU is trusted



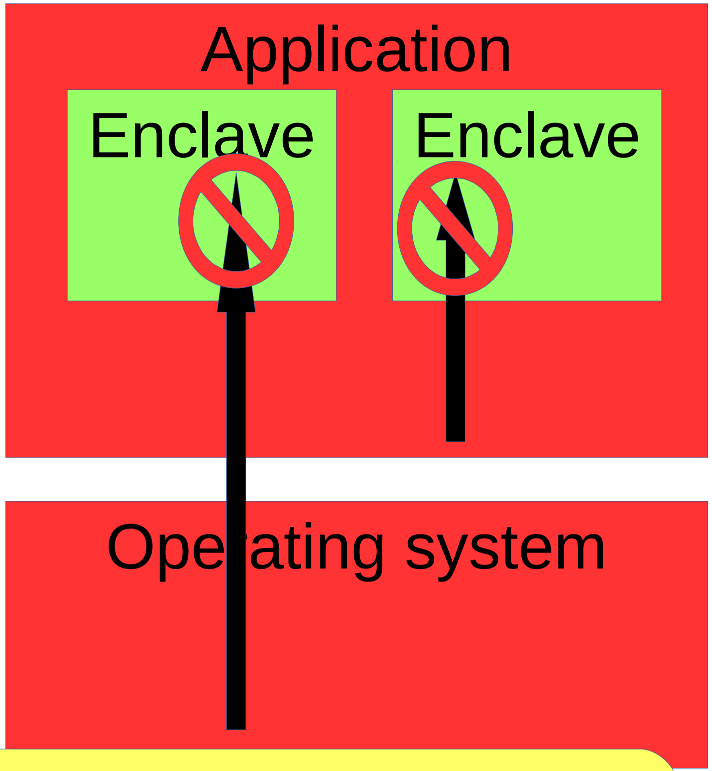
SGX enclaves are already here!

- Secured execution environment
- Reversed sandbox
- Small TCB
- Private code & data
  - Confidentiality
  - Integrity
  - Freshness
- Only CPU is trusted



SGX enclaves are already here!

- Secured execution environment
- Reversed sandbox
- Small TCB
- Private code & data
  - Confidentiality
  - Integrity
  - Freshness
- O

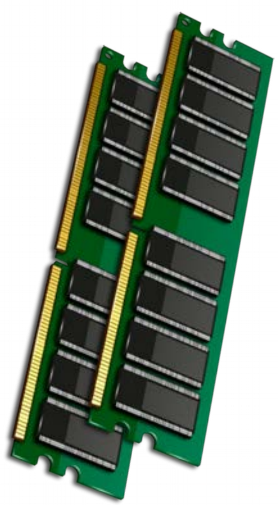


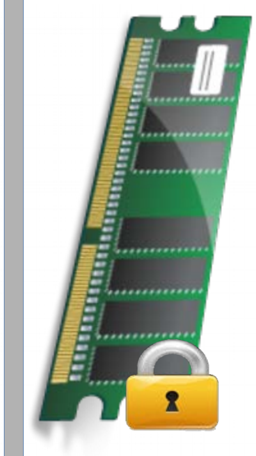
Lets look at
How to secure server applications with enclaves

Background:
Lifetime of a secured server

Untrusted (Host & OS)

Trusted (Enclave)

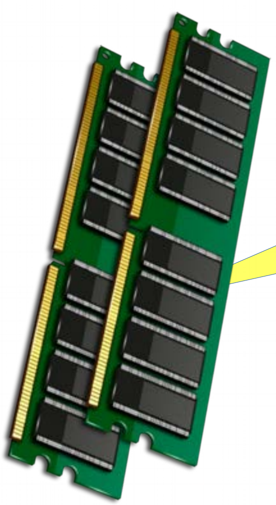




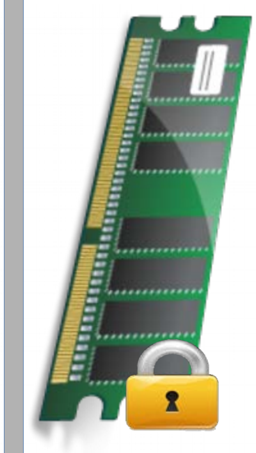
Background:
Lifetime of a secured server

Untrusted (Host & OS)

Trusted (Enclave)



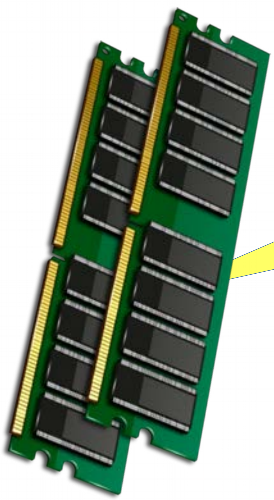
Untrusted memory
Unsecured access



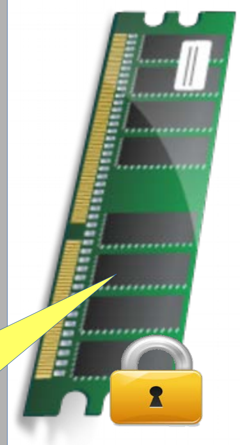
Background: Lifetime of a secured server

Untrusted (Host & OS)

Trusted (Enclave)



Untrusted memory
Unsecured access



Dedicated SGX mem
Limited to: 128 MB
Secured access

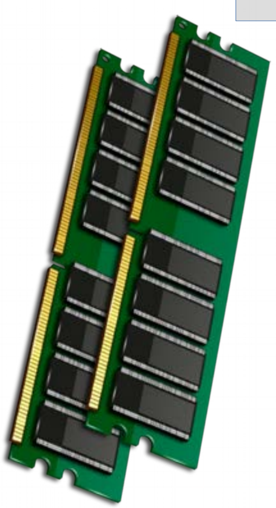
Background: Lifetime of a secured server

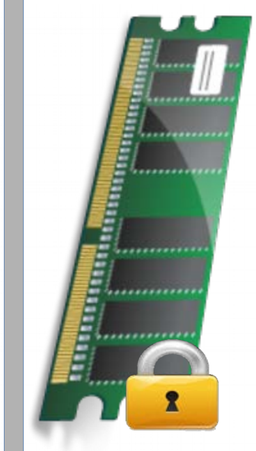
Untrusted (Host & OS)

Trusted (Enclave)

Host app

Wait for network requests





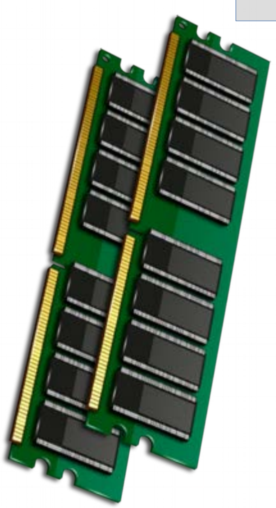
Background:
Lifetime of a secured server

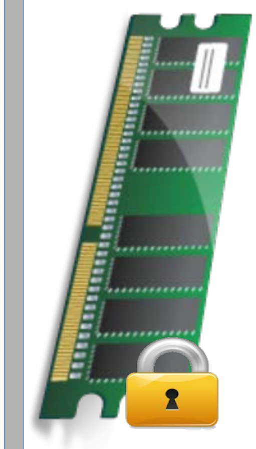
Untrusted (Host & OS)

Trusted (Enclave)

Host app

Wait for network requests

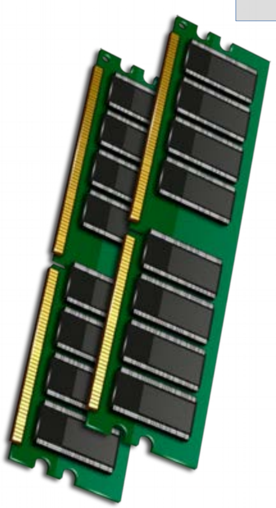


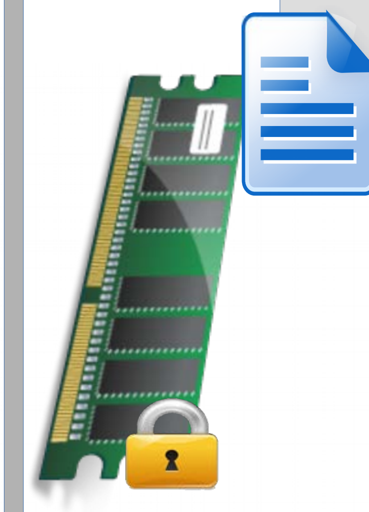


Background:
Lifetime of a secured server

Untrusted (Host & OS) | Trusted (Enclave)

Host app → **Wait for network requests** → Enter enclave → **Decrypt requests**





Background:
Lifetime of a secured server

Untrusted (Host & OS)

Trusted (Enclave)

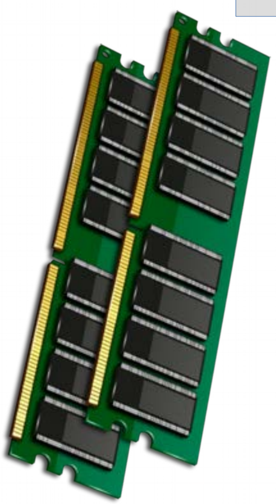
Host app

Wait for network requests

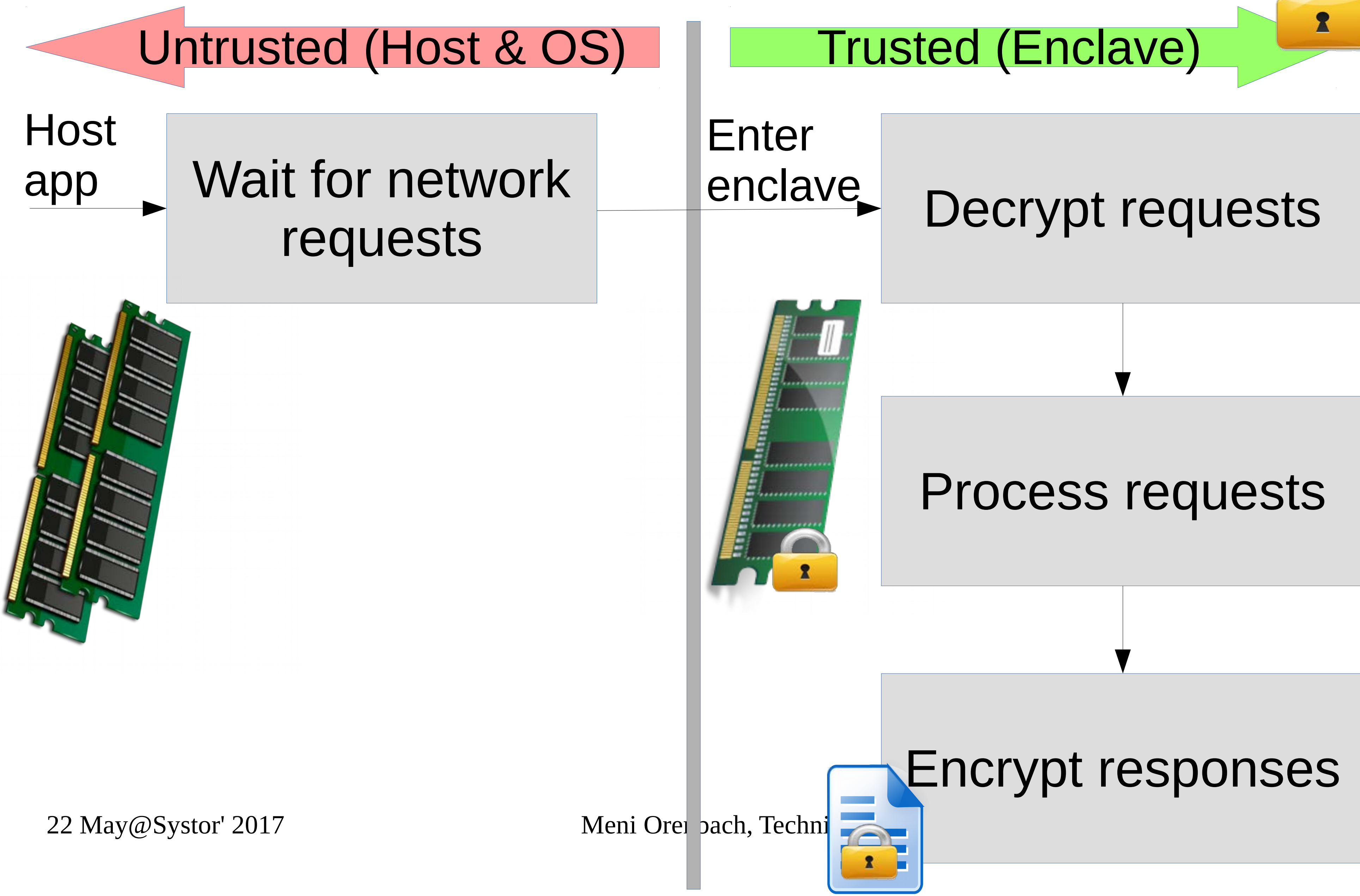
Enter enclave

Decrypt requests

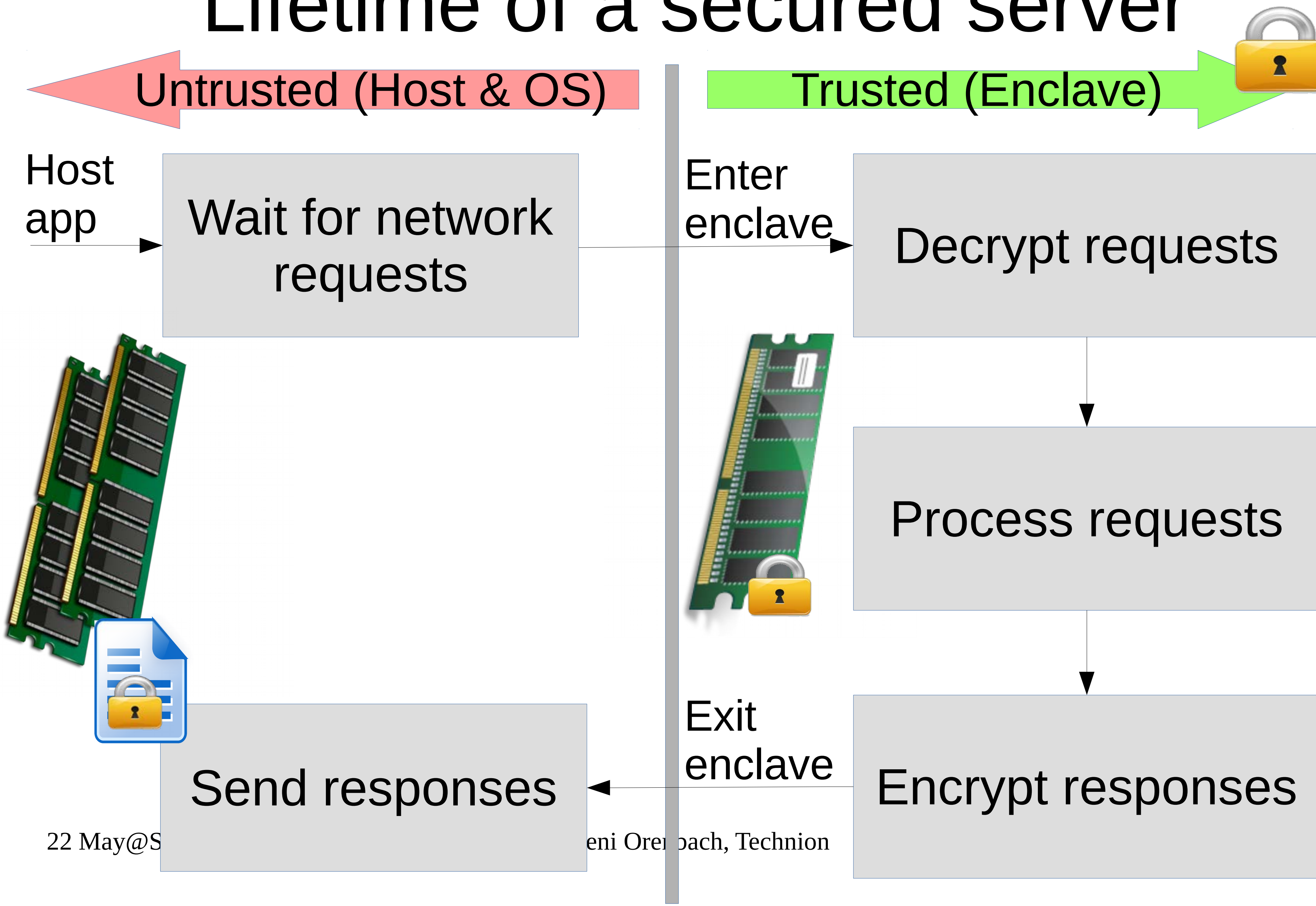
Process requests



Background:
Lifetime of a secured server

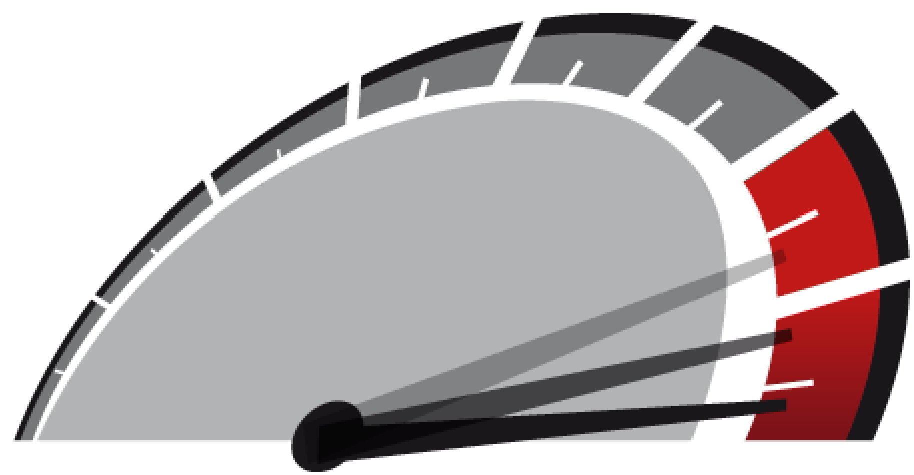


Background: Lifetime of a secured server



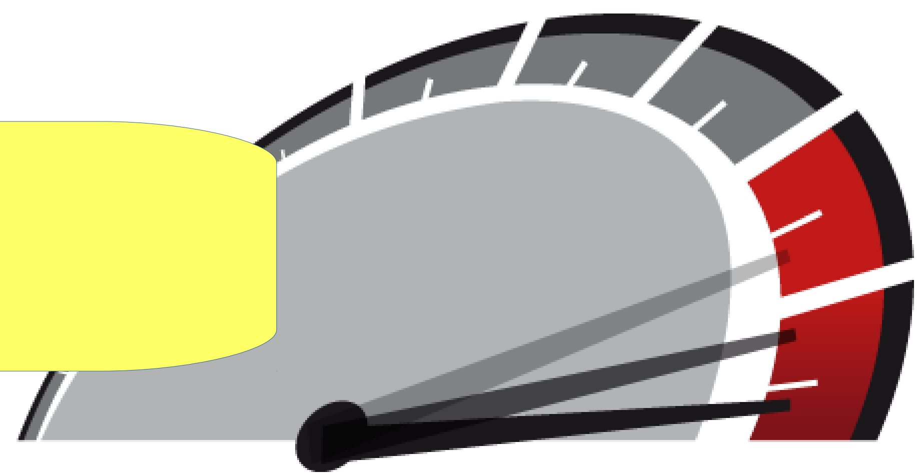
SGX enclaves should be fast

- ISA extensions
- Implemented in HW & Firmware
- Same CPU HW
- In-cache execution suffers no overheads



SGX enclaves should be fast

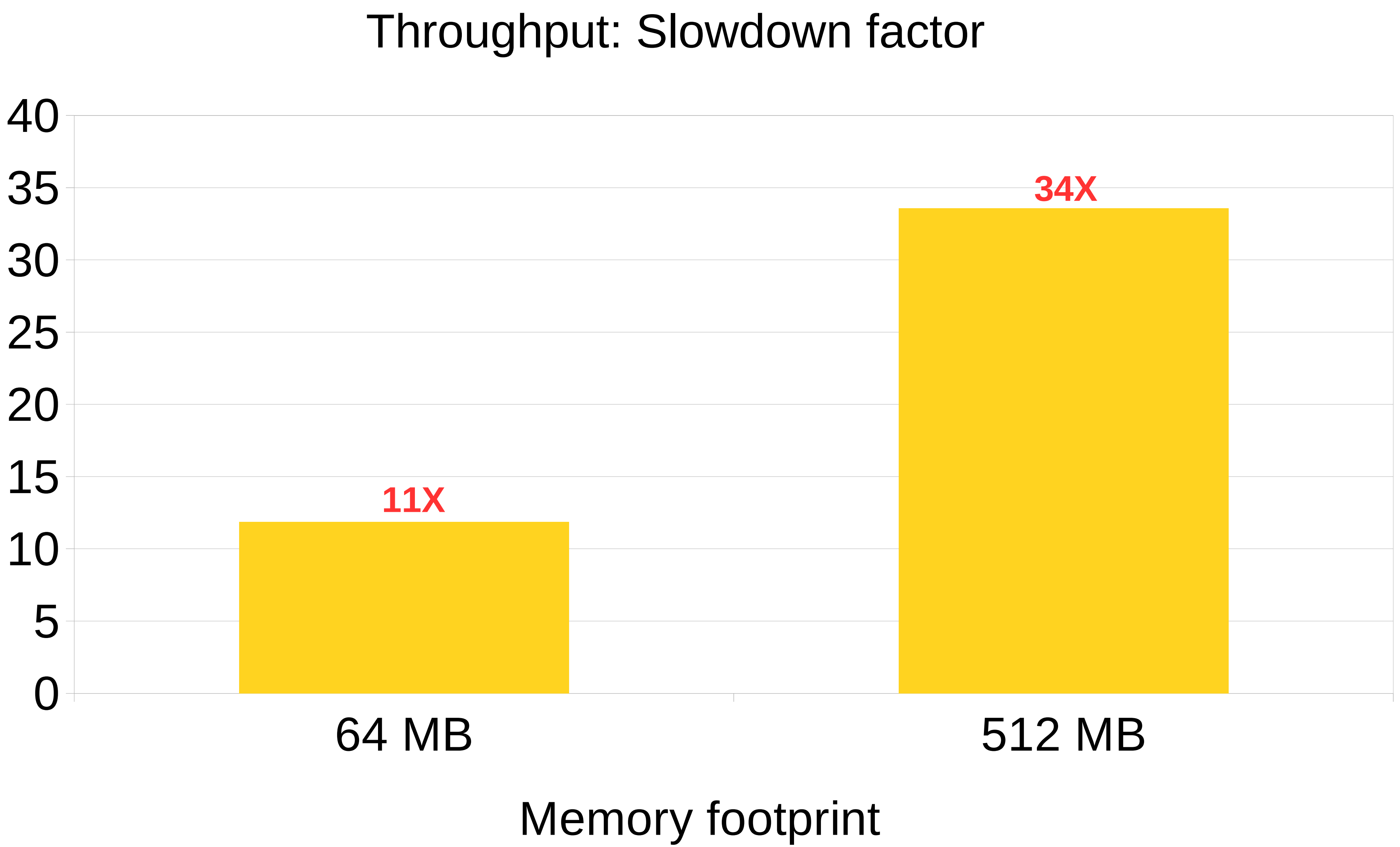
- ISA extensions
- Implemented in HW & Firmware
- Same CPU HW
- In-cache execution suffers no overheads



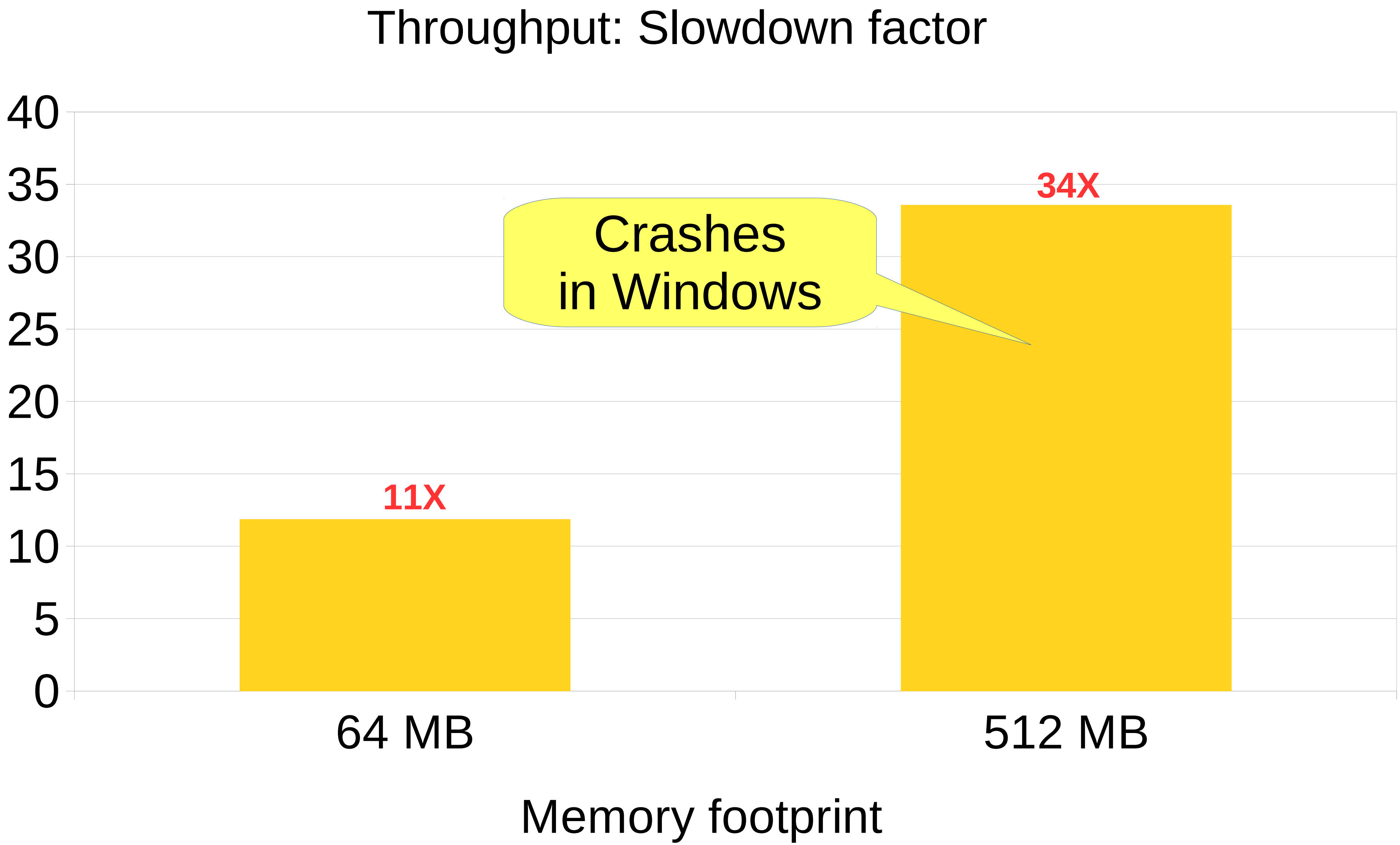
However...

Executing a Key-Value Store in enclave is slower

Executing a Key-Value Store in enclave is slower

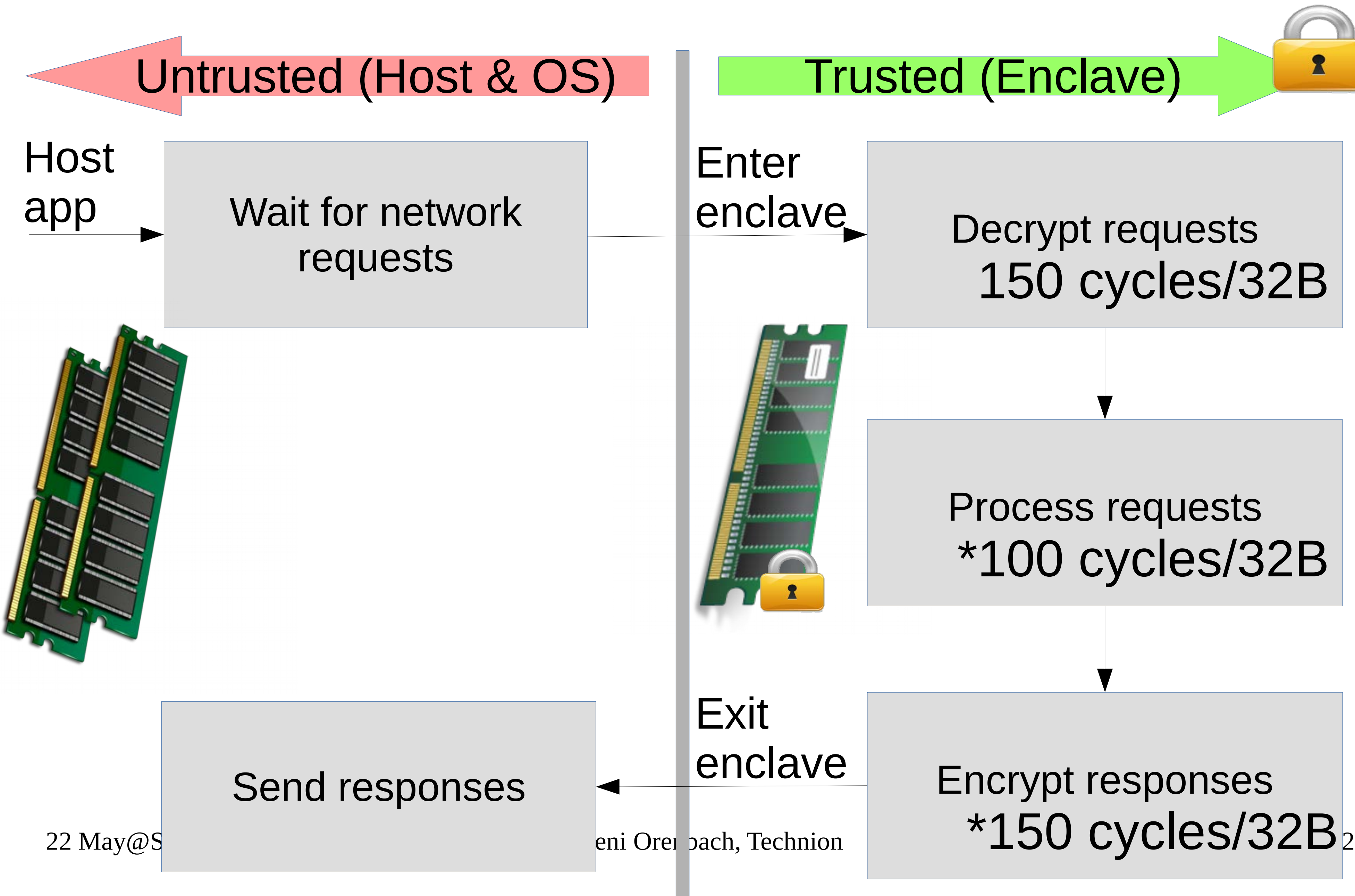


Executing a Key-Value Store in enclave is slower

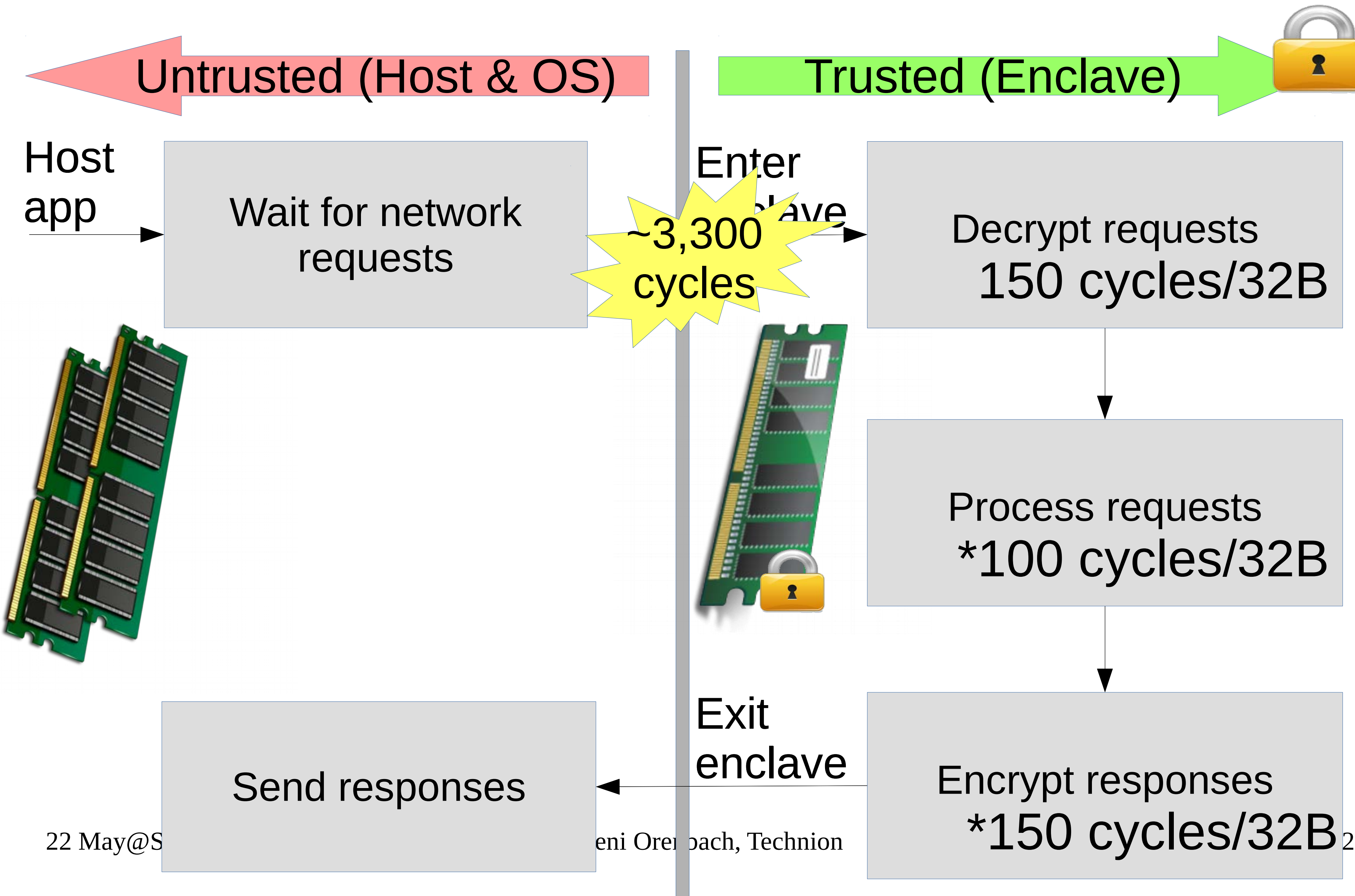


- Background
- Motivation
- **Overhead analysis**
- Eleos design
- Evaluation

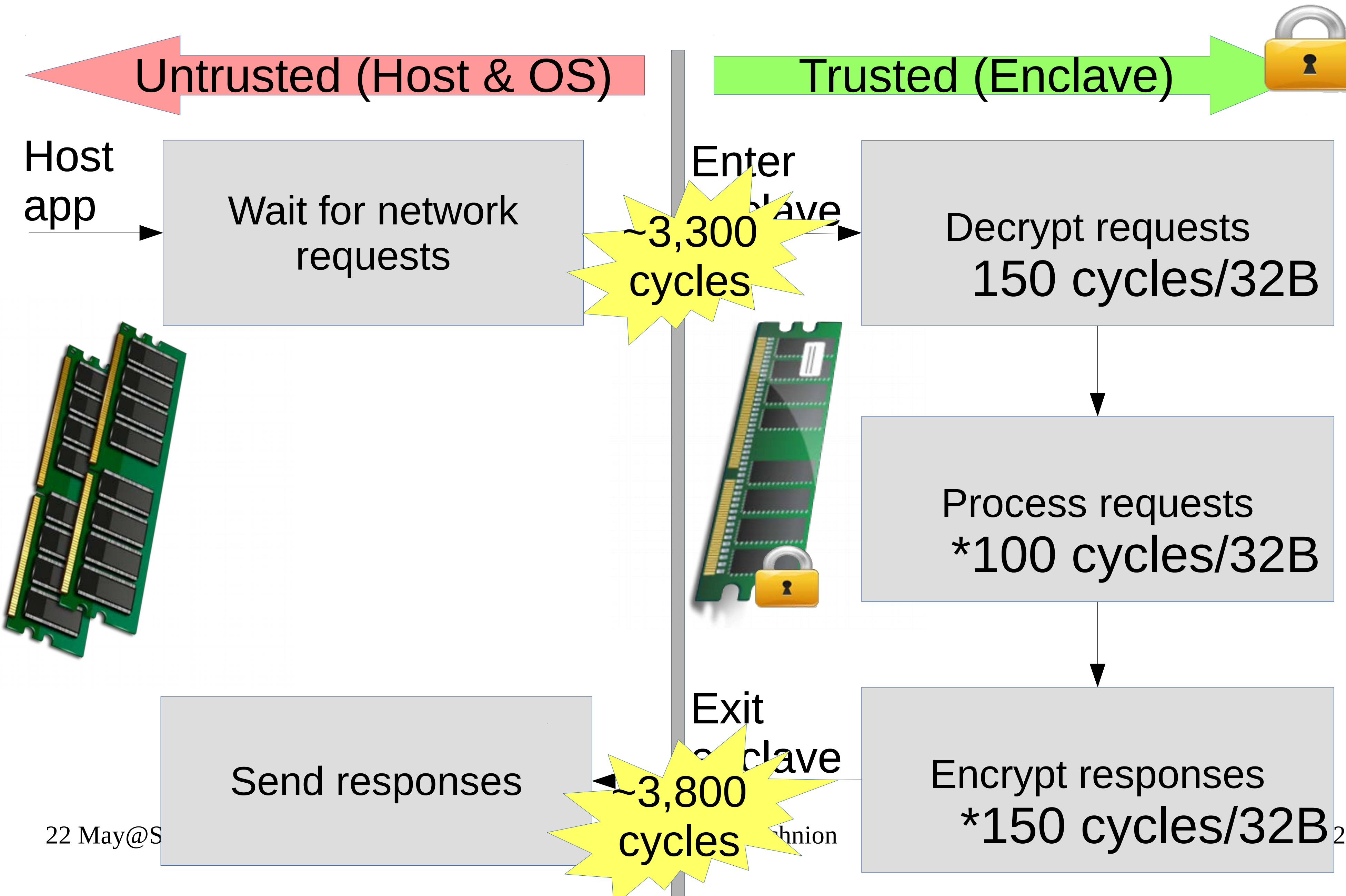
Overhead analysis



Overhead analysis



Overhead analysis



Overhead analysis

Untrusted (Host & OS) | Trusted (Enclave)

Host app → Wait for network requests

~3,300 cycles

Enter
clave

Decrypt requests
150 cycles/32B

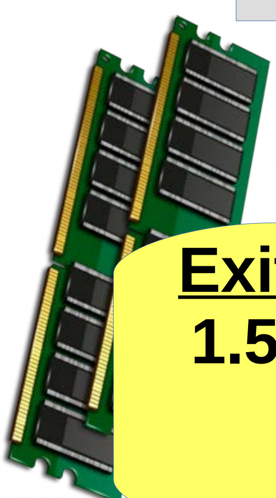
Process requests
*100 cycles/32B

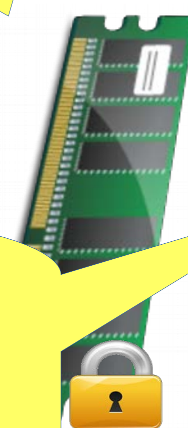
Encrypt responses
*150 cycles/32B

Exit
clave

~3,800 cycles

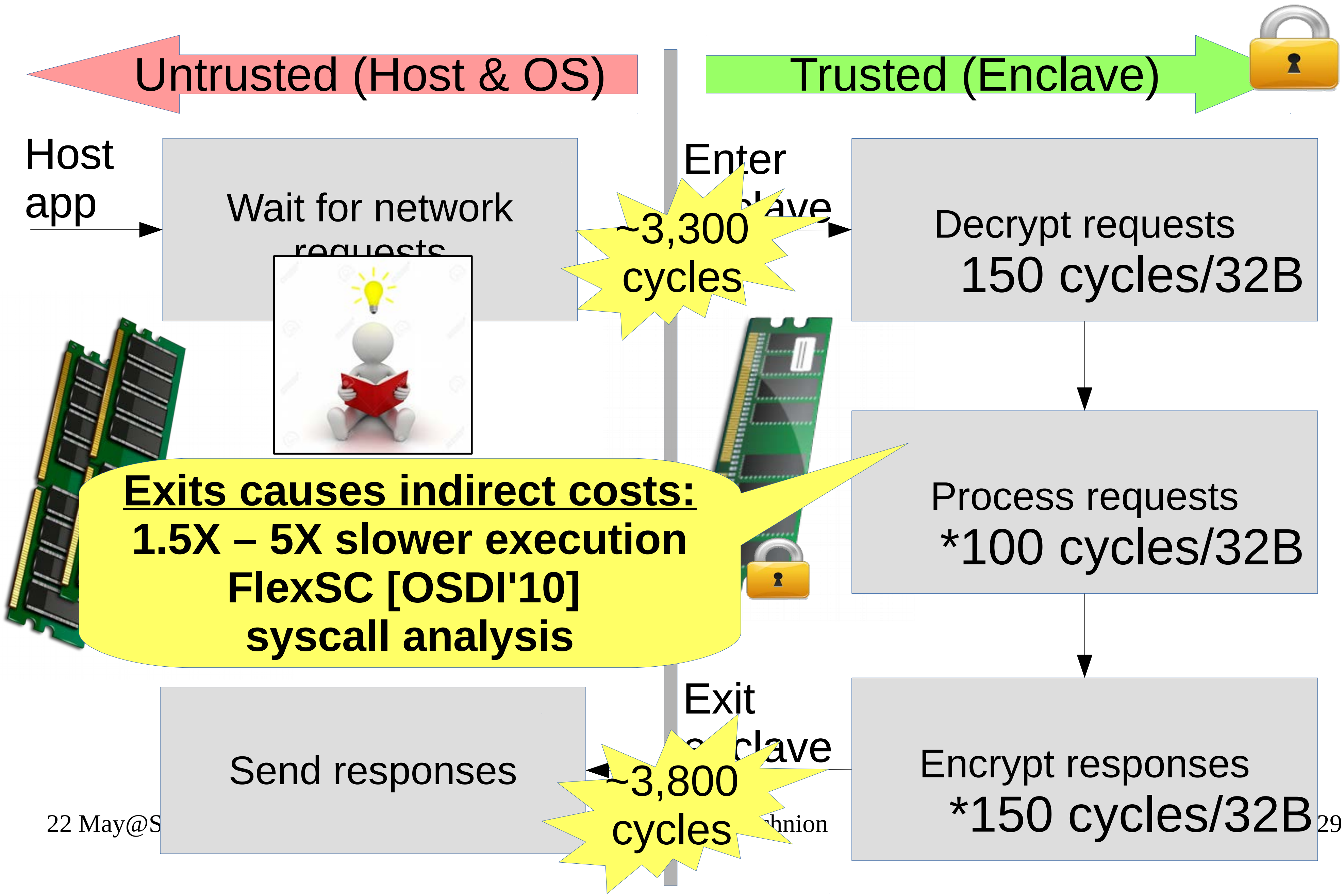
Send responses



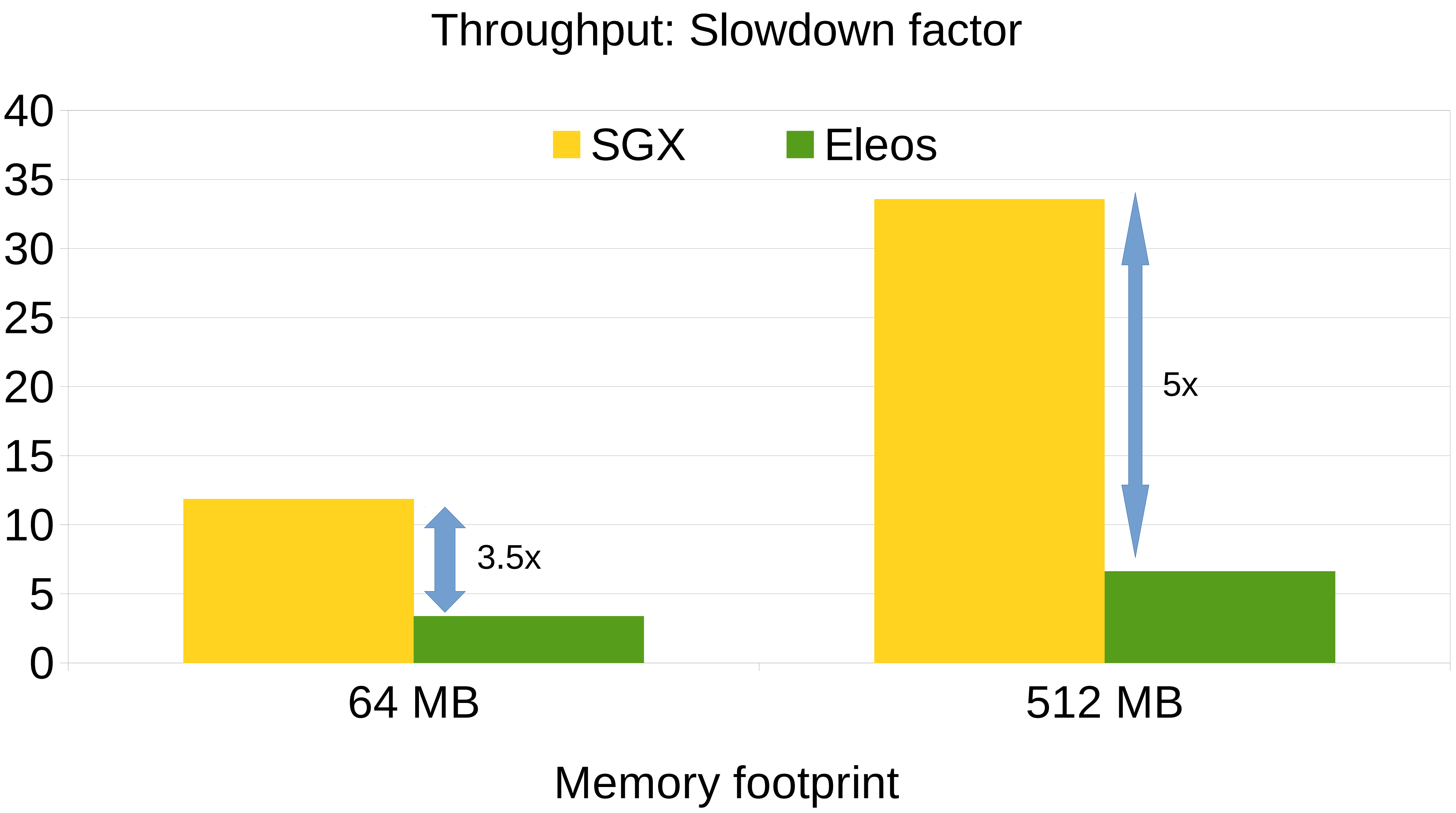


<u>Exits causes indirect costs:</u>
1.5X – 5X slower execution
FlexSC [OSDI'10]
syscall analysis

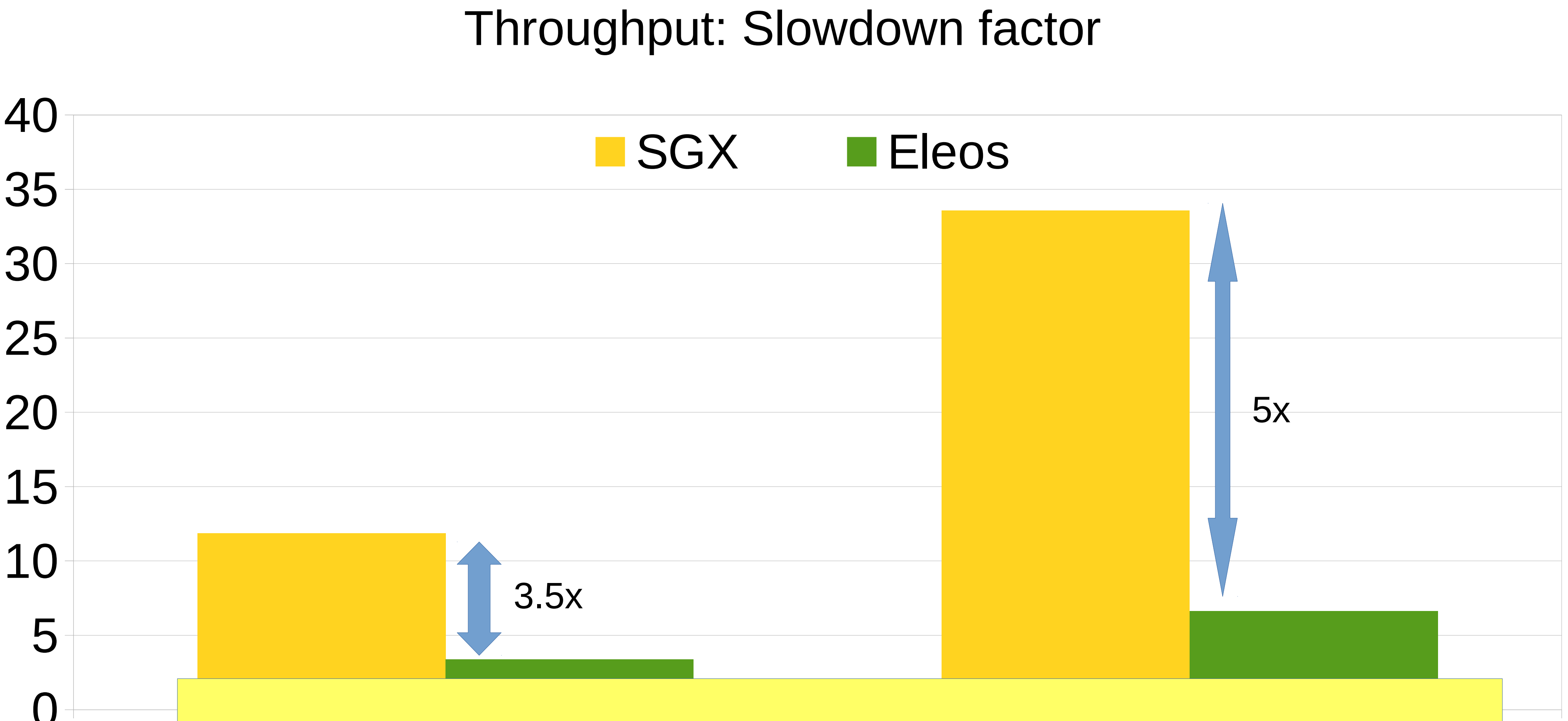
Overhead analysis



Eleos does better!



Eleos does better!



How does Eleos achieve this?

Eleos: **Exit-less** services

Exit-less system calls with RPC infrastructure

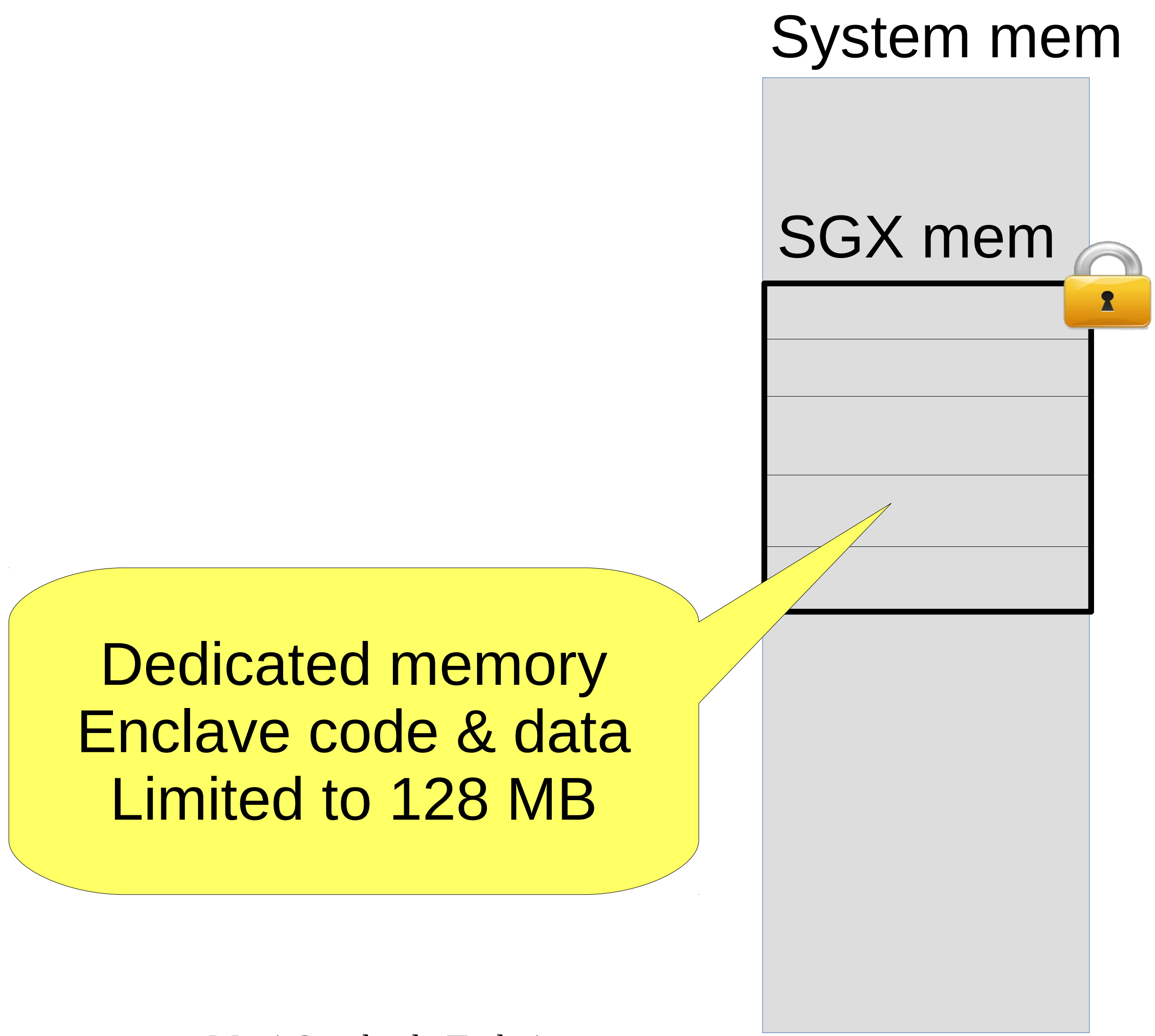
Exit-less SGX paging

Eleos: **Exit-less** services

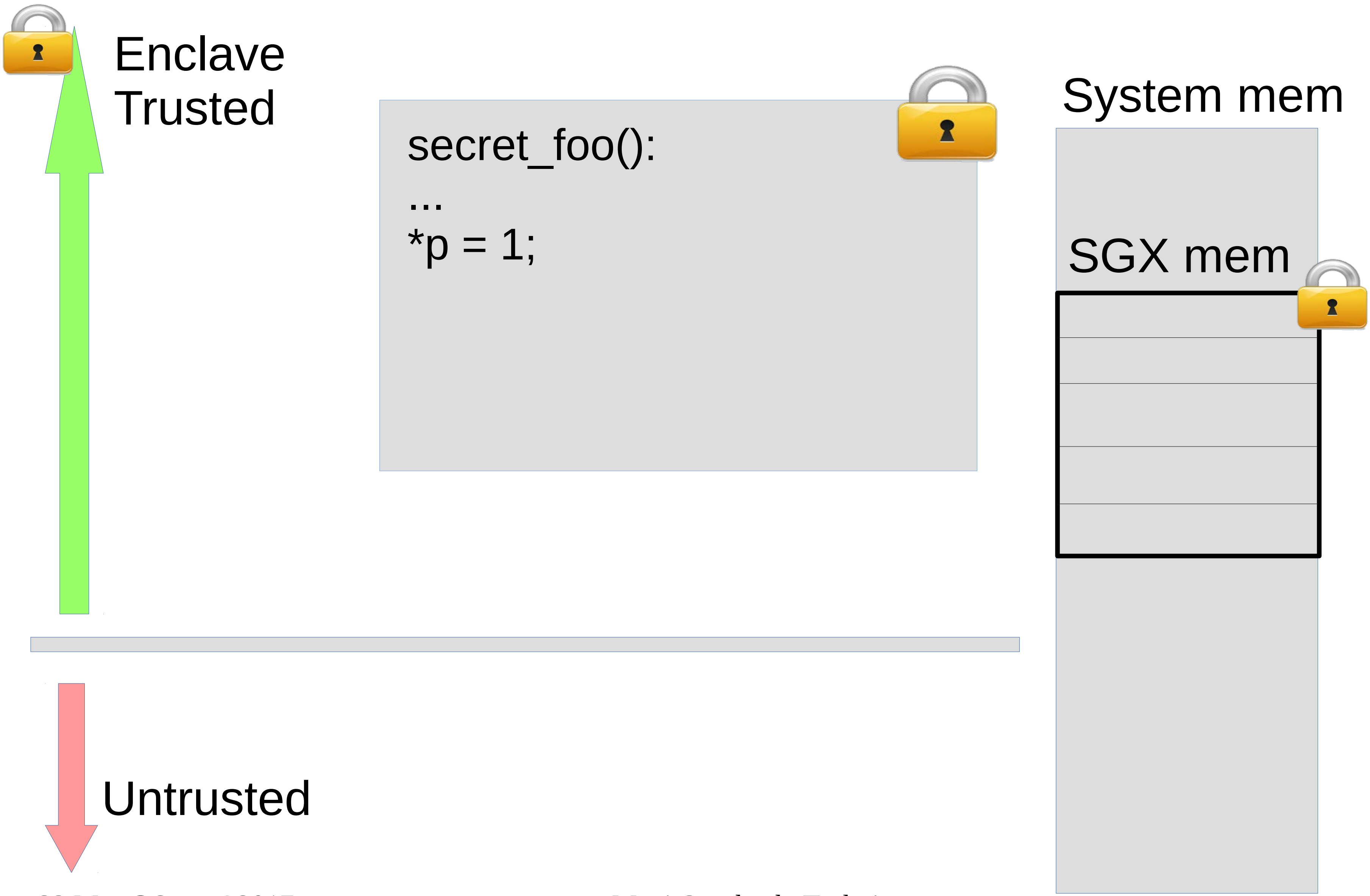
Exit-less system calls with RPC infrastructure

Exit-less SGX paging

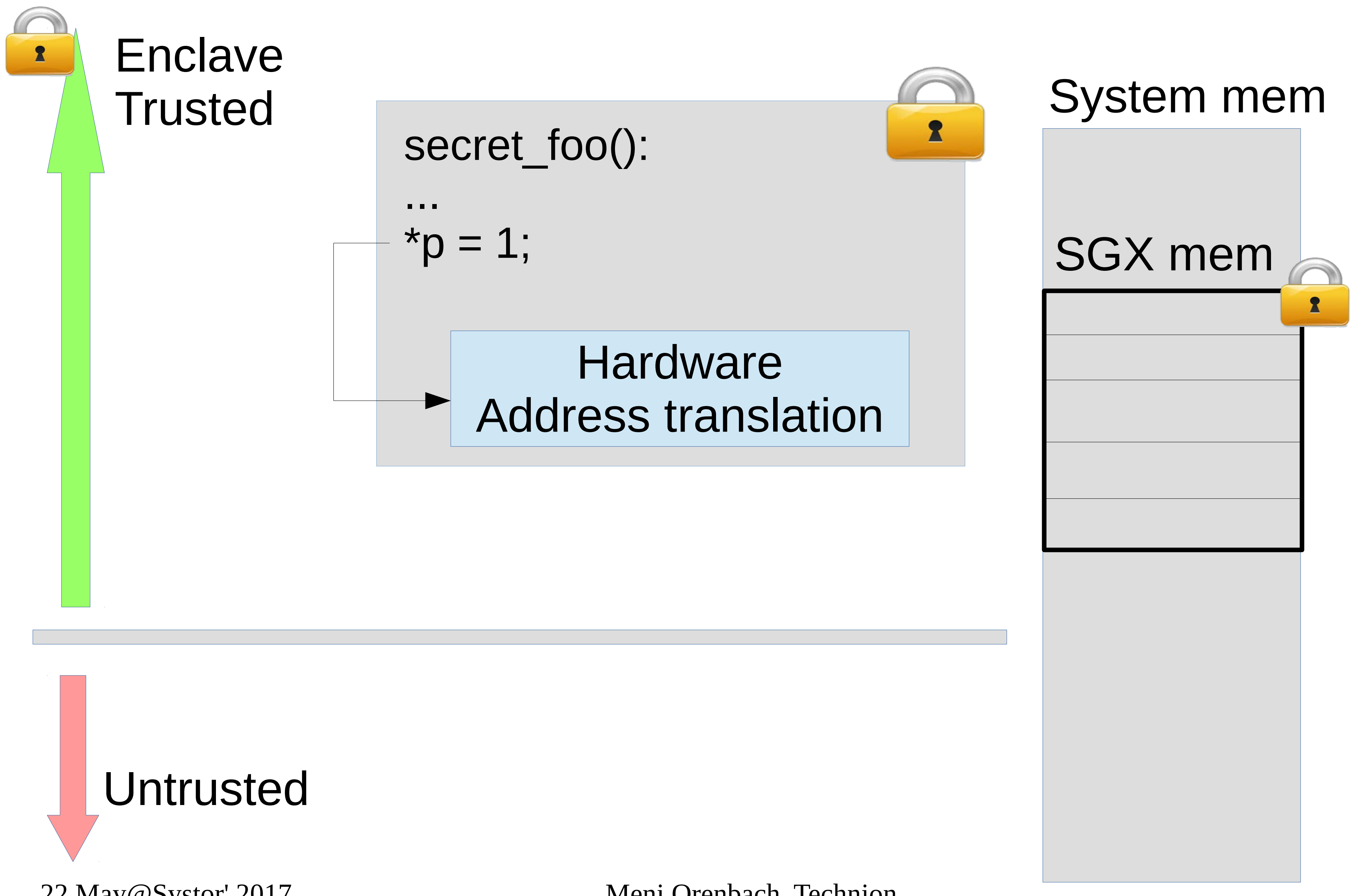
Background: SGX paging



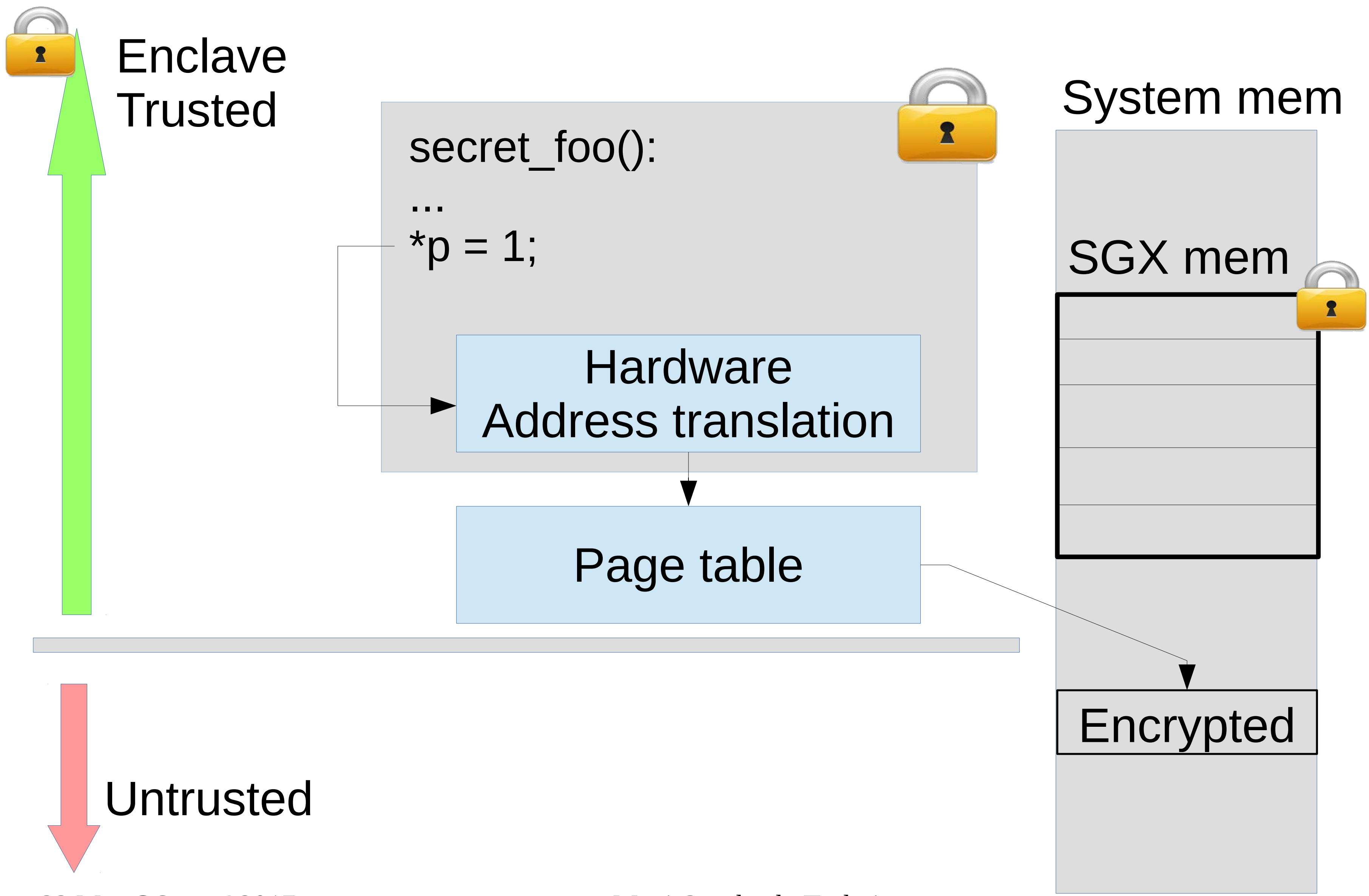
Background: SGX paging



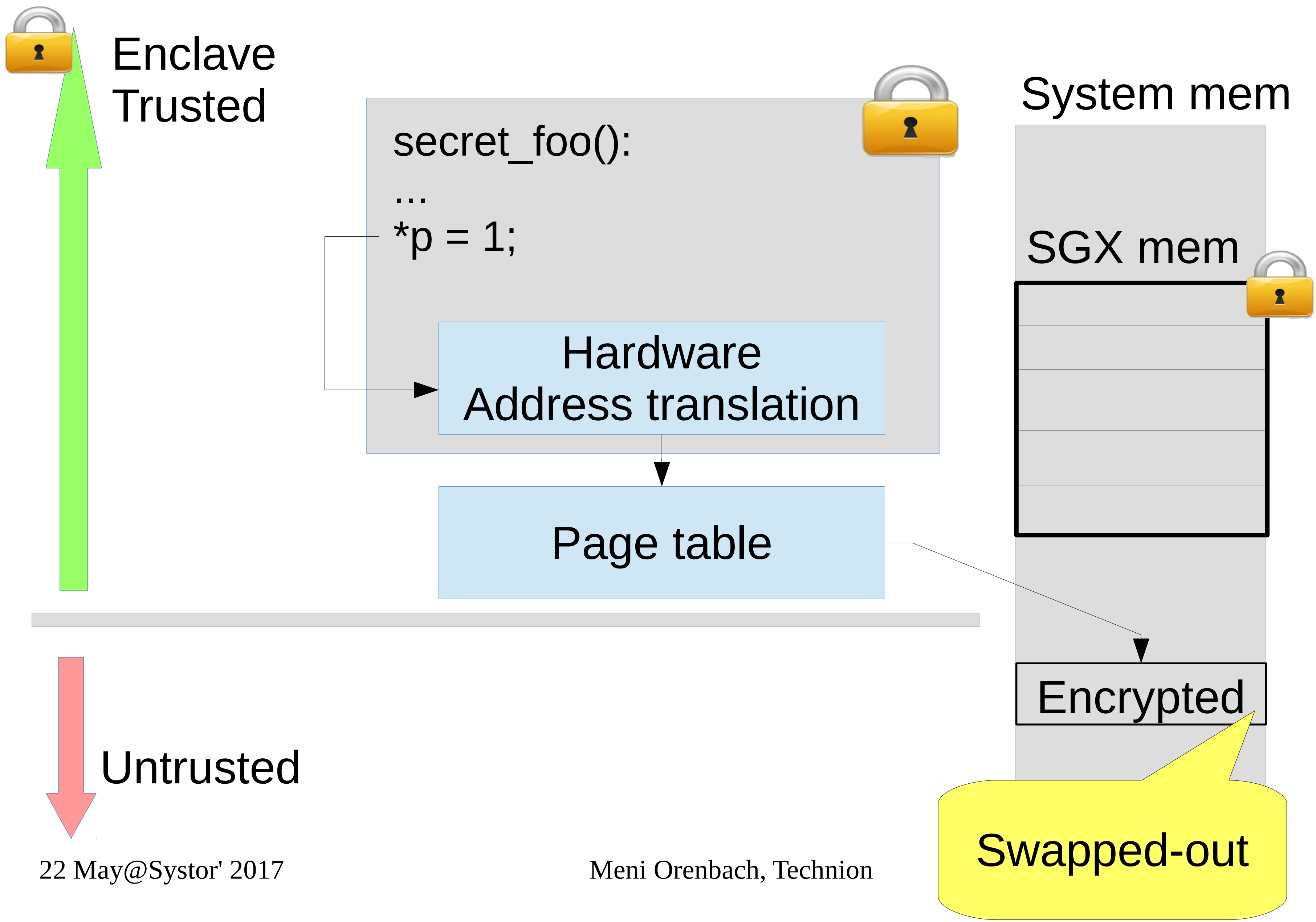
Background: SGX paging



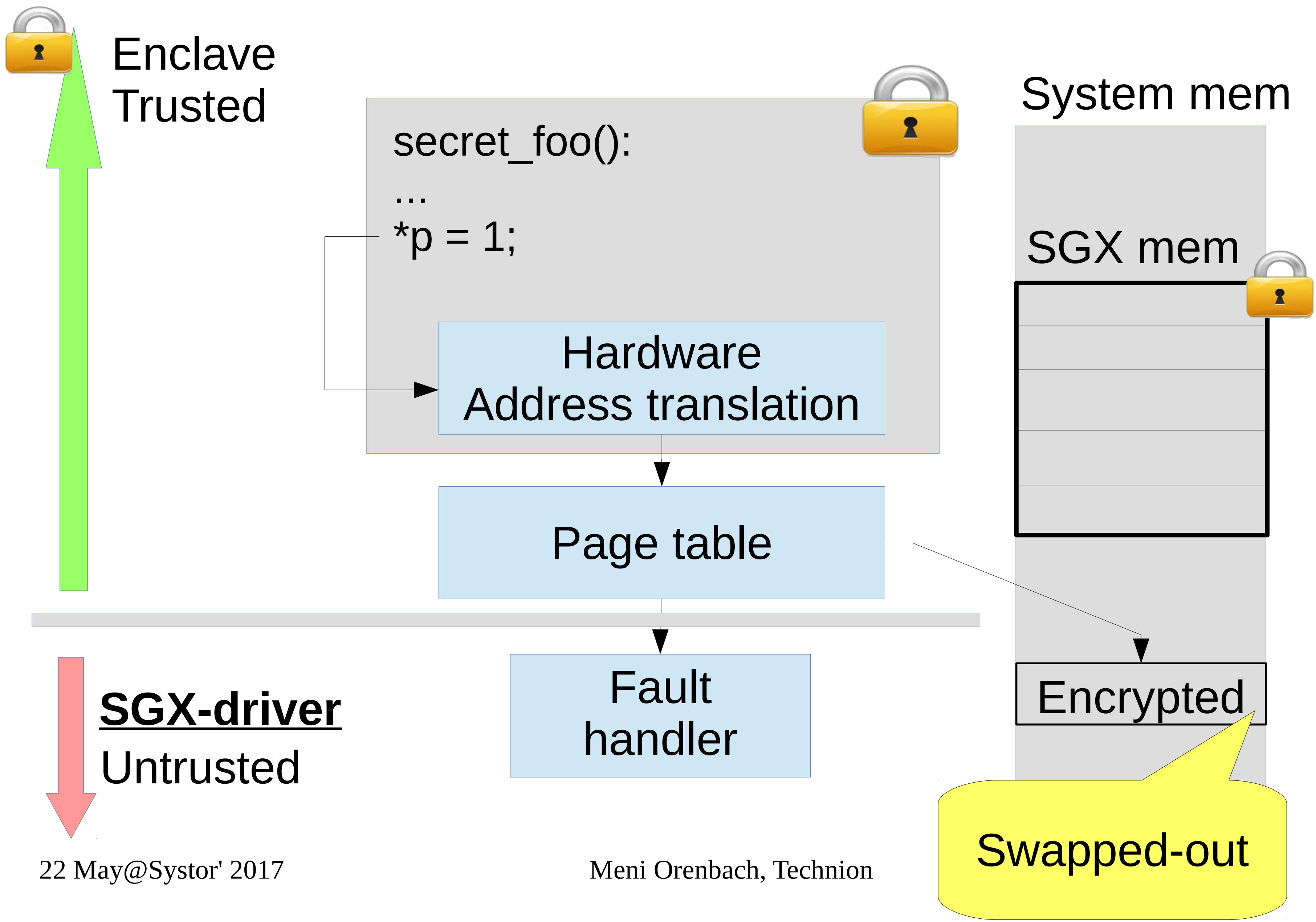
Background: SGX paging



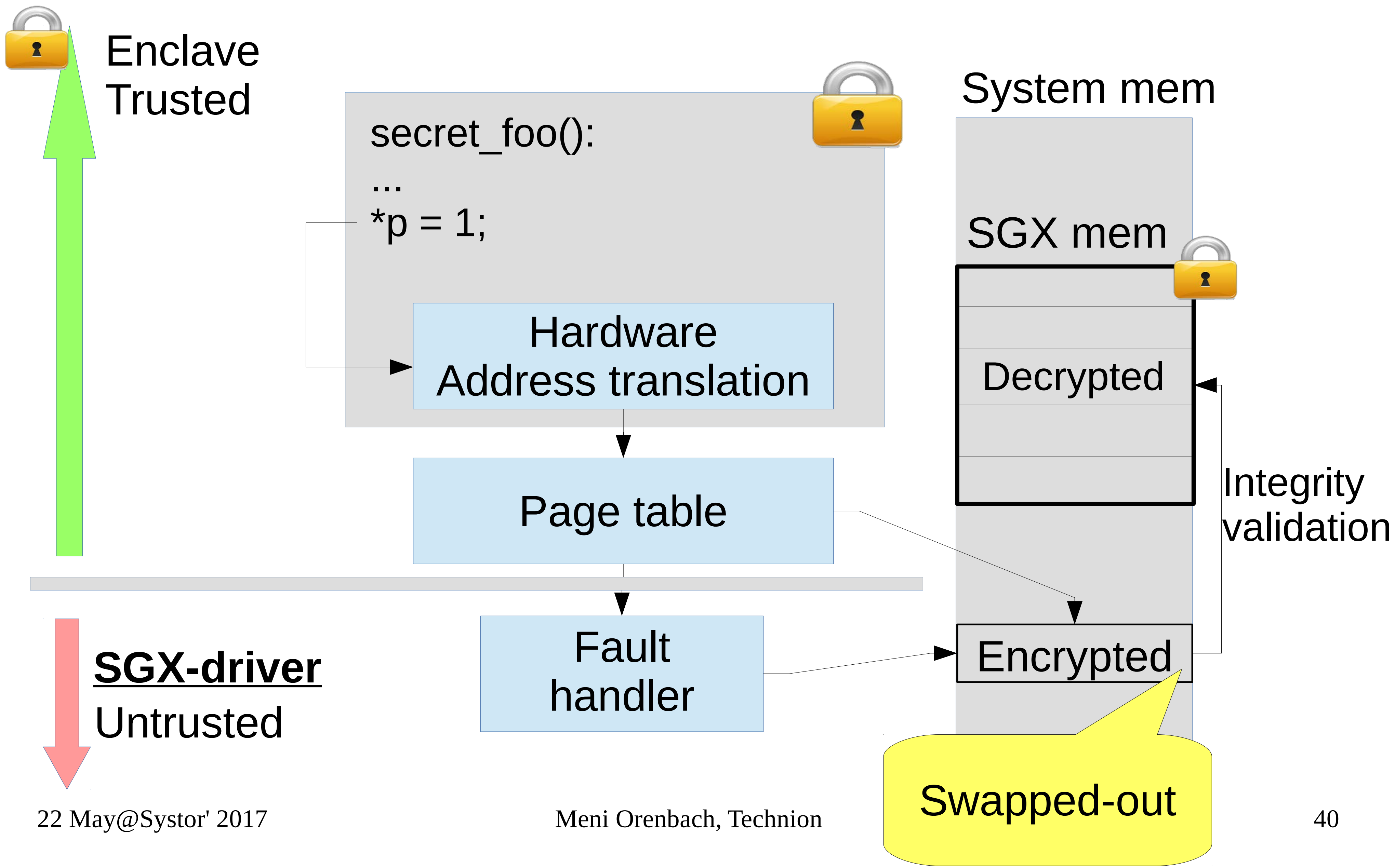
Background: SGX paging



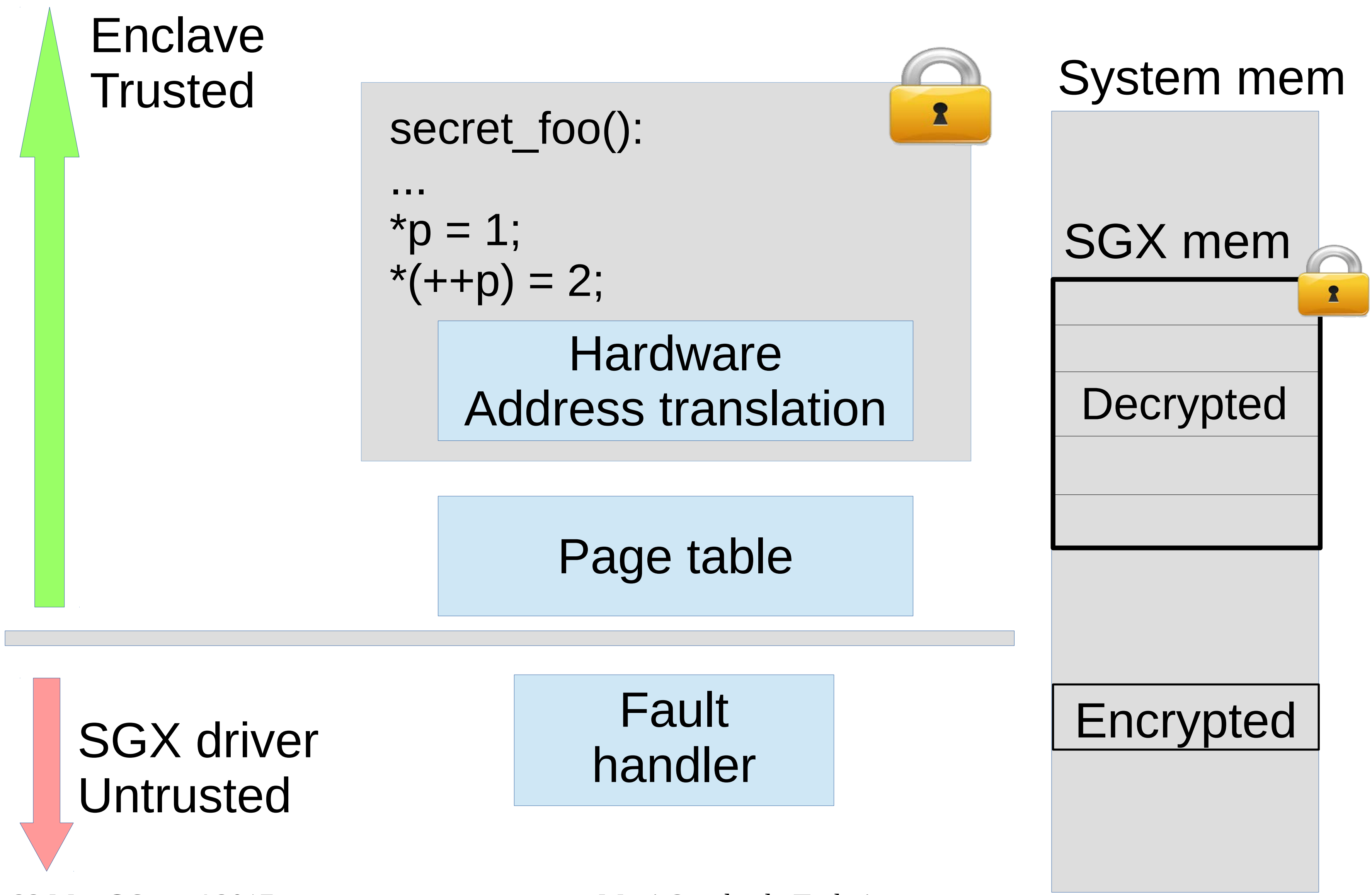
Background: SGX paging



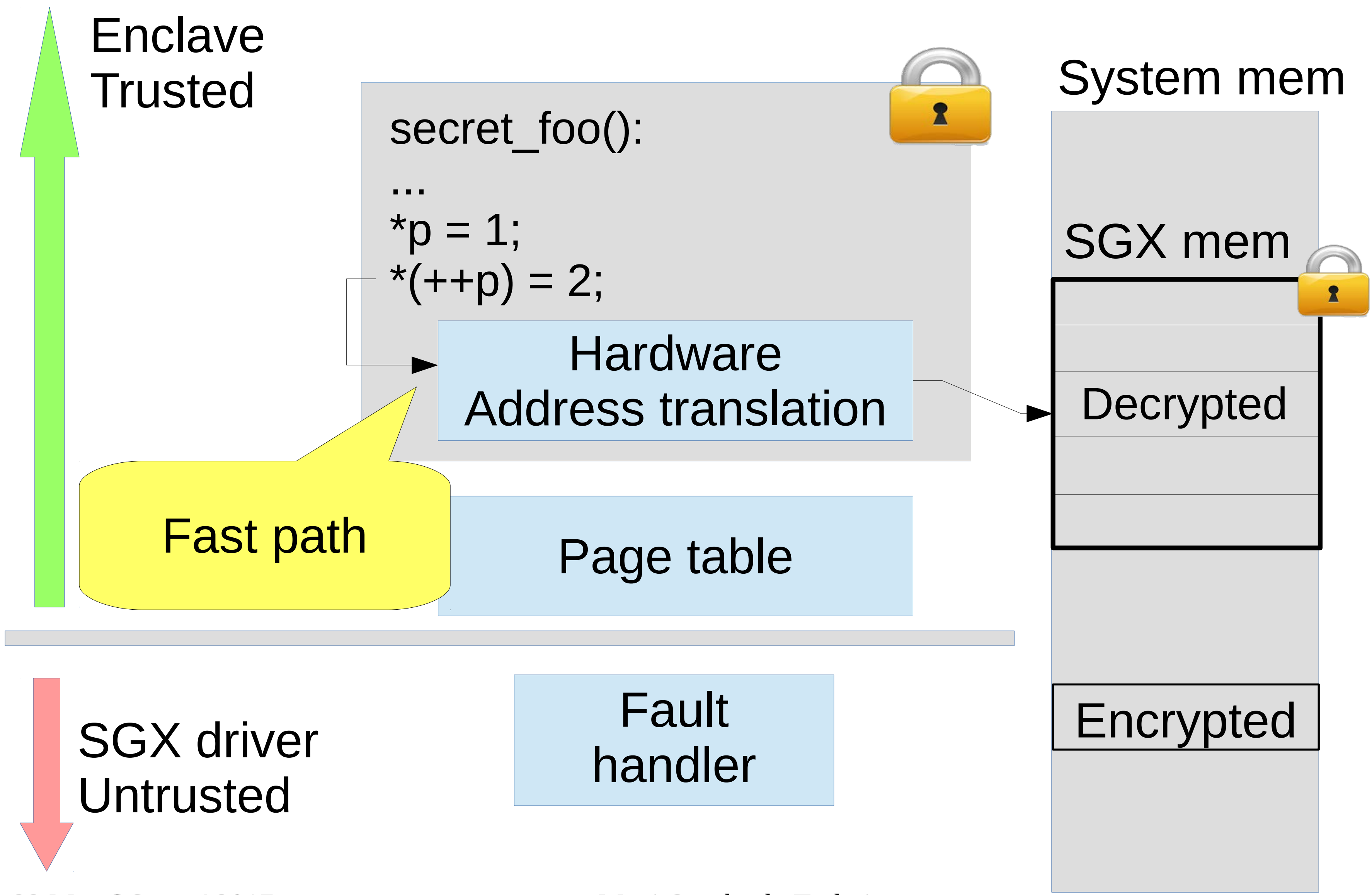
Background: SGX paging



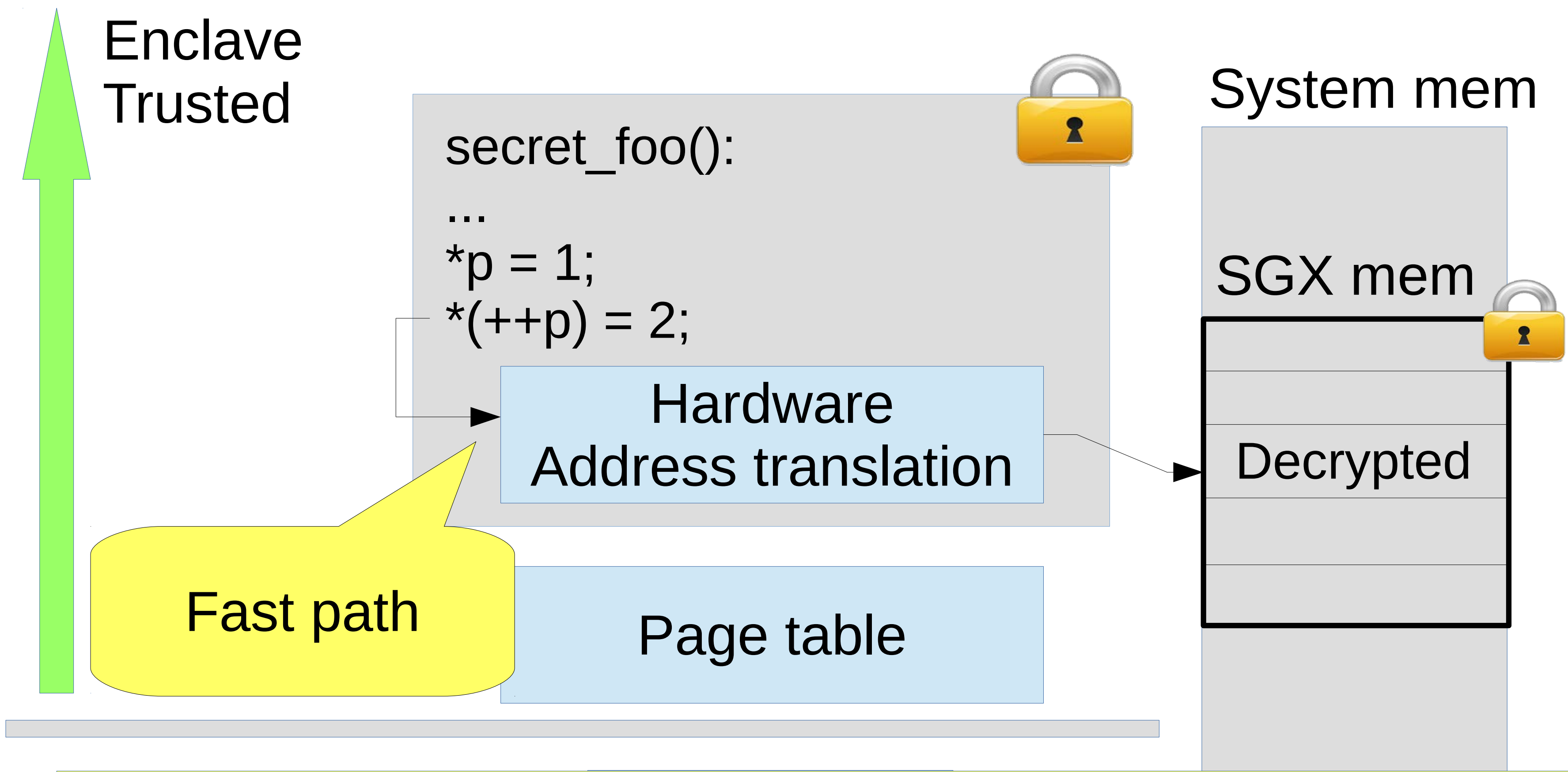
Background: SGX paging



Background: SGX paging

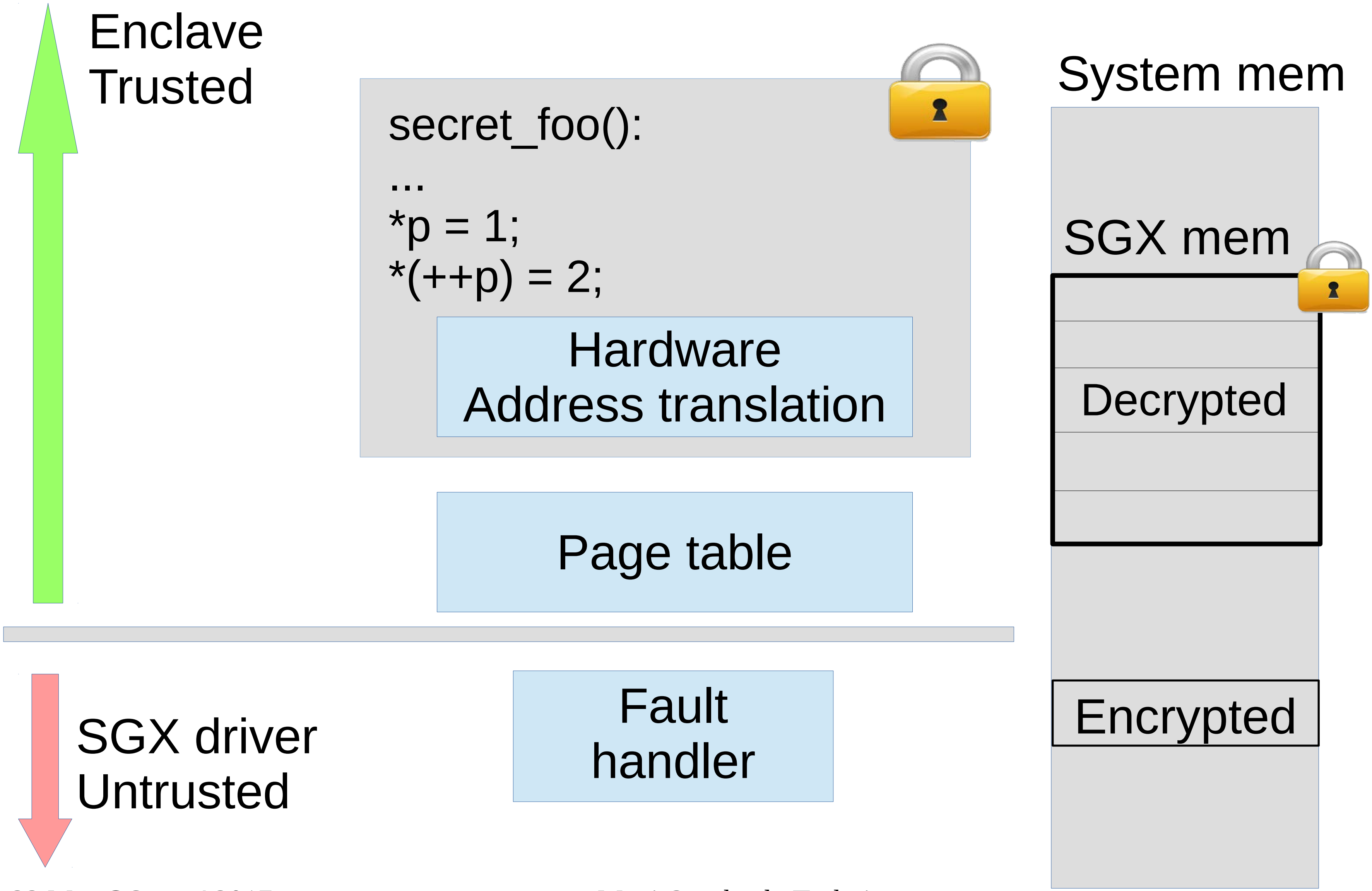


Background: SGX paging

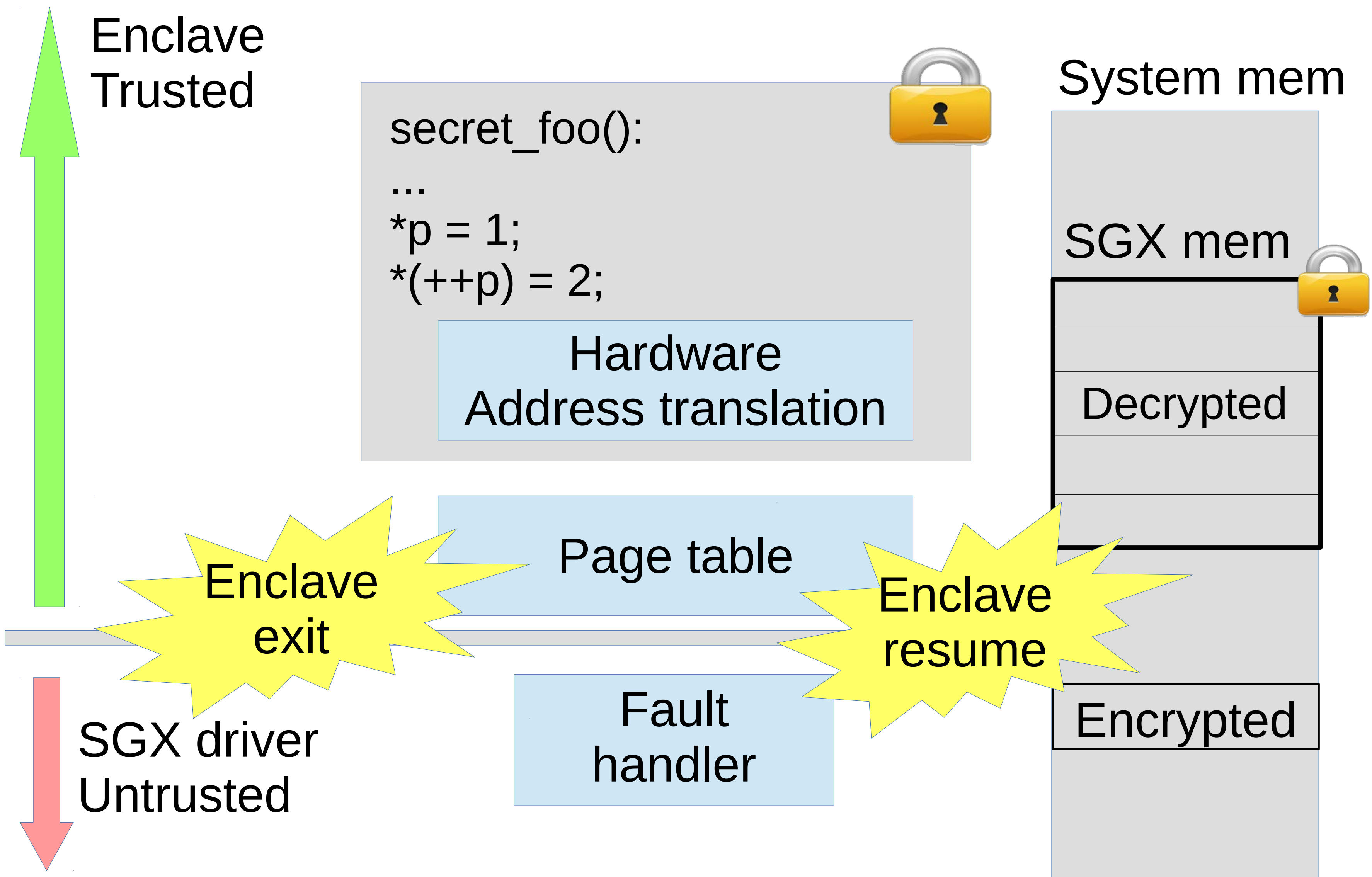


Since SGX memory is small
paging is not as rare as in native applications
What are the overheads?

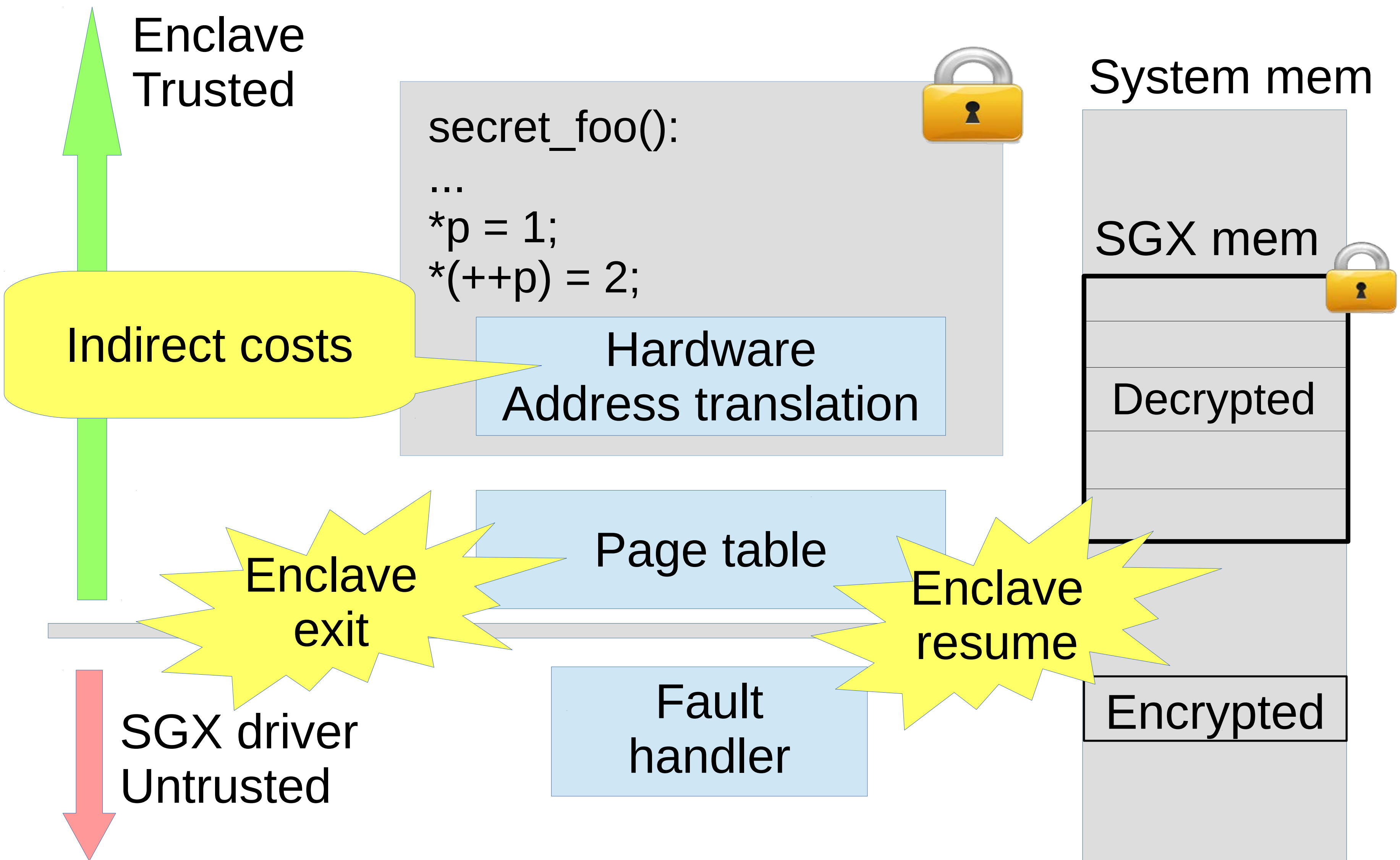
Background: SGX paging



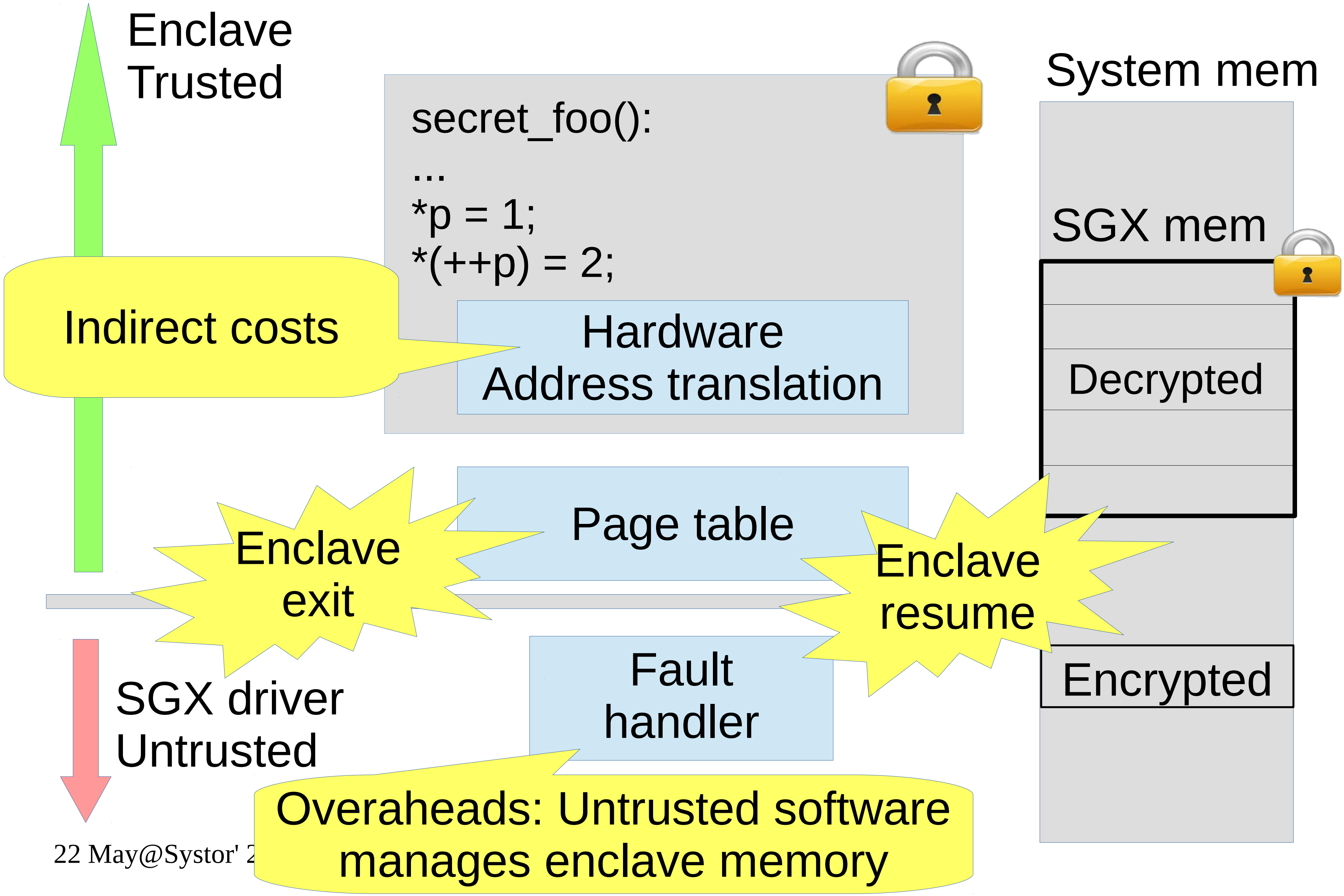
SGX paging overheads



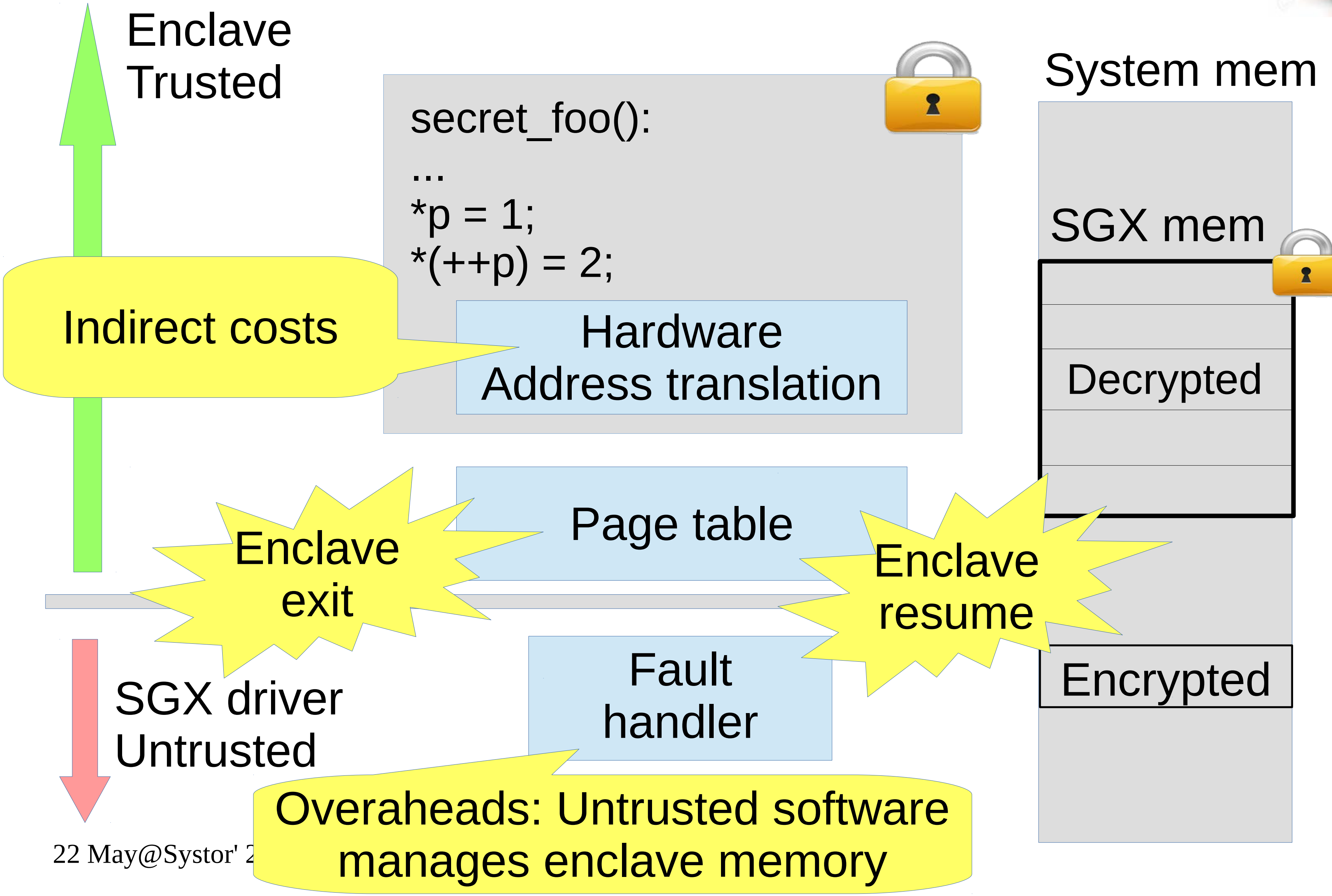
SGX paging overheads



SGX paging overheads



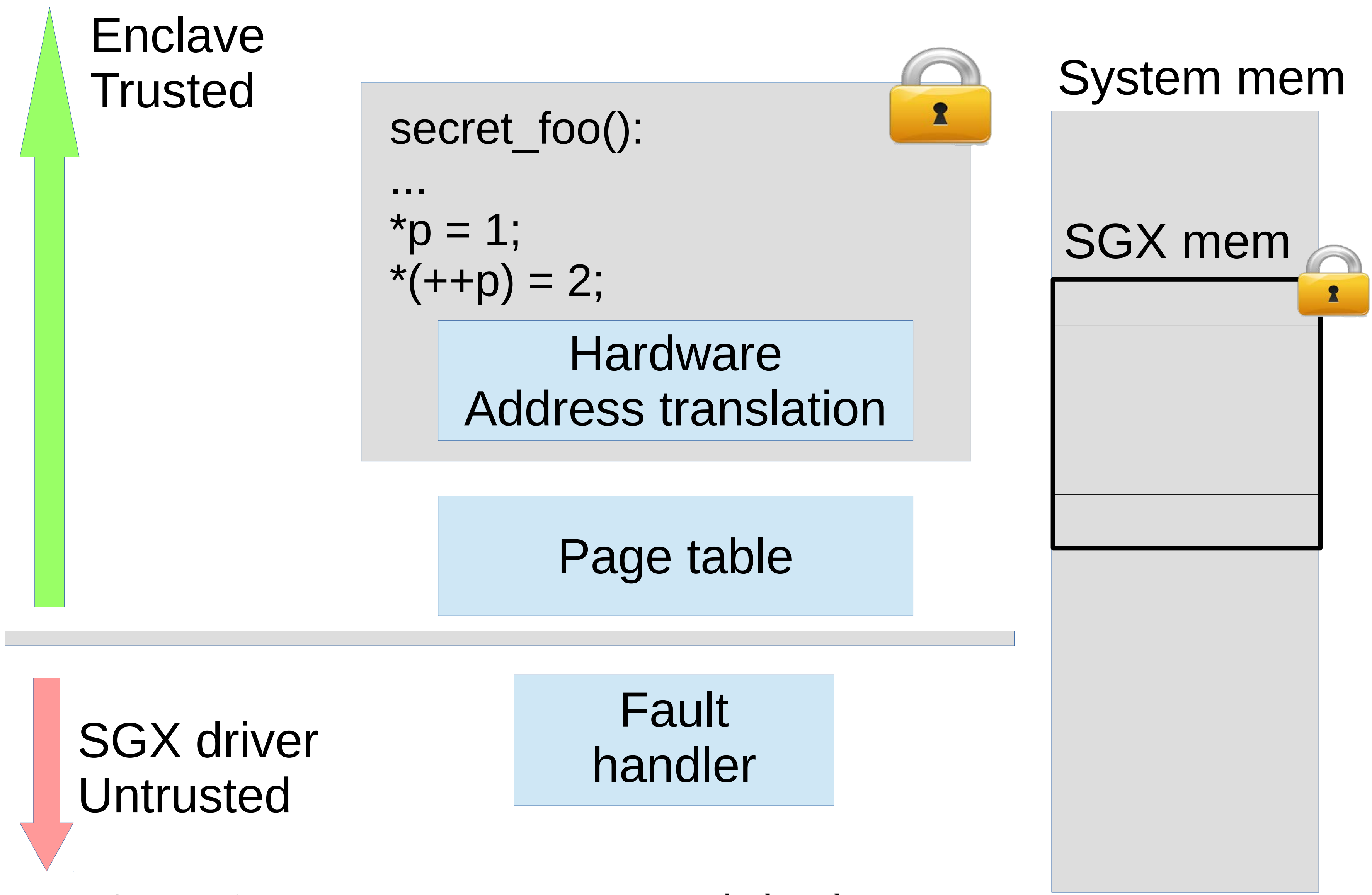
SGX paging overheads



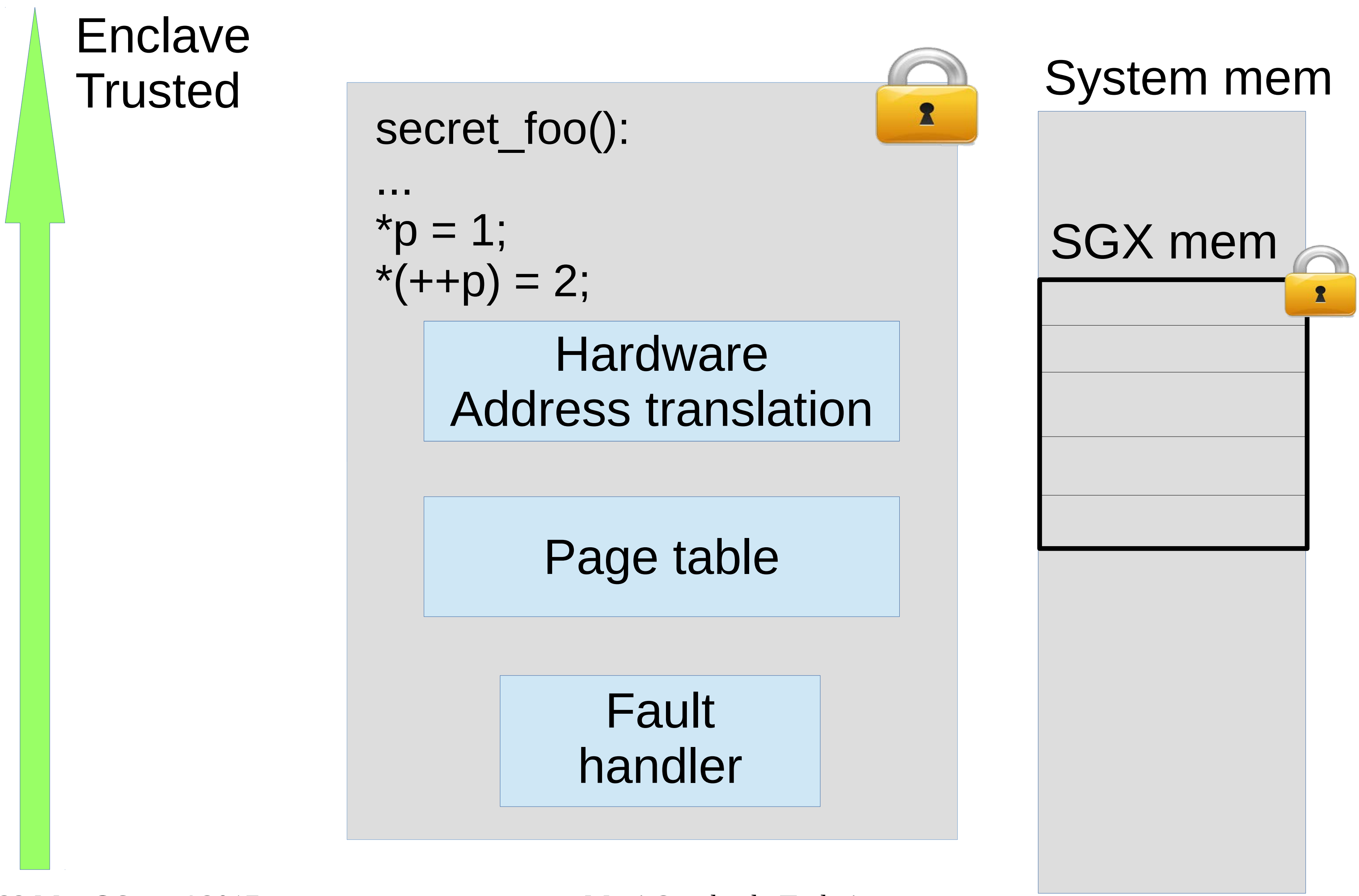
Wanted: In-enclave virtual memory management

No more exits!

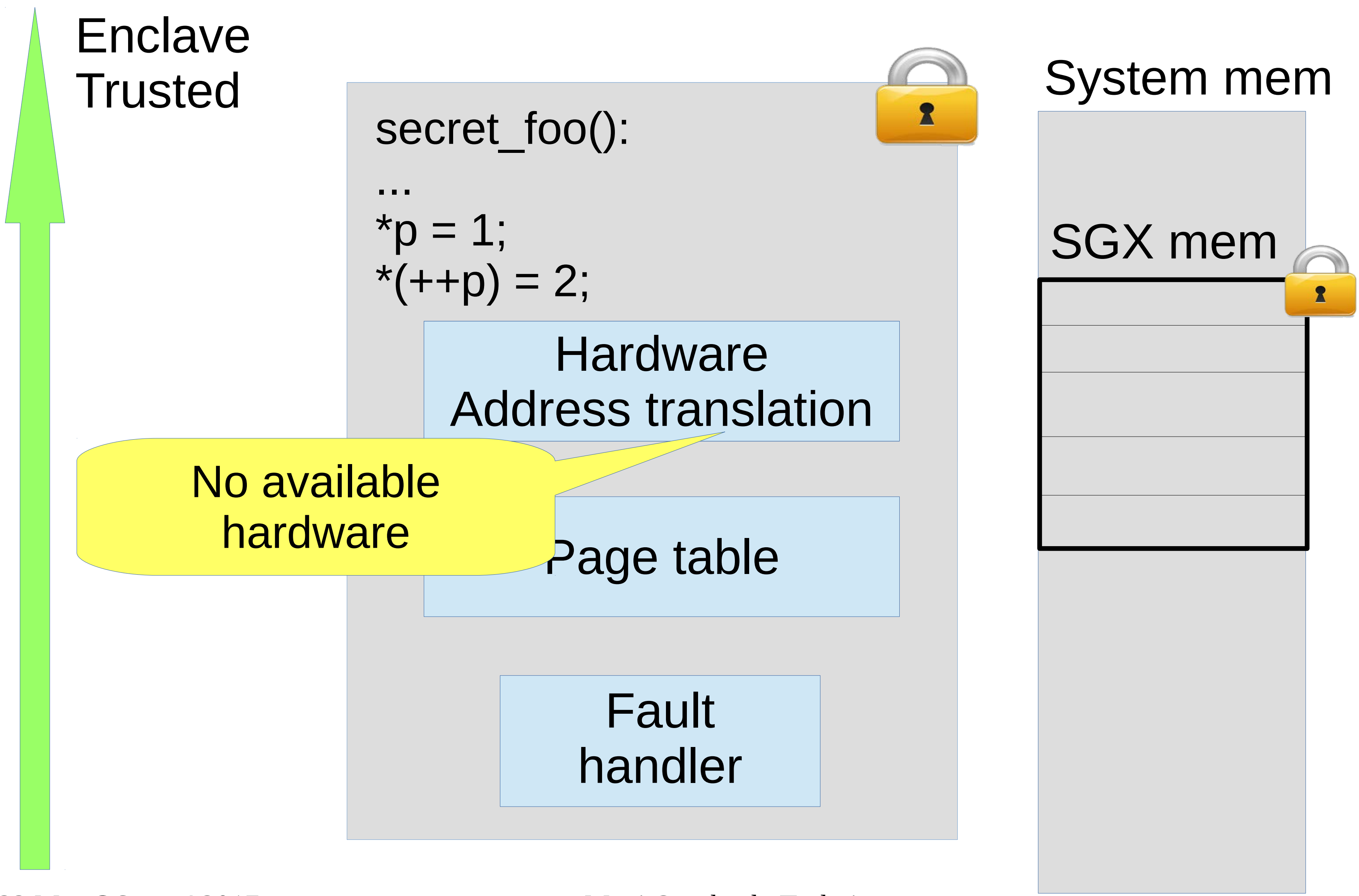
Ideal in-enclave VM management



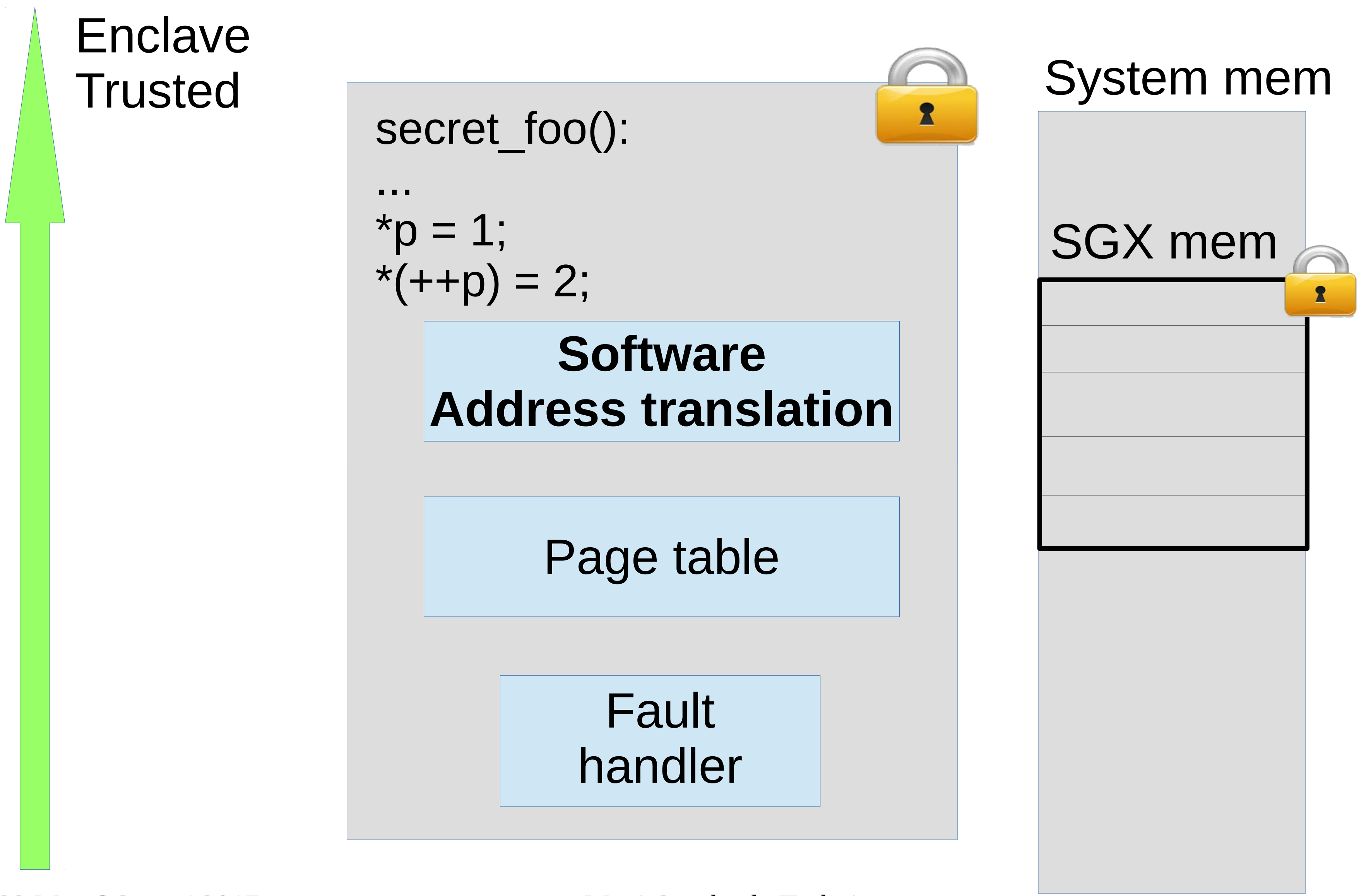
Ideal in-enclave VM management



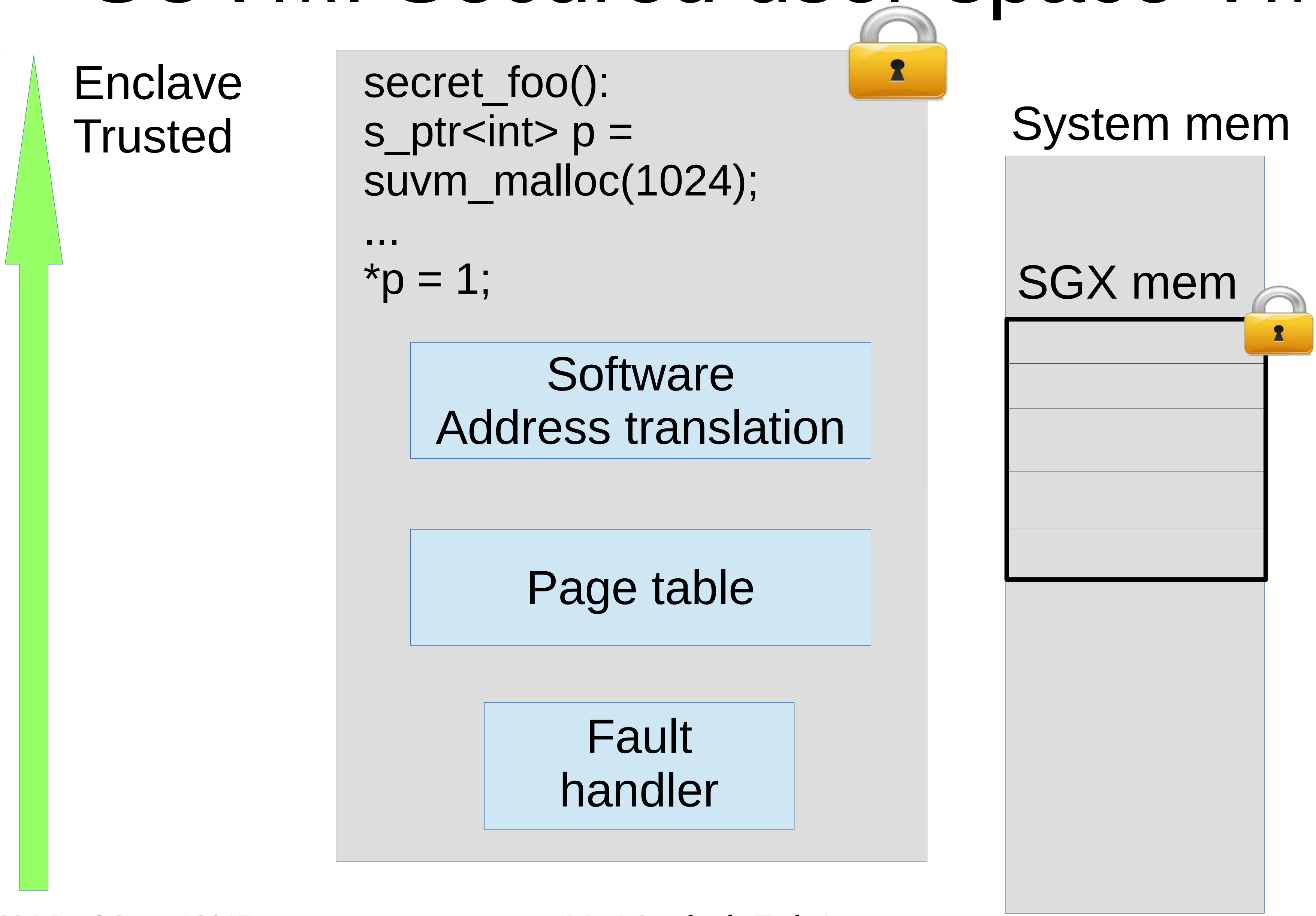
Ideal in-enclave VM management



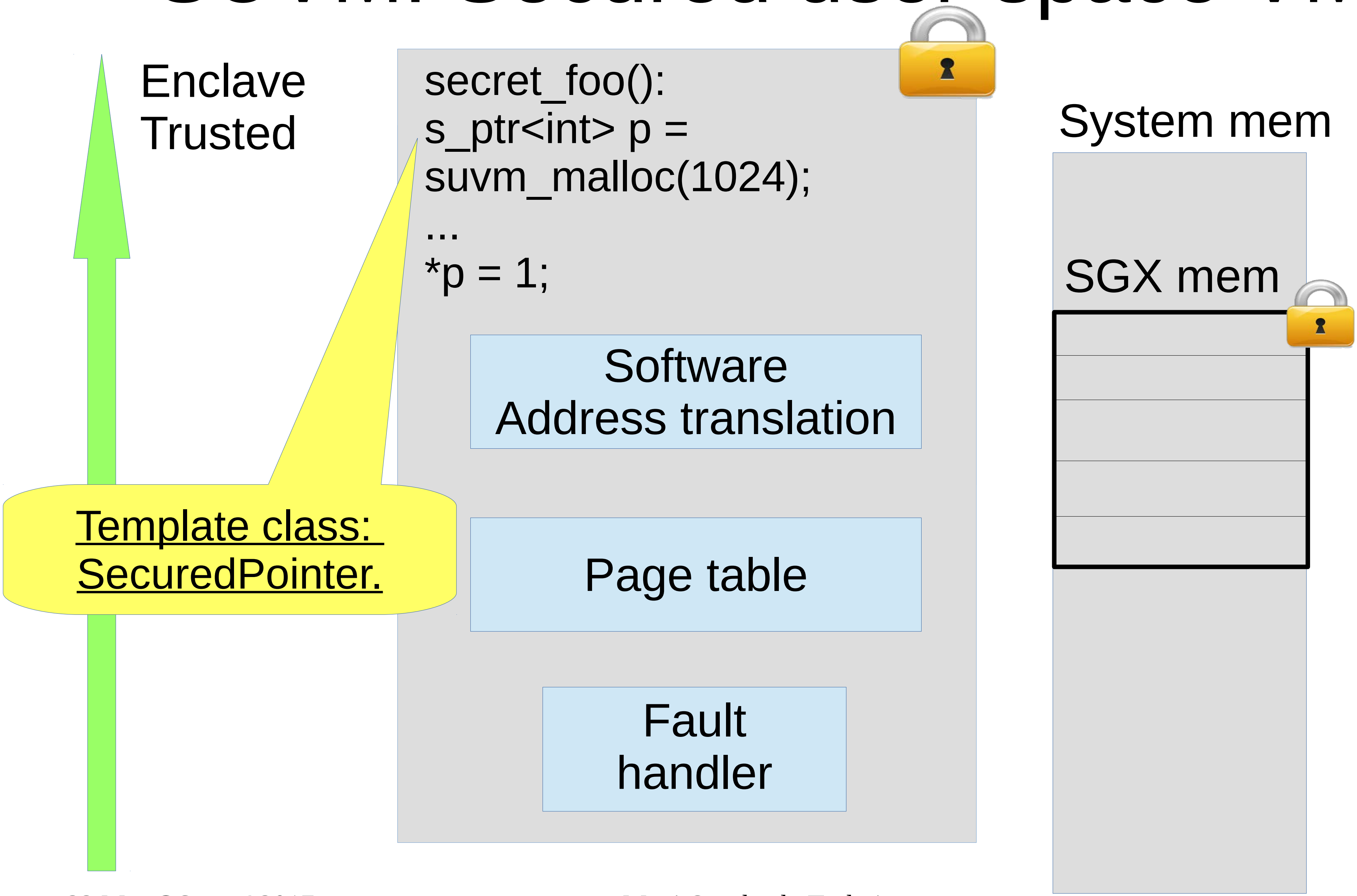
Ideal in-enclave VM management



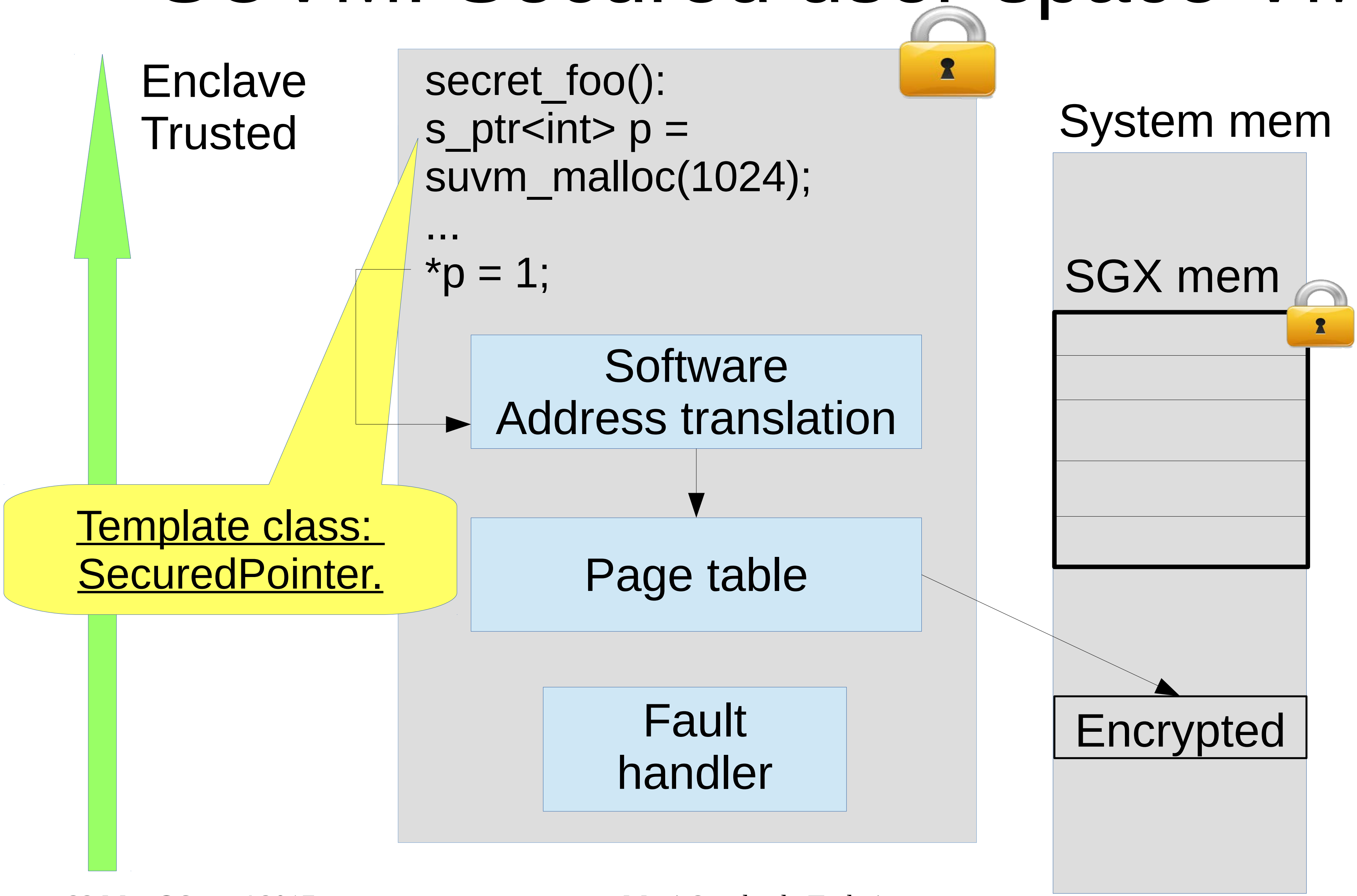
SUVM: Secured user-space VM



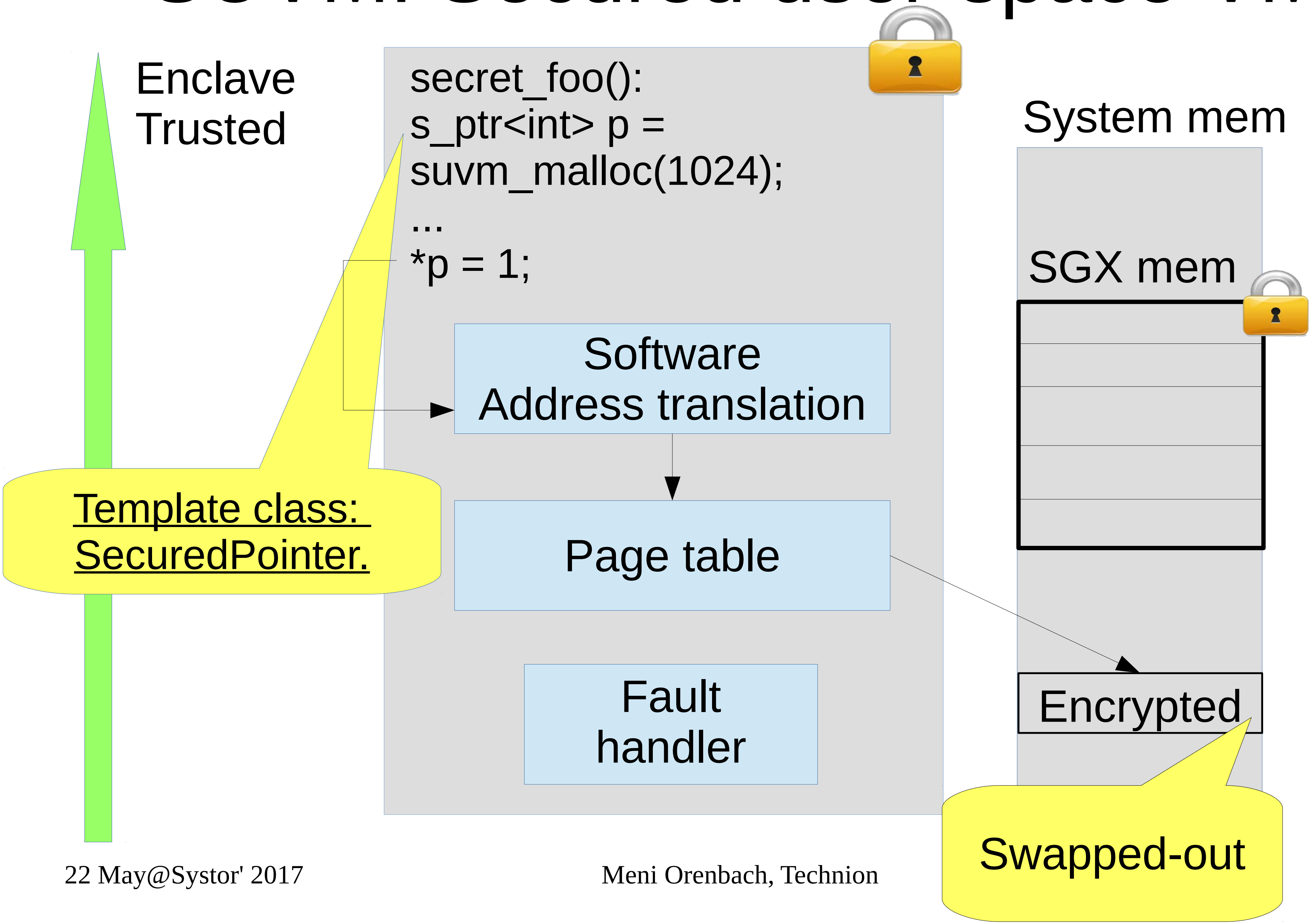
SUVM: Secured user-space VM



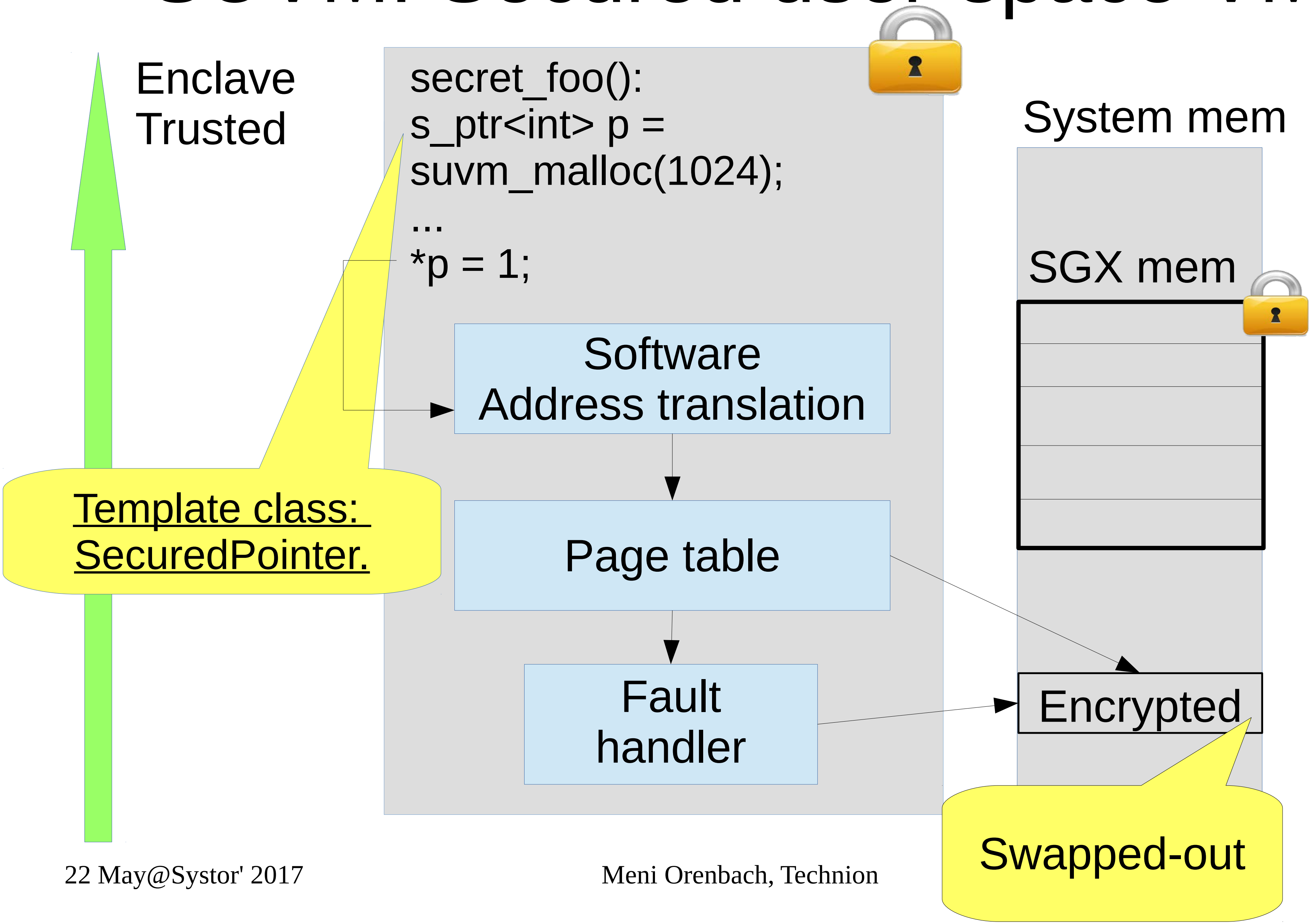
SUVM: Secured user-space VM



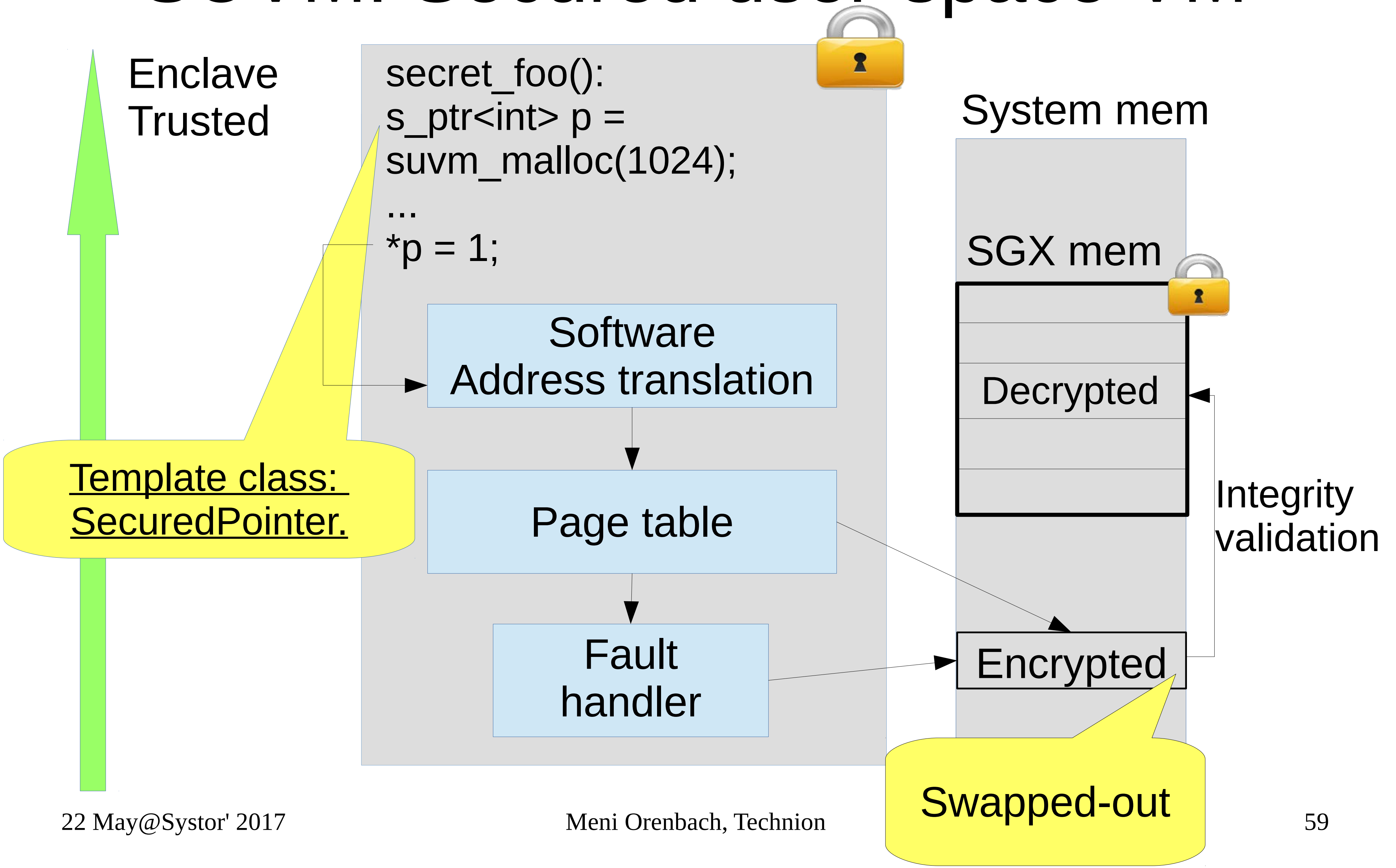
SUVM: Secured user-space VM



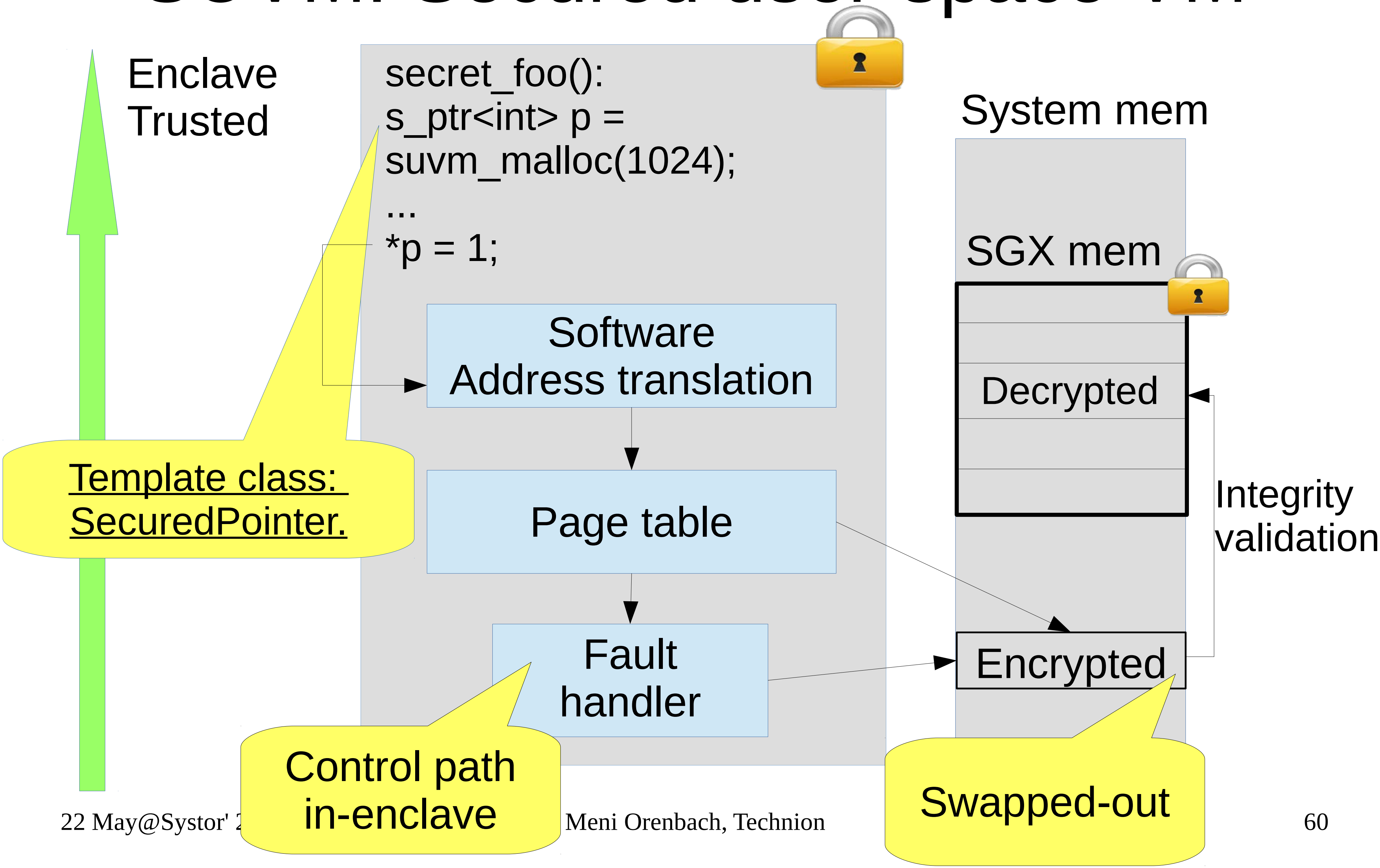
SUVM: Secured user-space VM



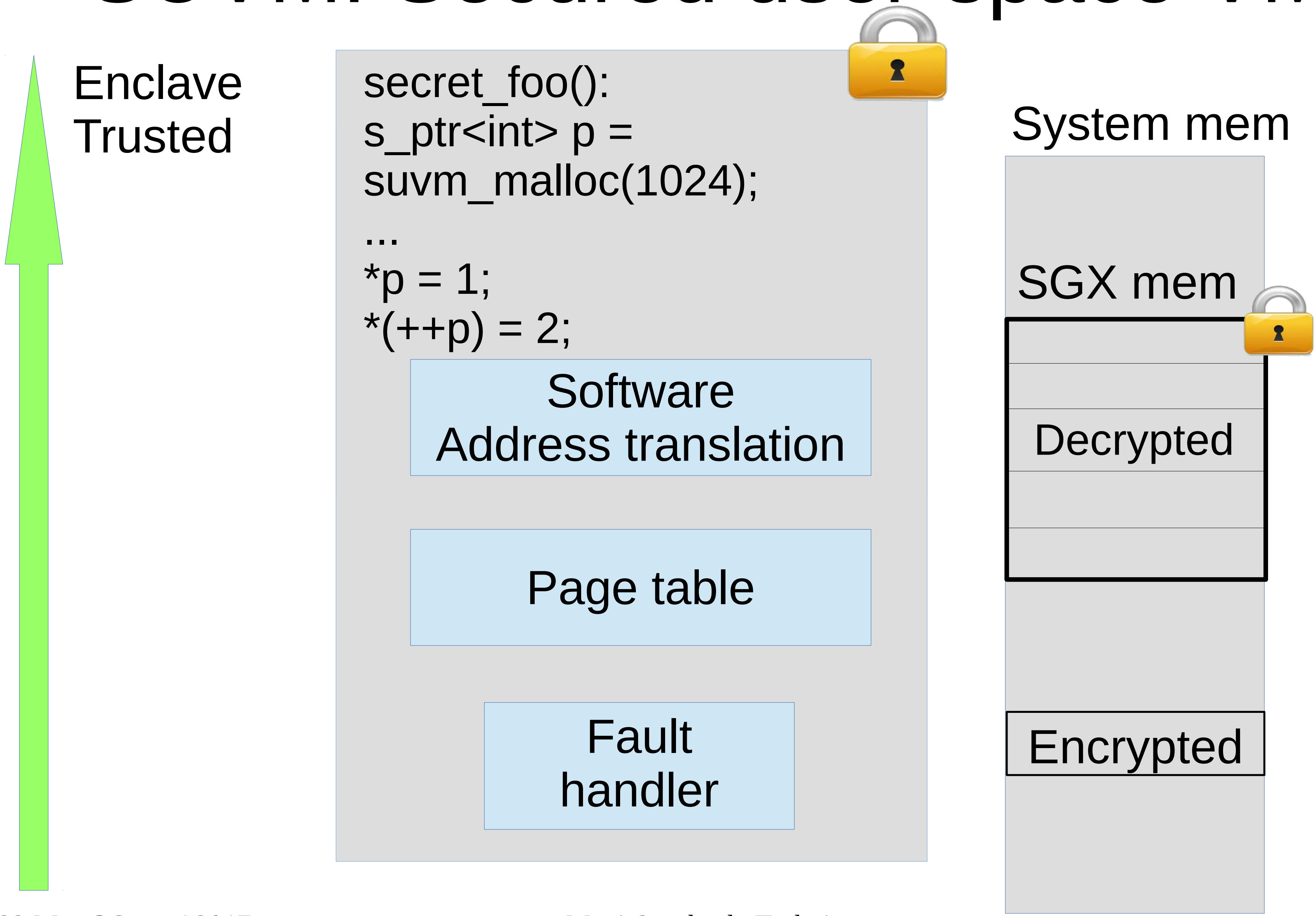
SUVM: Secured user-space VM



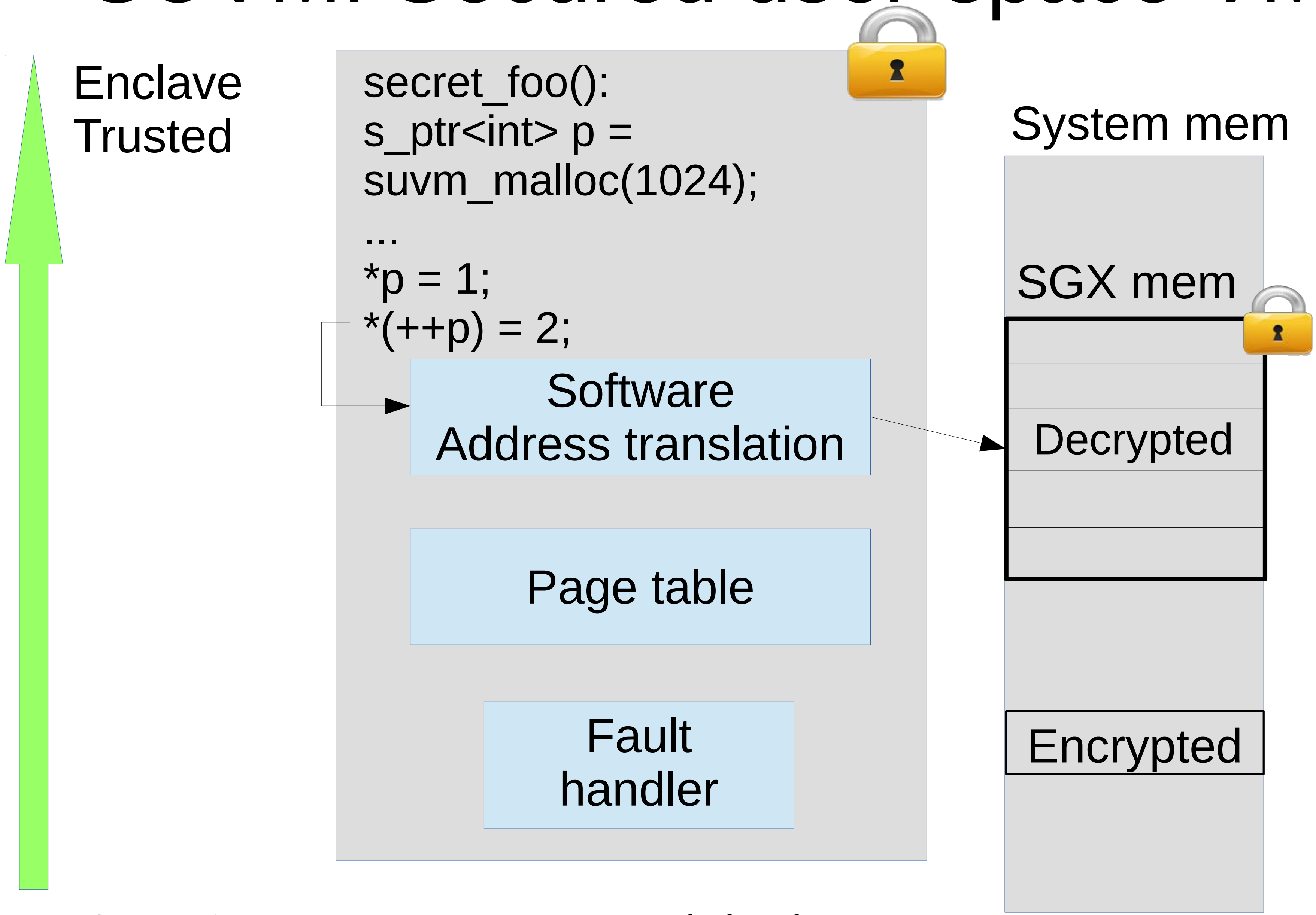
SUVM: Secured user-space VM



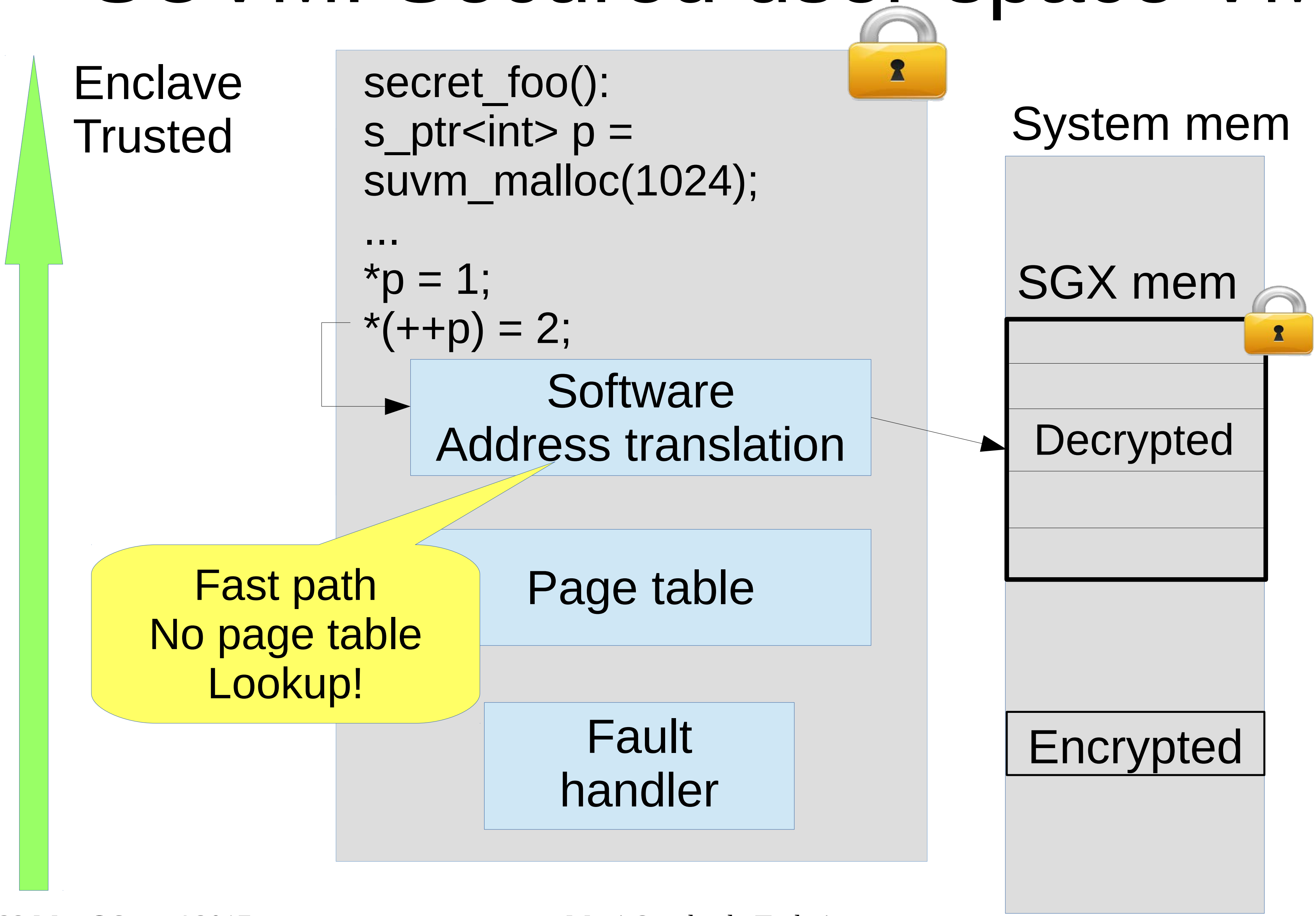
SUVM: Secured user-space VM



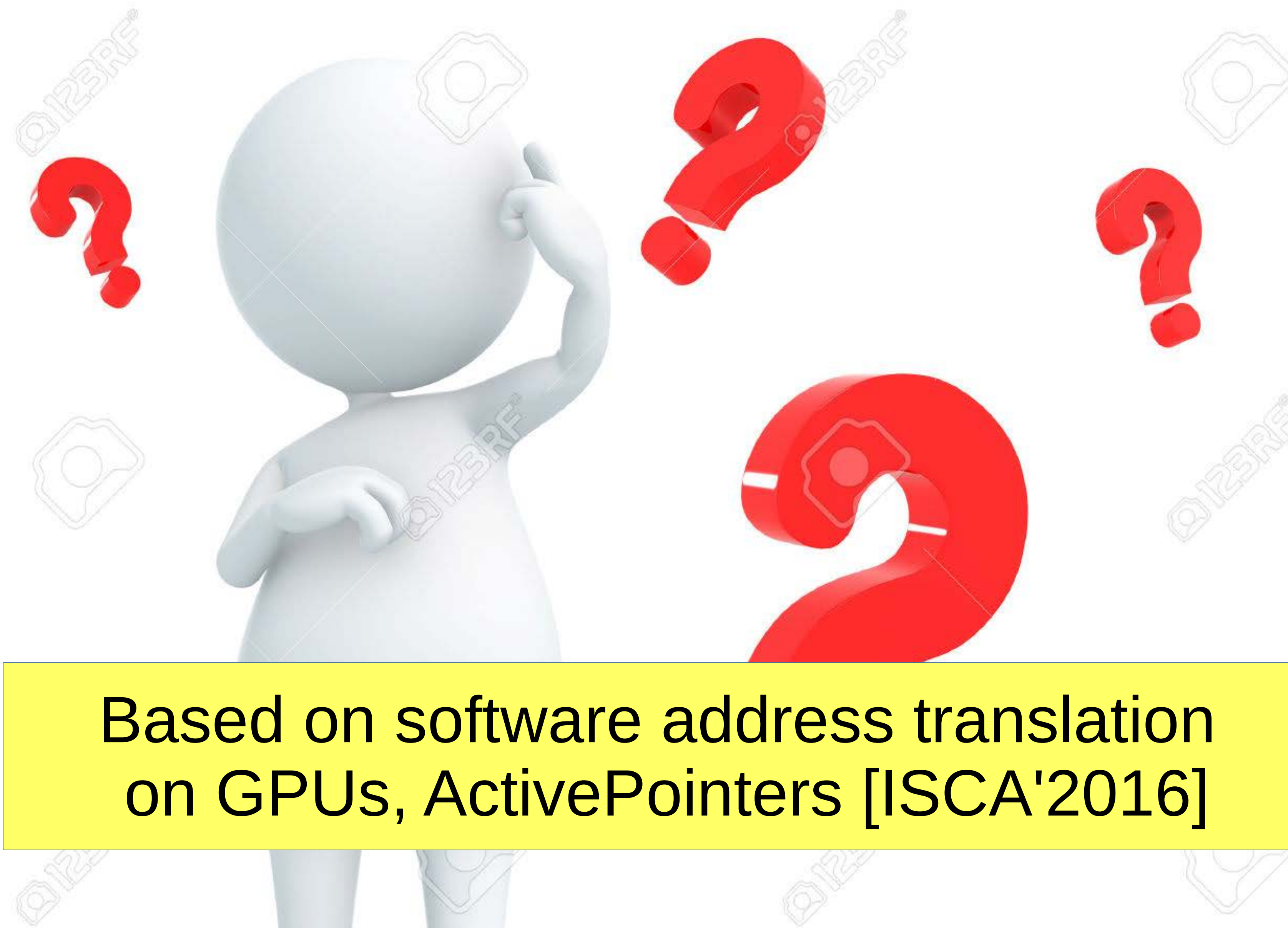
SUVM: Secured user-space VM



SUVM: Secured user-space VM



Wait...Software based VM management?



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

SUVM key contributions

- Multi-threaded

Compared to SGX:

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
- Multi-enclave memory coordination
- Write-back only dirty pages
- Sub-page direct access to backing store

Virtual Machine ballooning

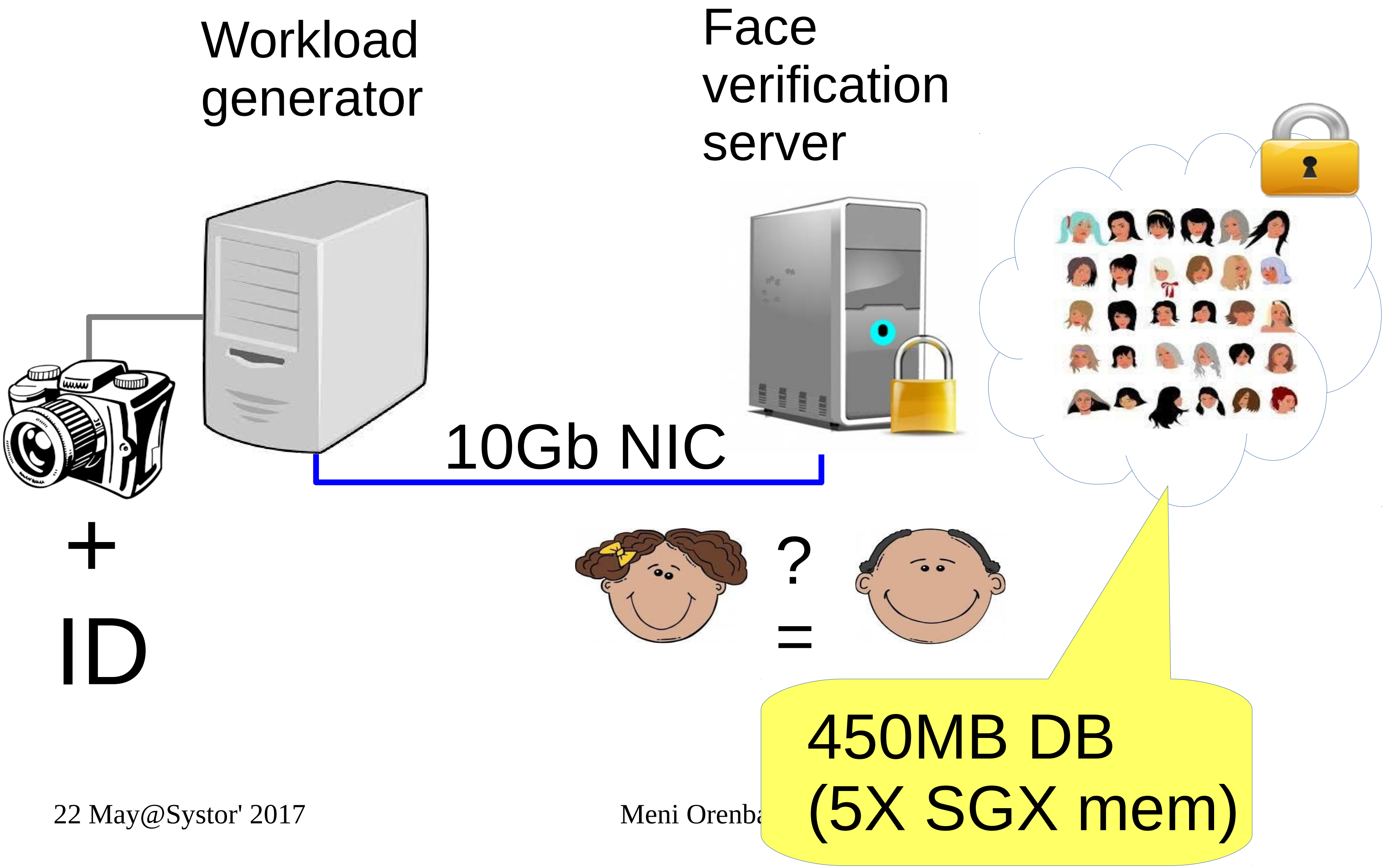
Software address translation offers new optimizations

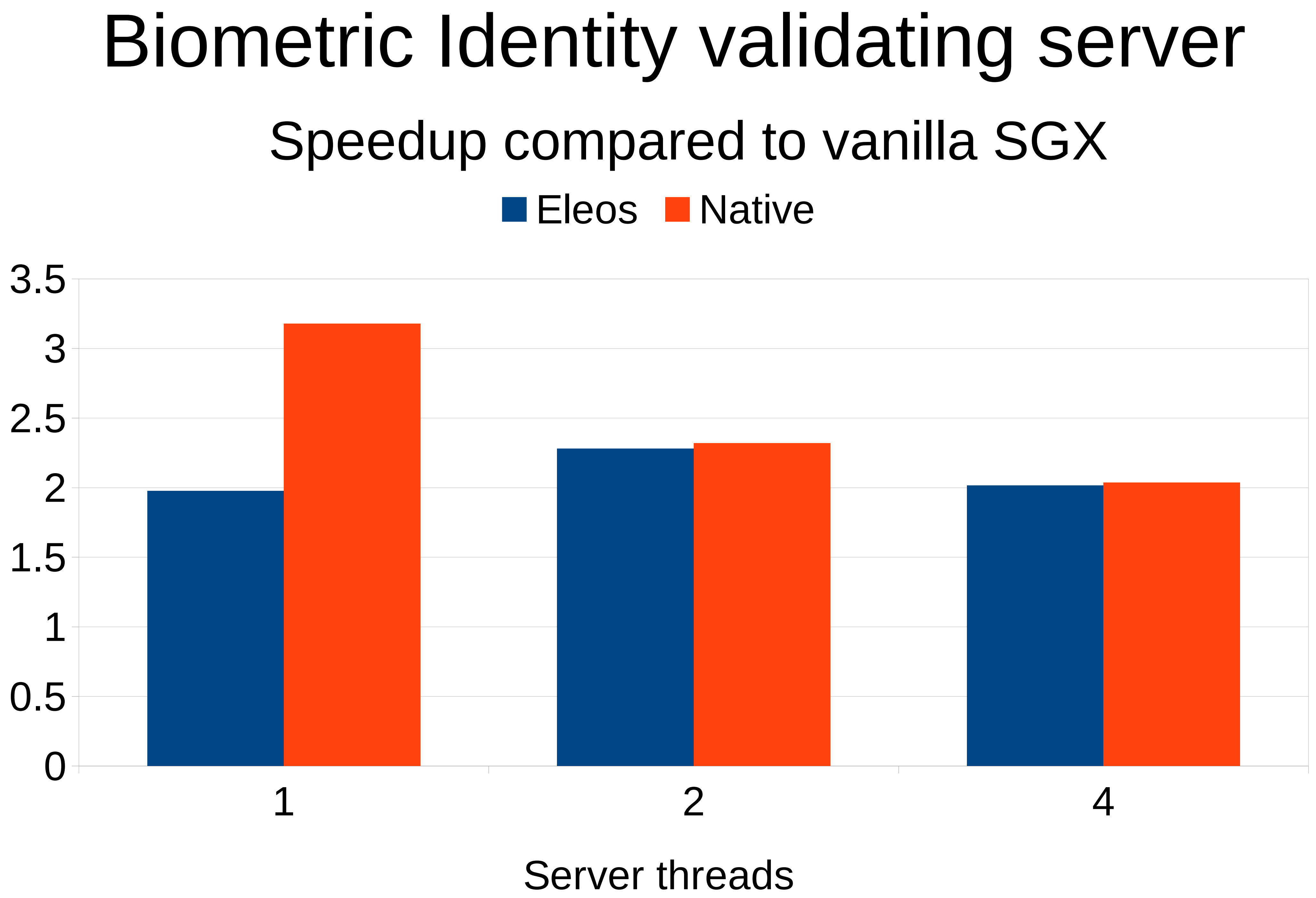
- Customized page size
- Customized eviction policy
- Multi-enclave memory coordination
- Write-back only dirty pages
- Sub-page direct access to backing store

Virtual Machine ballooning

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

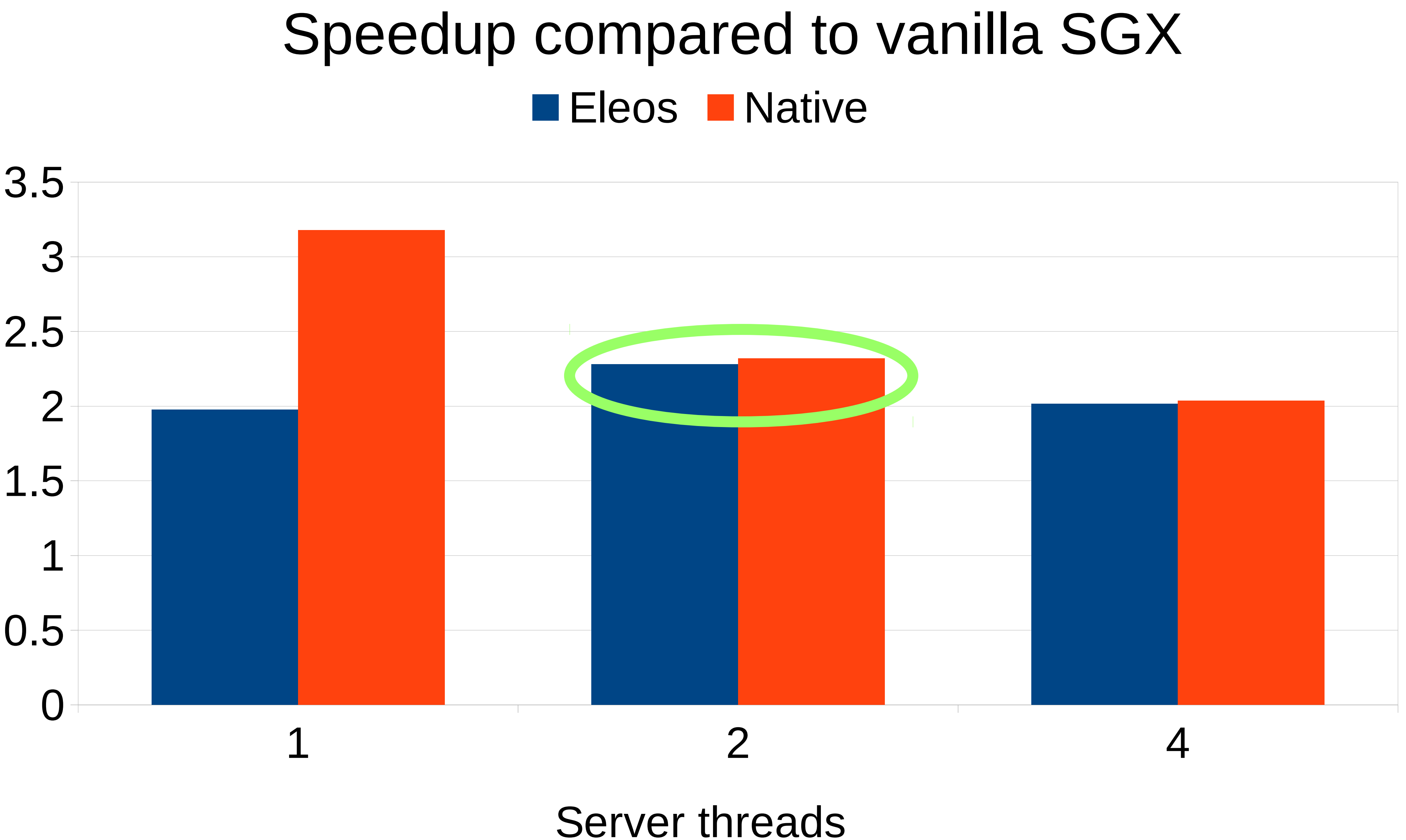
Biometric Identity checking server



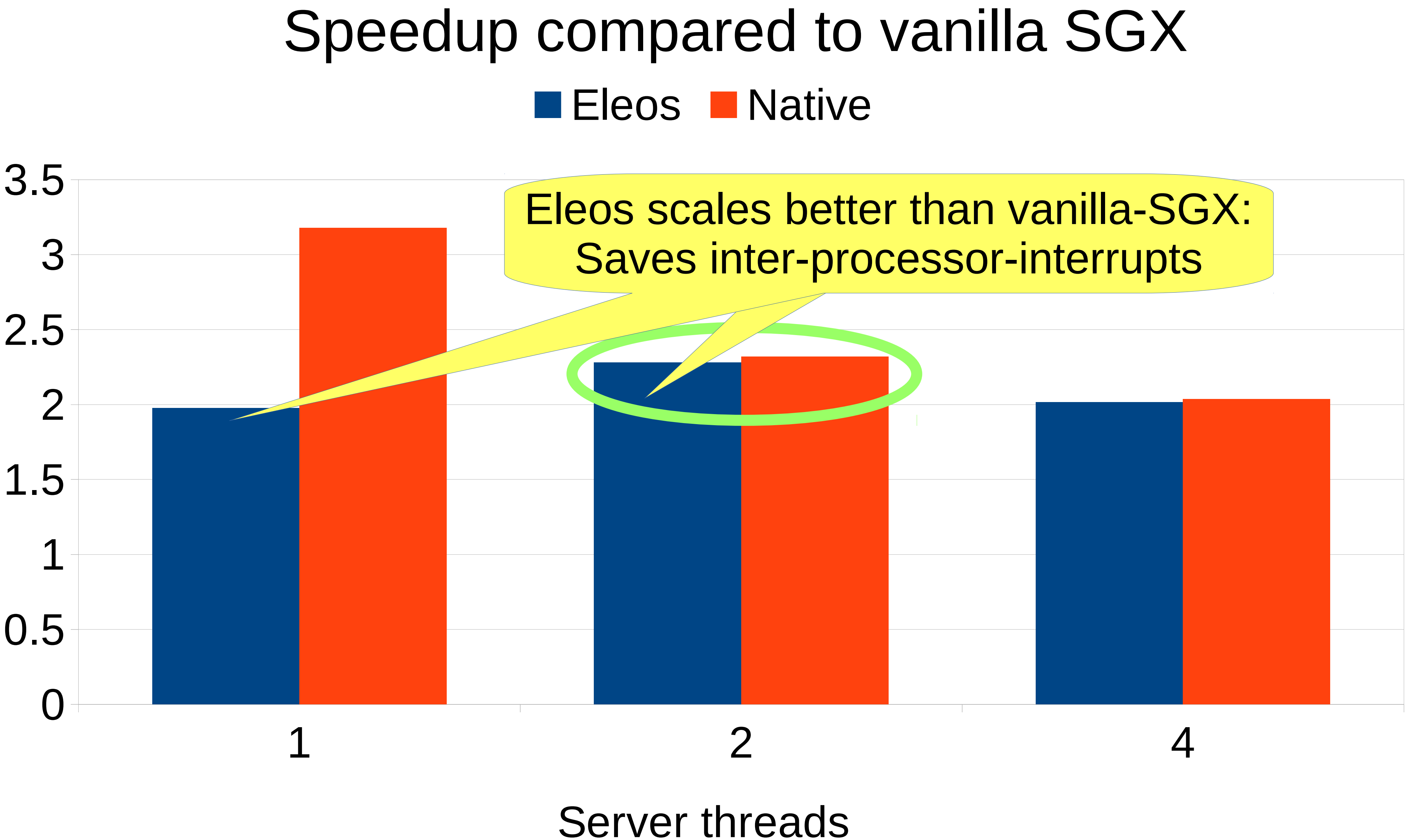


Biometric Identity validating server
Speedup compared to vanilla SGX
Eleos
Native
3.5
3
2.5
2
1.5
1
0.5
0
1
2
4
Server threads

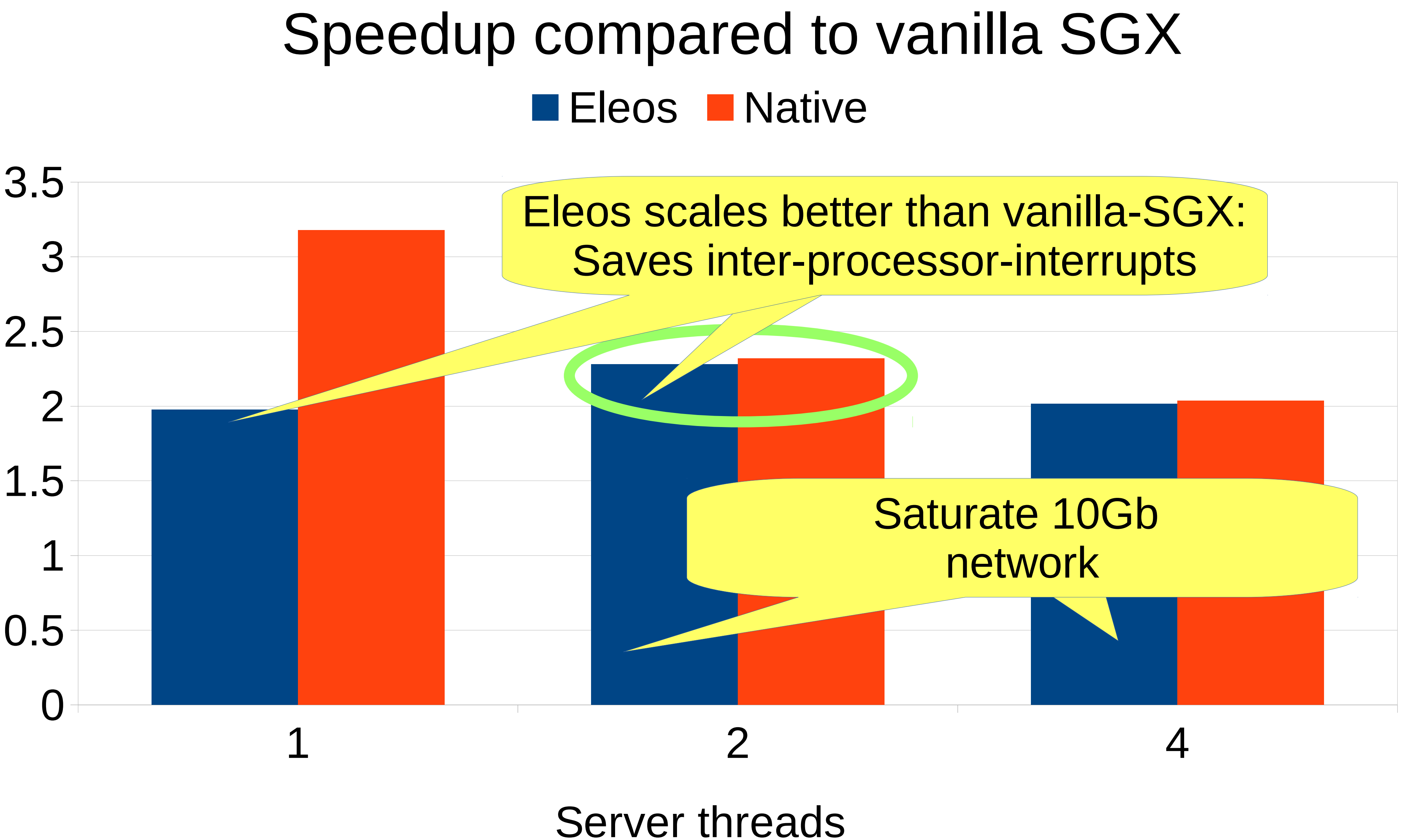
Biometric Identity validating server



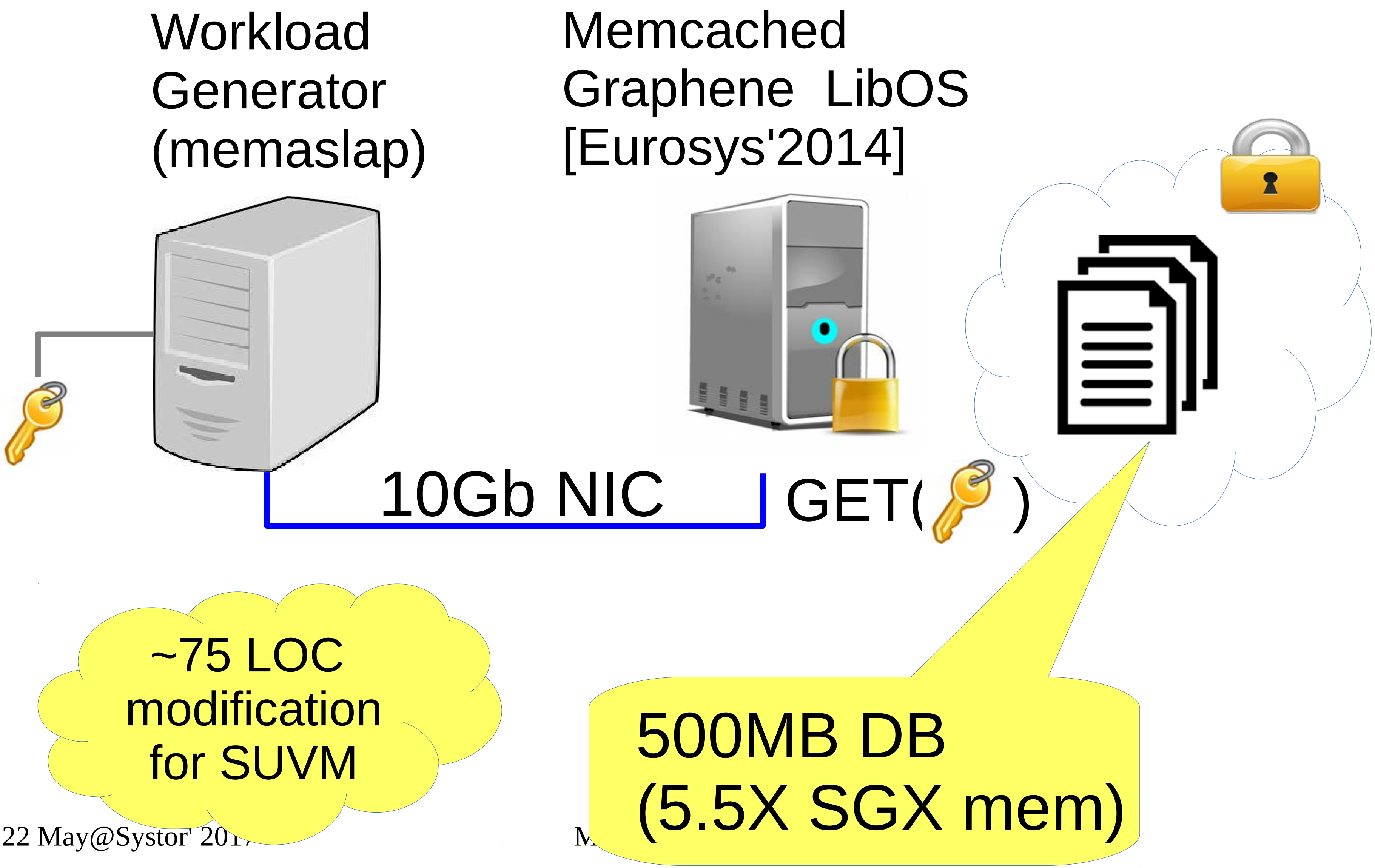
Biometric Identity validating server



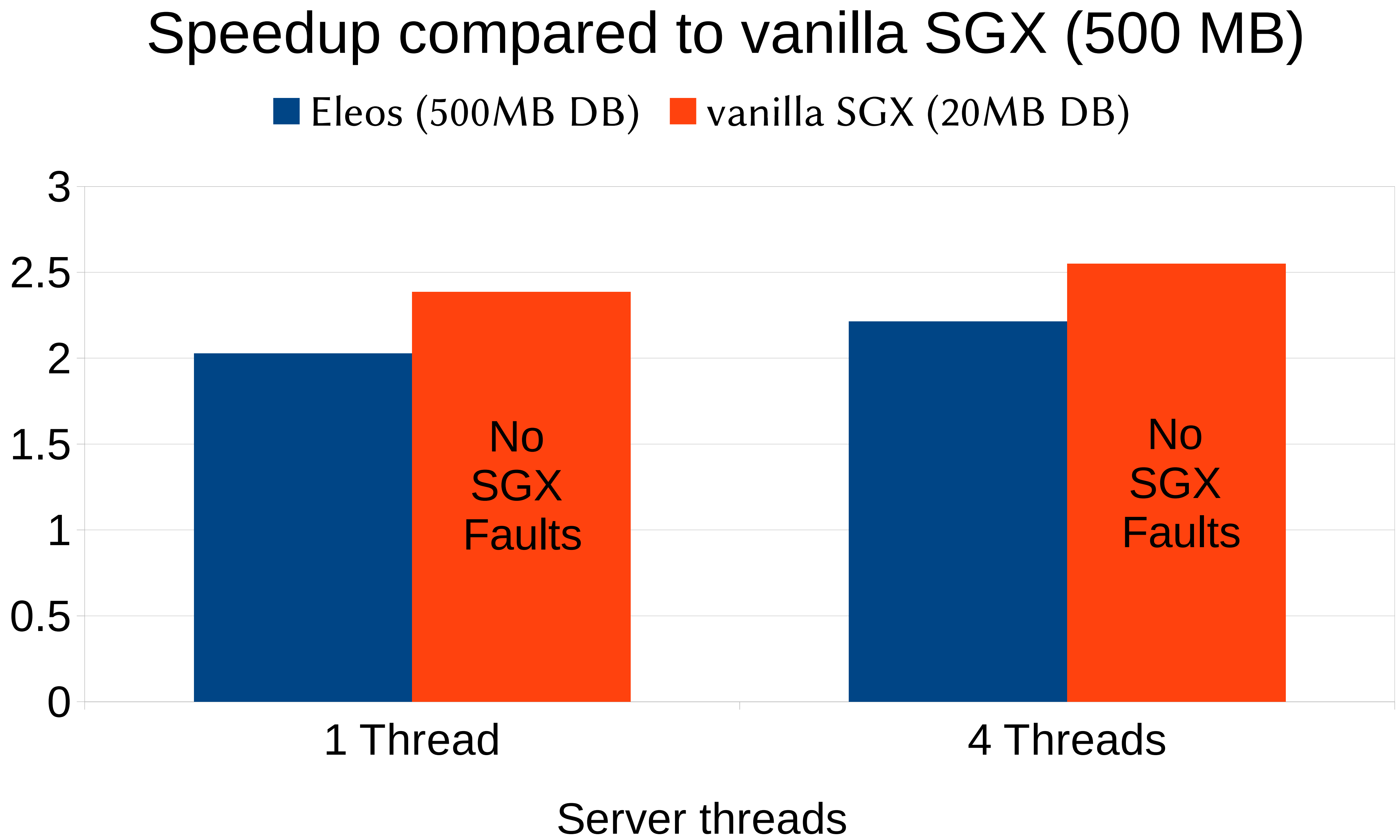
Biometric Identity validating server



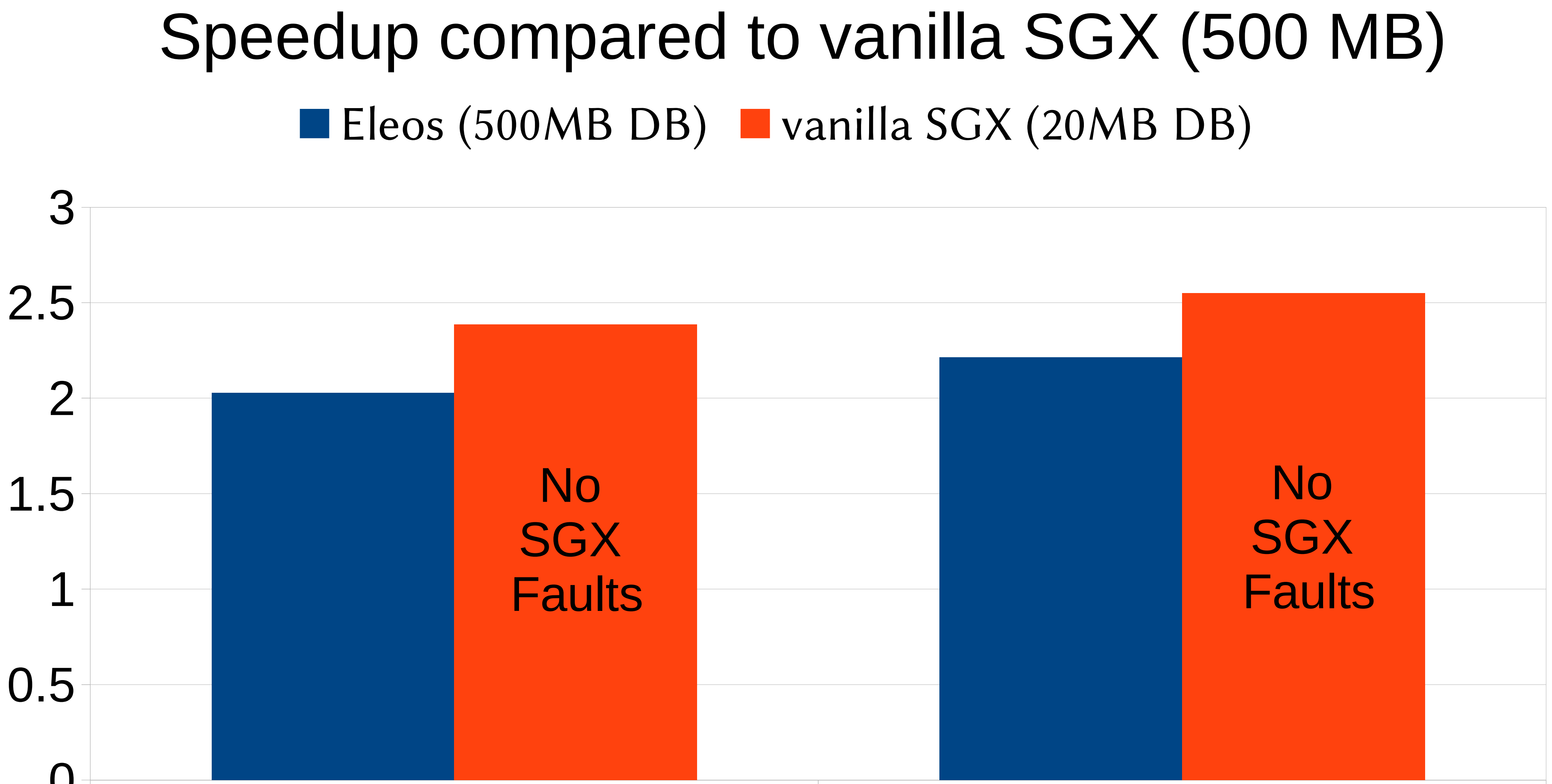
Memcached


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)
3
2.5
2
1.5
1
0.5
0
No SGX Faults
No SGX Faults
1 Thread
4 Threads
Server threads

Memcached

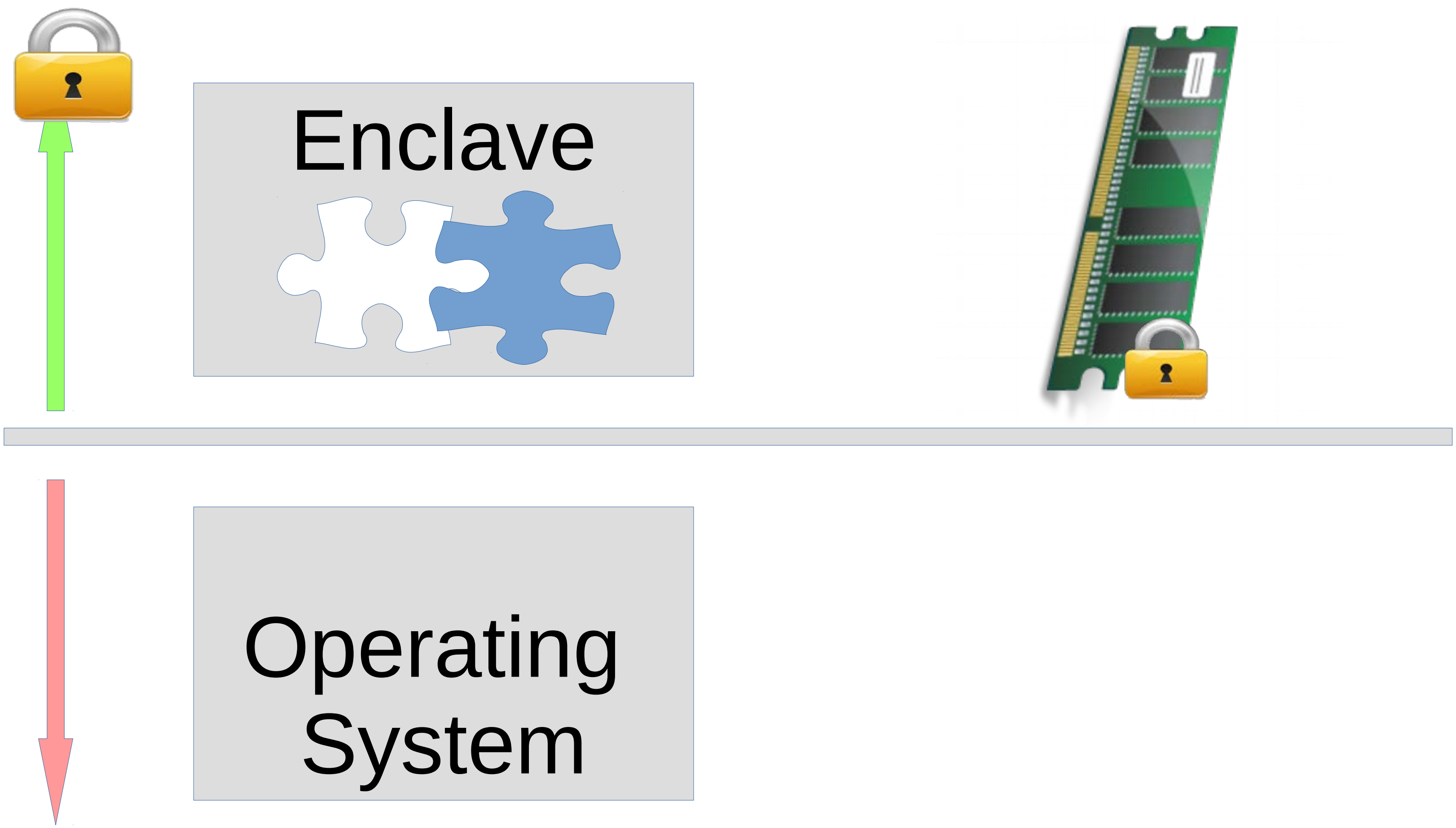


Disclaimer: Eleos+Graphene is 3x slower than native

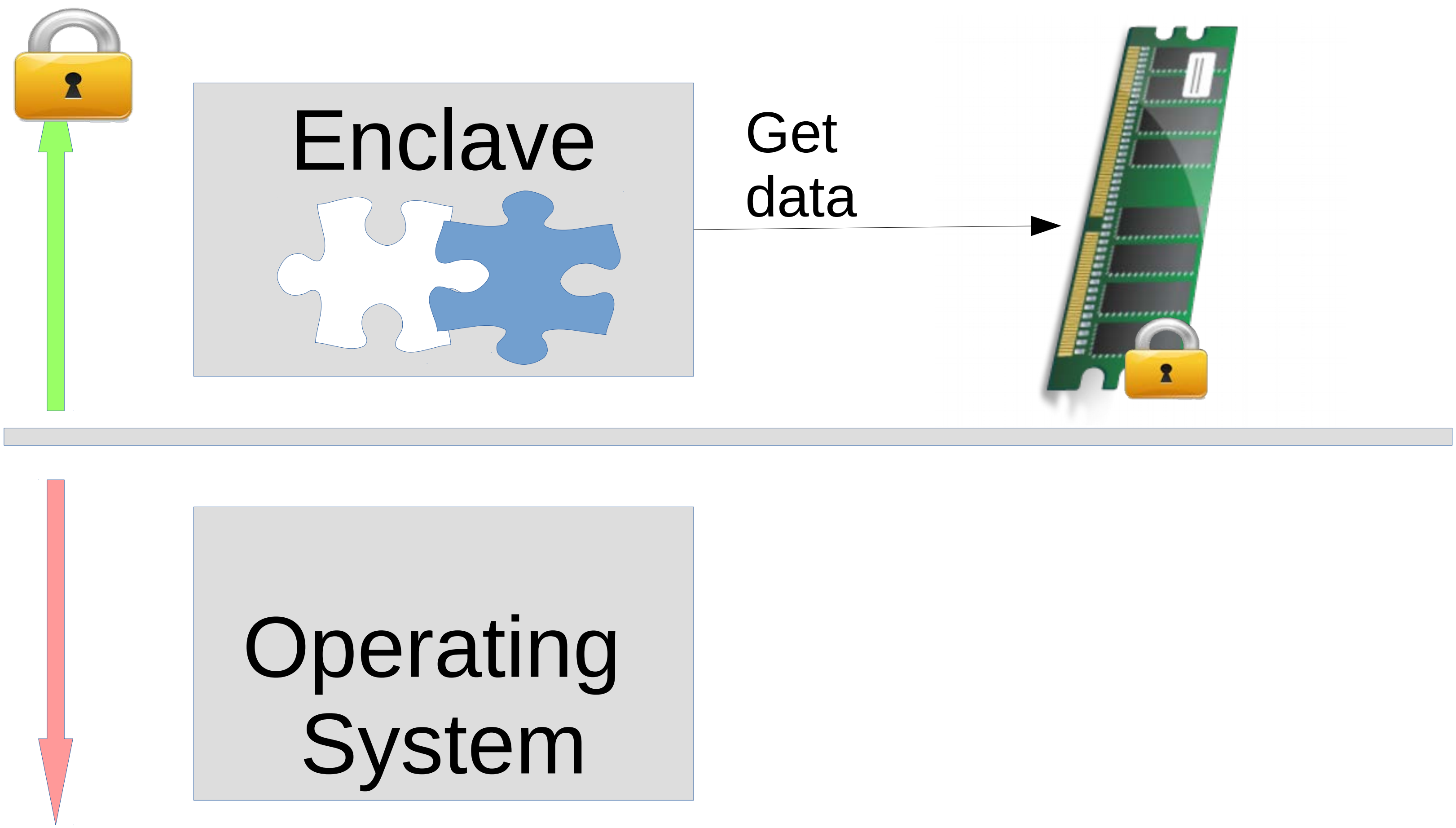
Take aways

- 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 small TCB

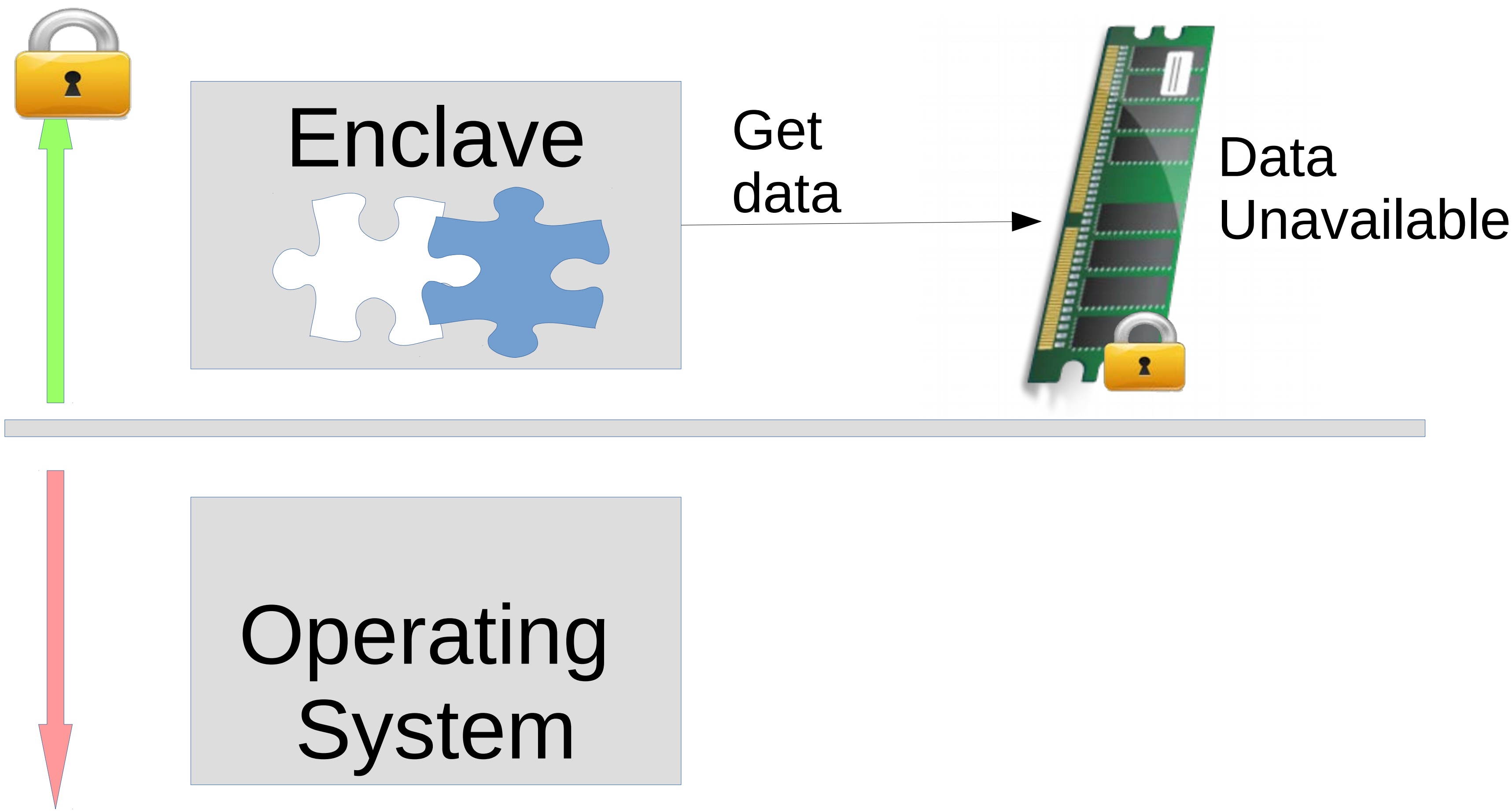
Traditional SGX:
Host-centric OS services



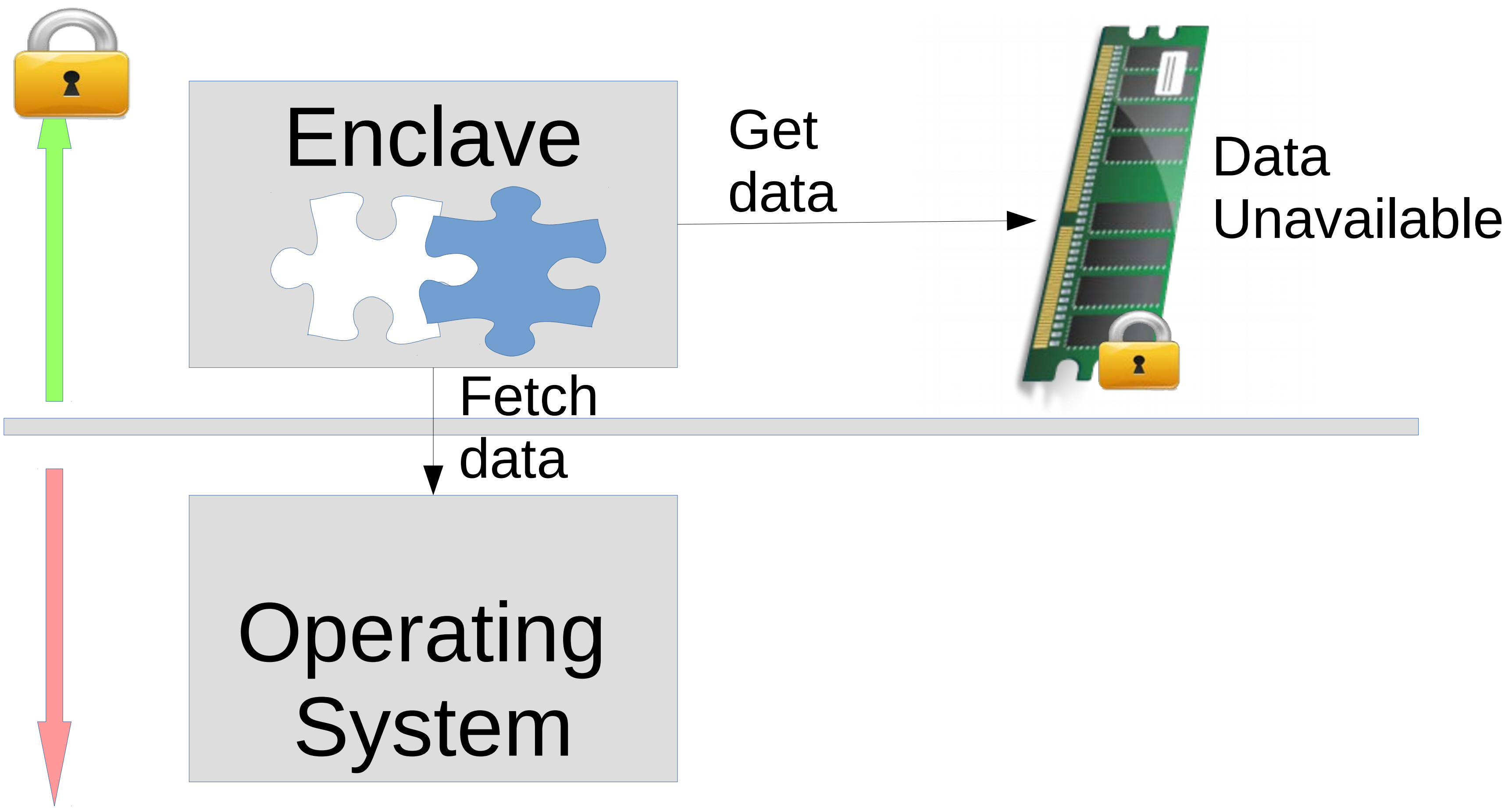
Traditional SGX:
Host-centric OS services



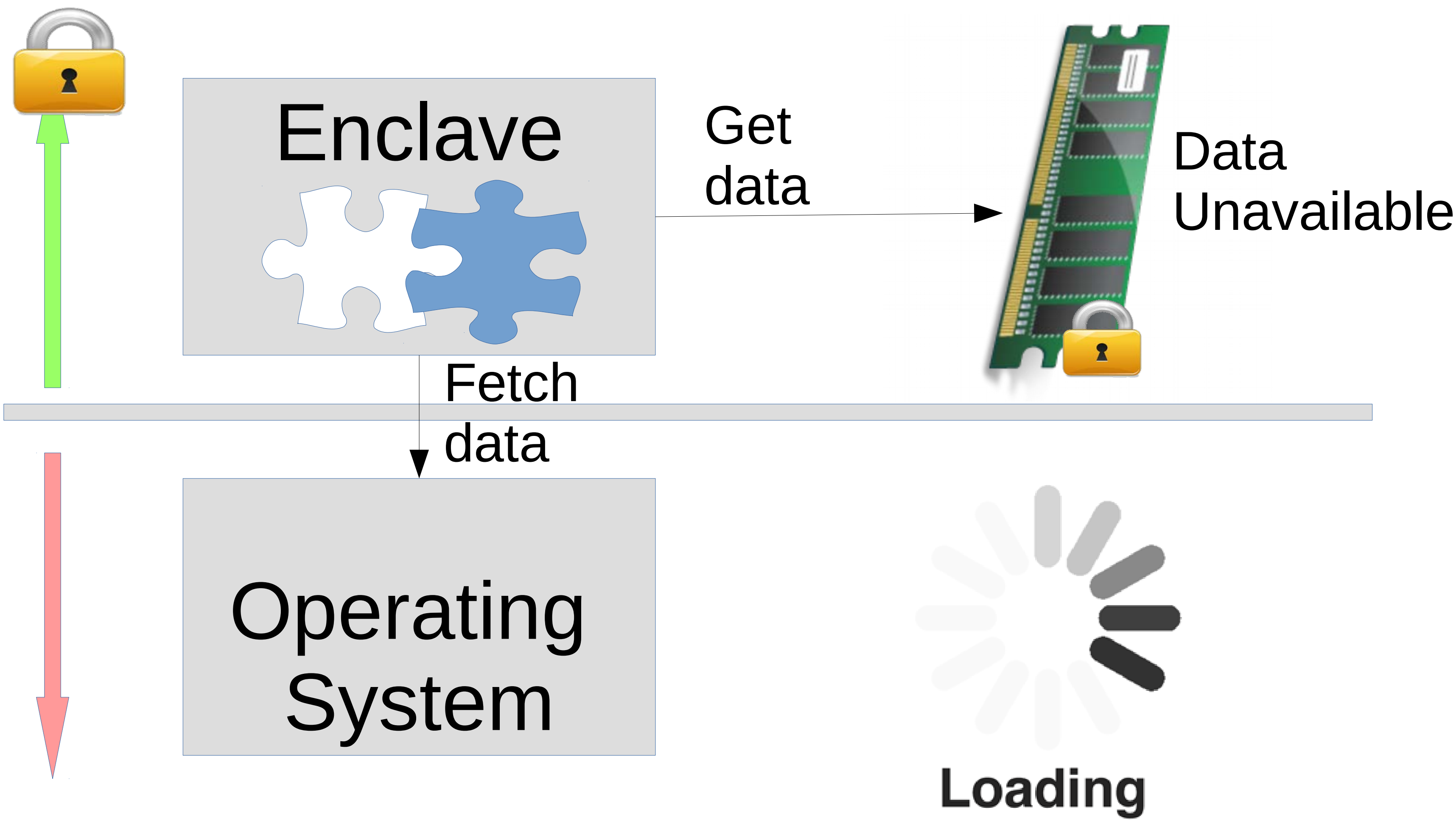
Traditional SGX:
Host-centric OS services



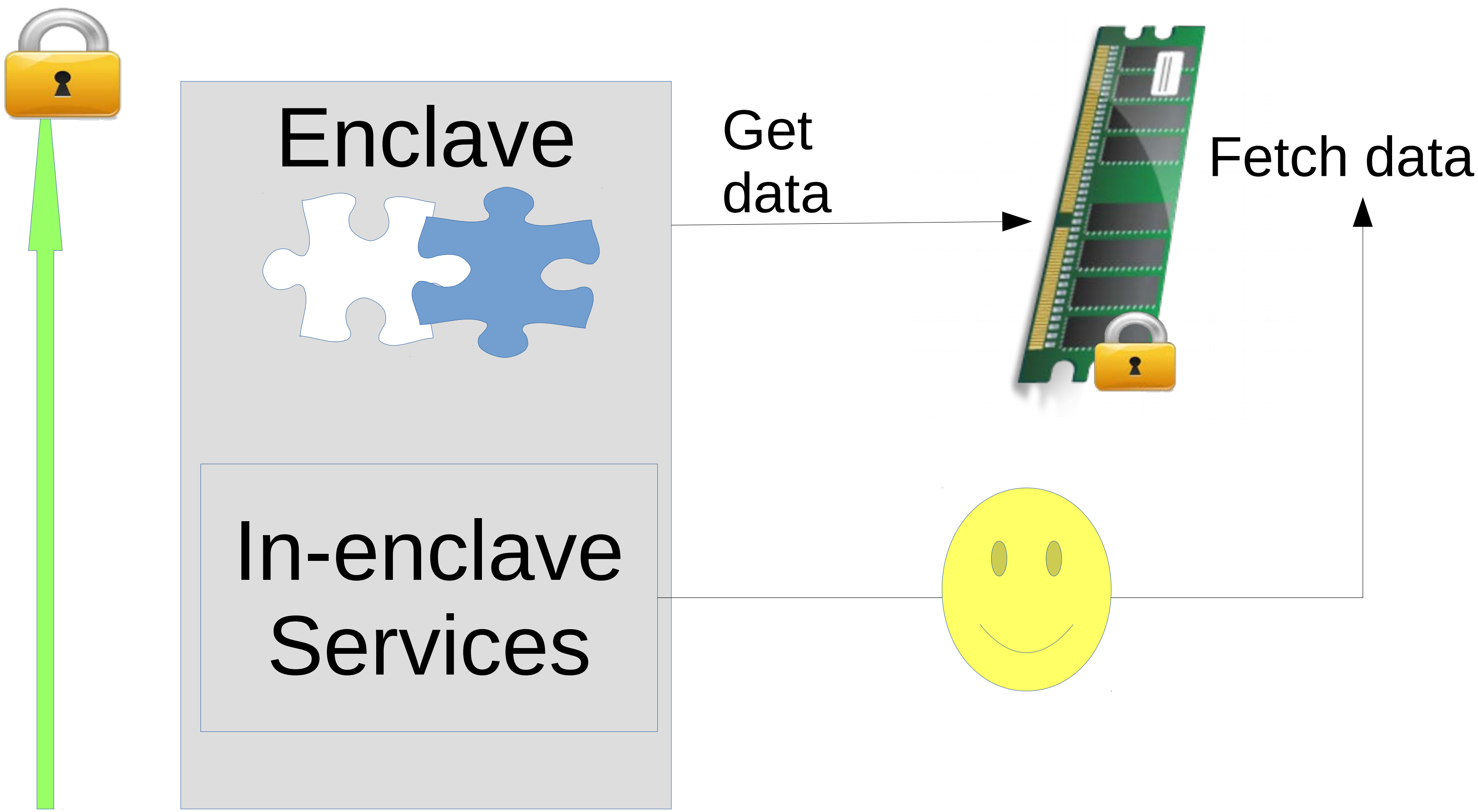
Traditional SGX:
Host-centric OS services



Traditional SGX: Host-centric OS services



Eleos Insight:
Enclave-centric OS services



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

<u>More information at</u>:

“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