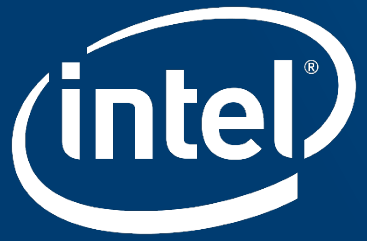


EnclaveDom

Privilege Separation for Large-TCB Applications in Trusted Execution Environments

Marcela S. Melara

Joint work with Mic Bowman, Michael J. Freedman (Princeton)

Legal Disclaimers

- Intel provides these materials as-is, with no express or implied warranties.
- All products, dates, and figures specified are preliminary, based on current expectations, and are subject to change without notice.
- Intel processors, chipsets, and desktop boards may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.
- Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No product or component can be absolutely secure. Check with your system manufacturer or retailer or learn more at http://intel.com.
- Some results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.
- Intel and the Intel logo are trademarks of Intel Corporation in the United States and other countries.
- *Other names and brands may be claimed as the property of others.

Library-Centric Software Development

Save time and effort

Easy availability online

Library-Centric Software Development

Save time and effort

Easy availability online

No time/expertise to properly vet

Risky supply chain

Risky Third-Party Library Supply Chain

PC REVIEWS BEST PICKS HOW-TO NEWS SMART HOME BUSINESS SHOP

#FitbitVersa2 #SwitchLite #GoogleCalendarSpam #KernelPanic #iPhone11Rumors

News & Analysis

Malware Discovered in Popular Android App CamScanner

The malicious code was sourced back to a third-party advertising library on the app that could use a victim's Android phone to download additional malware. CamScanner says it will take 'immediate legal actions against' the scammers.

By Michael Kan August 28, 2019 2:29PM EST

ars TECHNICA BIZ & IT TECH SCIENCE POLICY CARS GAMING & CULTURE

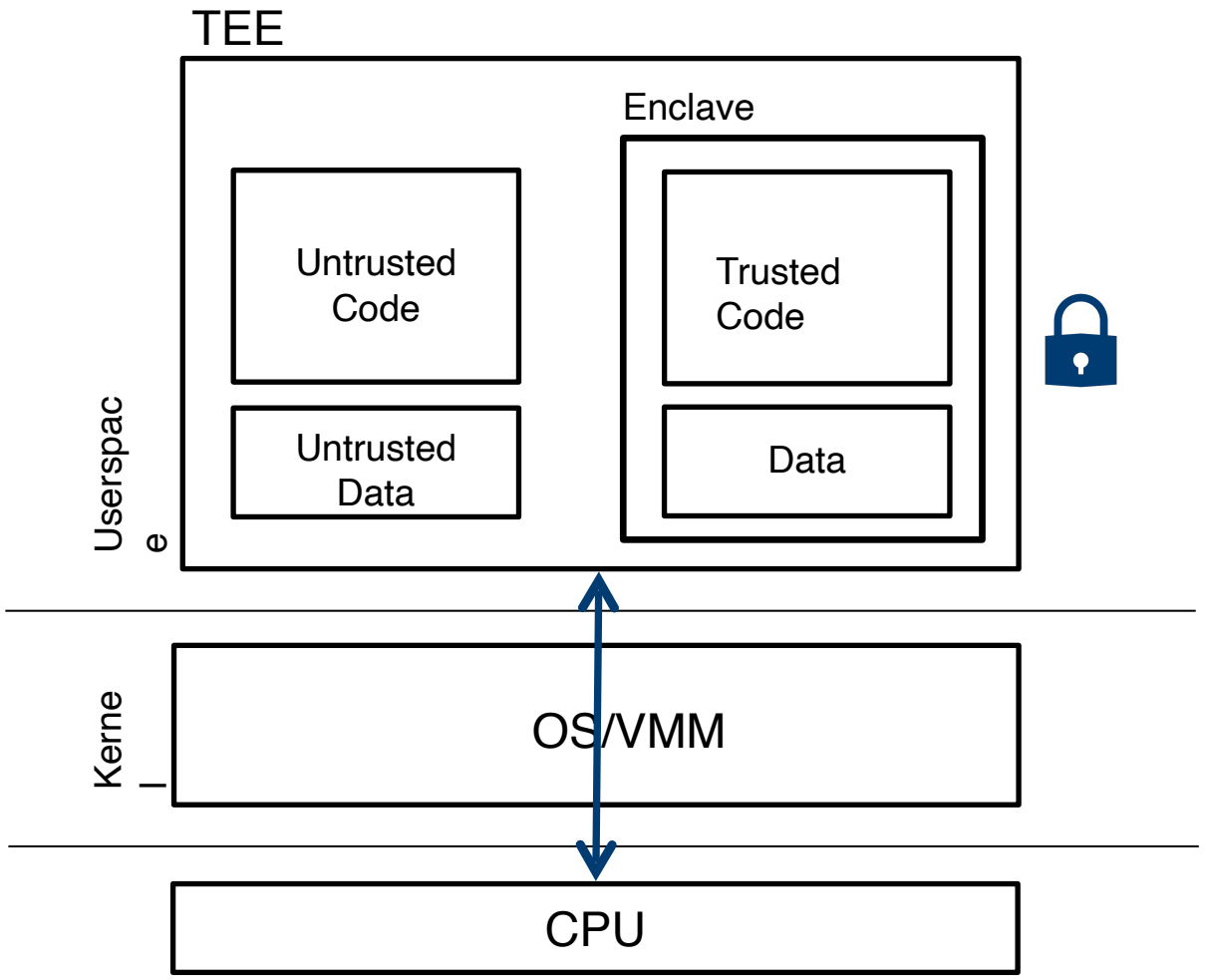
BIZ & IT —

Rage-quit: Coder unpublished 17 lines of JavaScript and "broke the Internet"

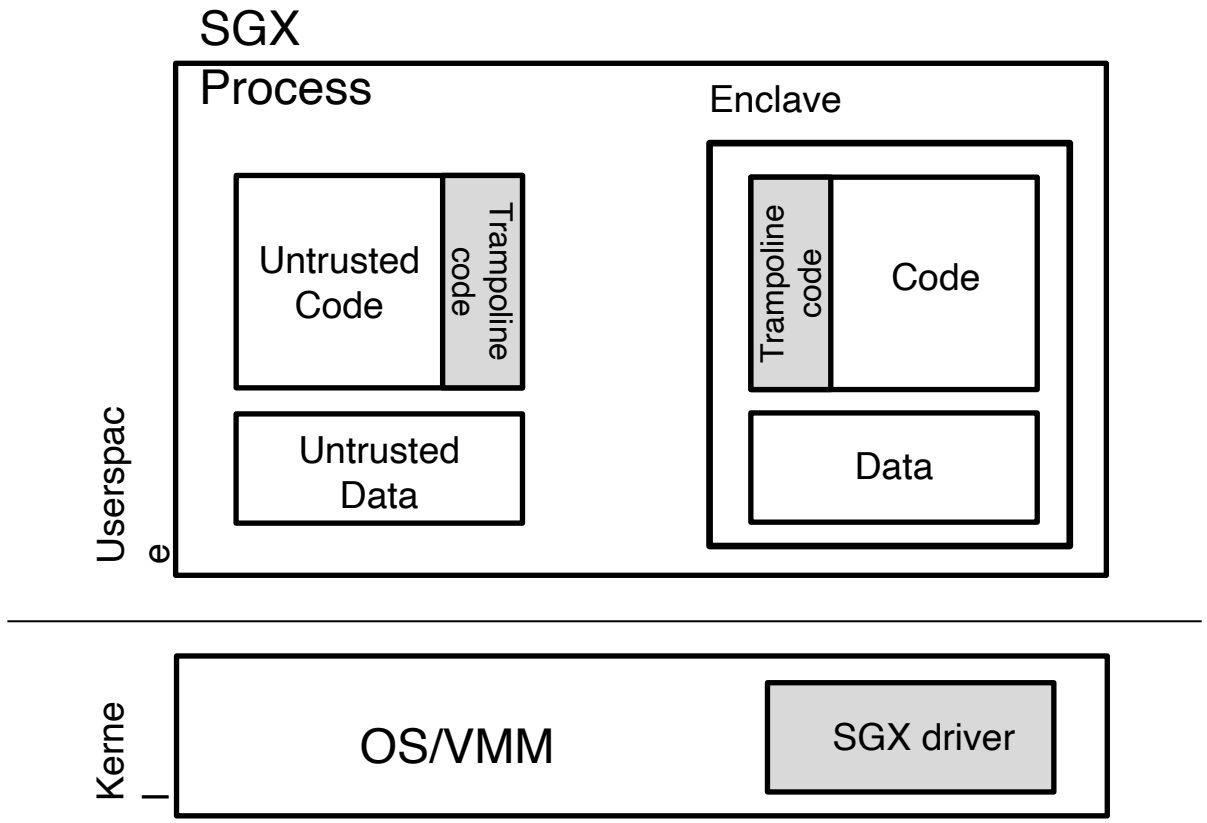
Dispute over module name in npm registry became giant headache for developers.

SEAN GALLAGHER - 3/24/2016, 10:10 PM

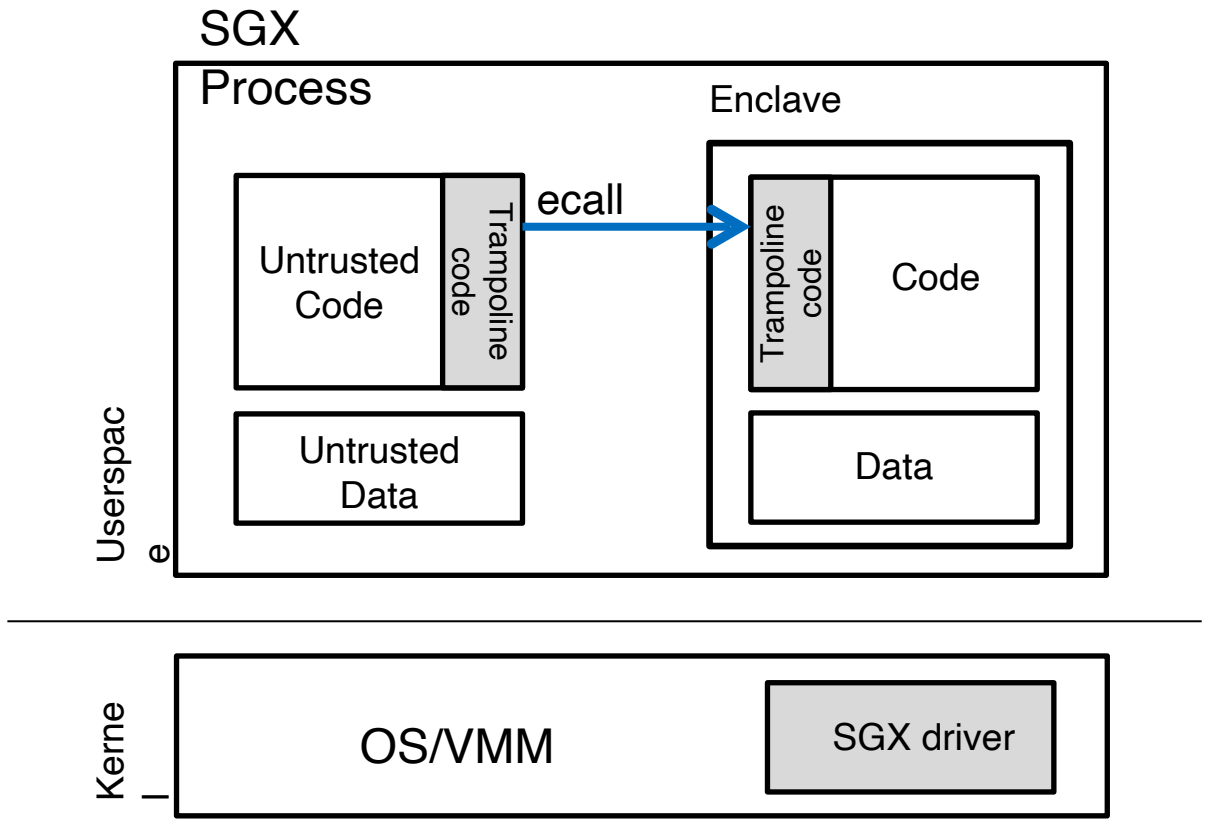
Trusted Execution Environments



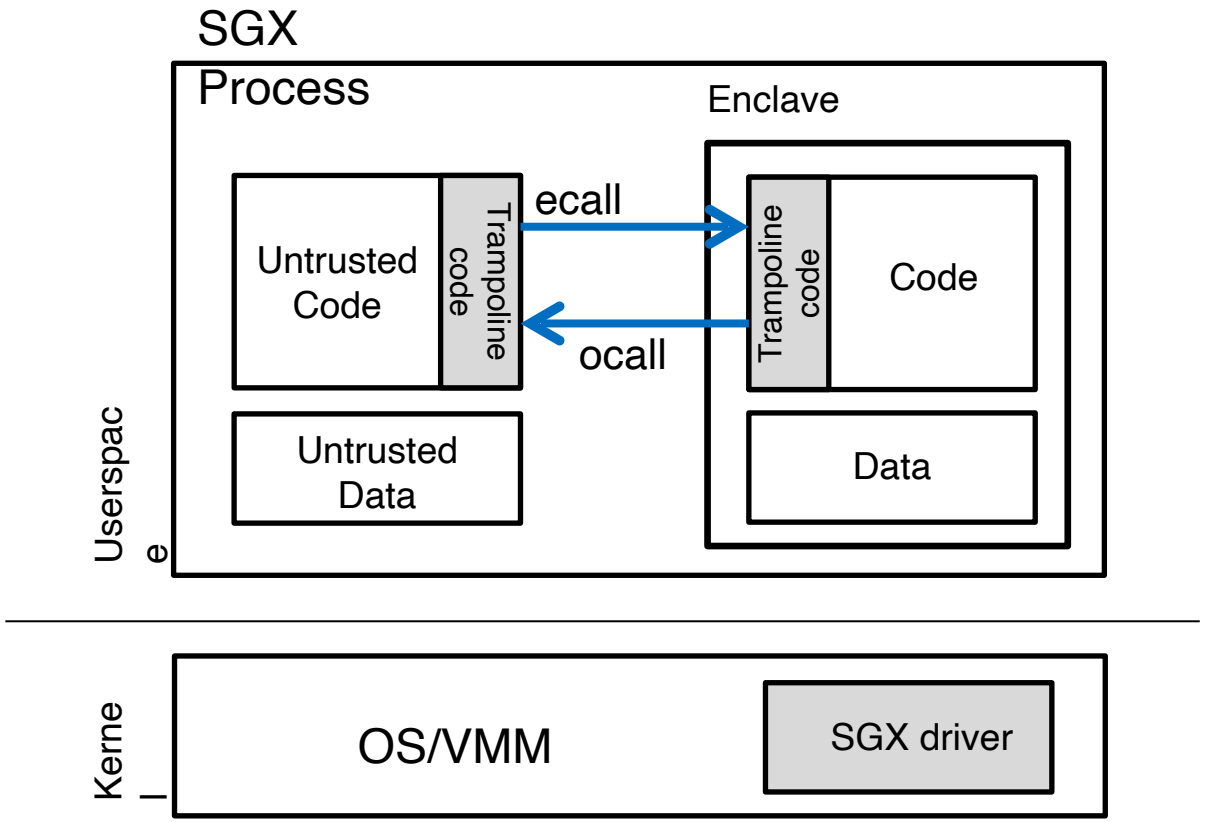
Intel SGX Primer



Intel SGX Primer



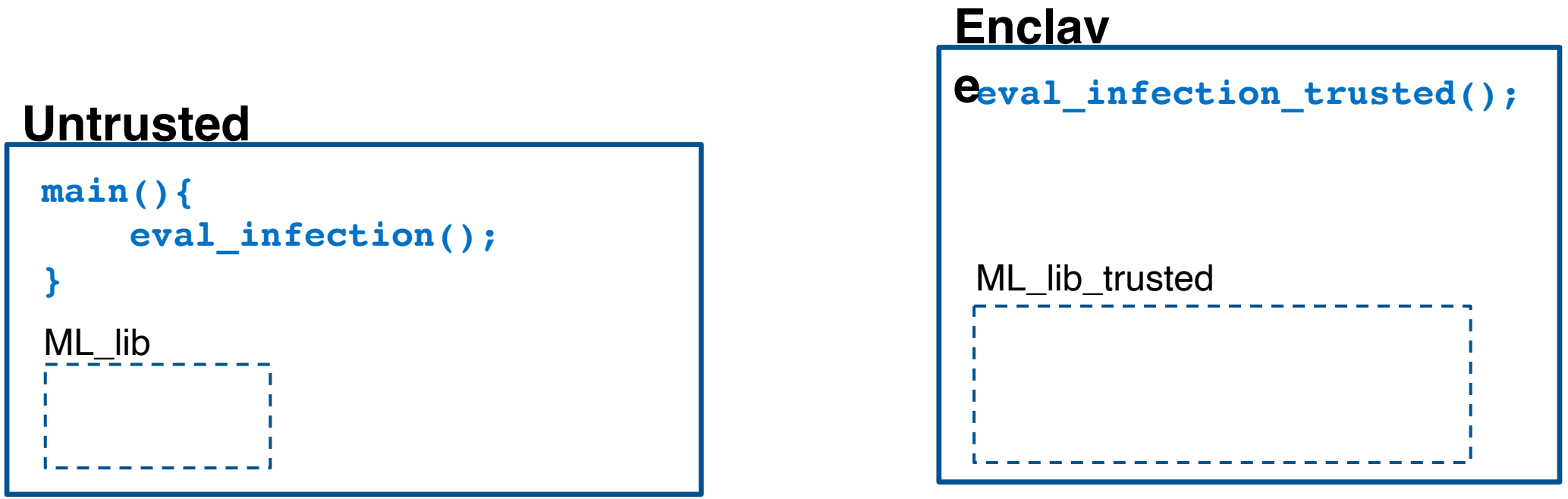
Intel SGX Primer



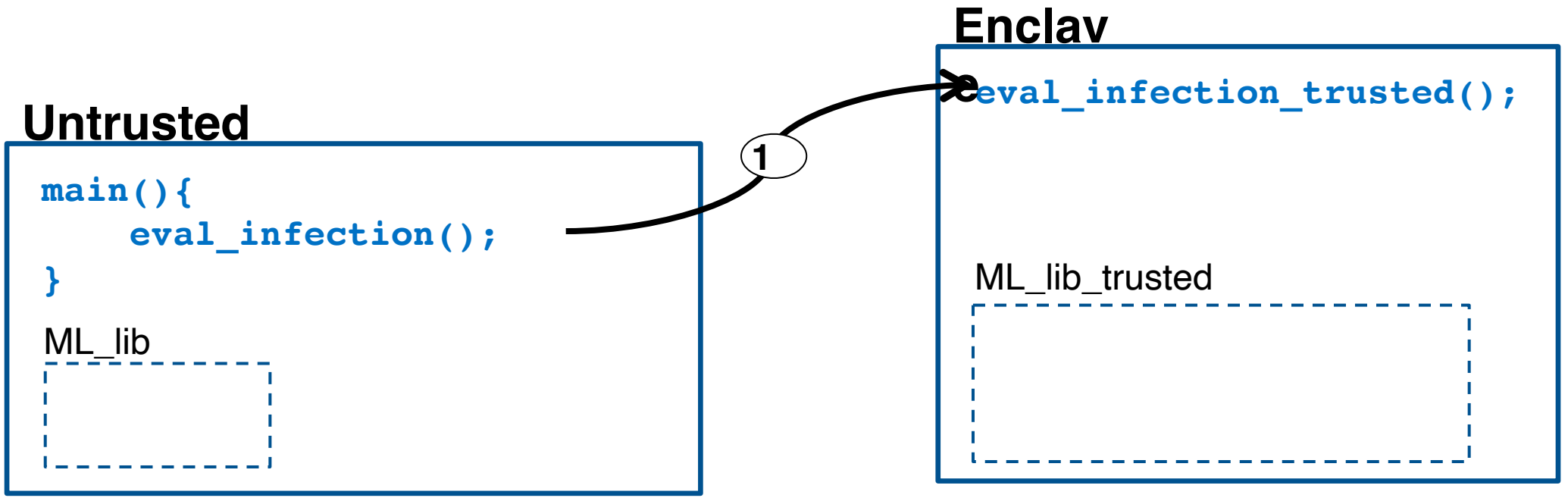
Toy Example: Image classification app

```
eval_infection(){
    ML_lib.load_training_set();

    data_file = open(“x-ray.jpg”);

    result = ML_lib.classify(data_file);
}
```

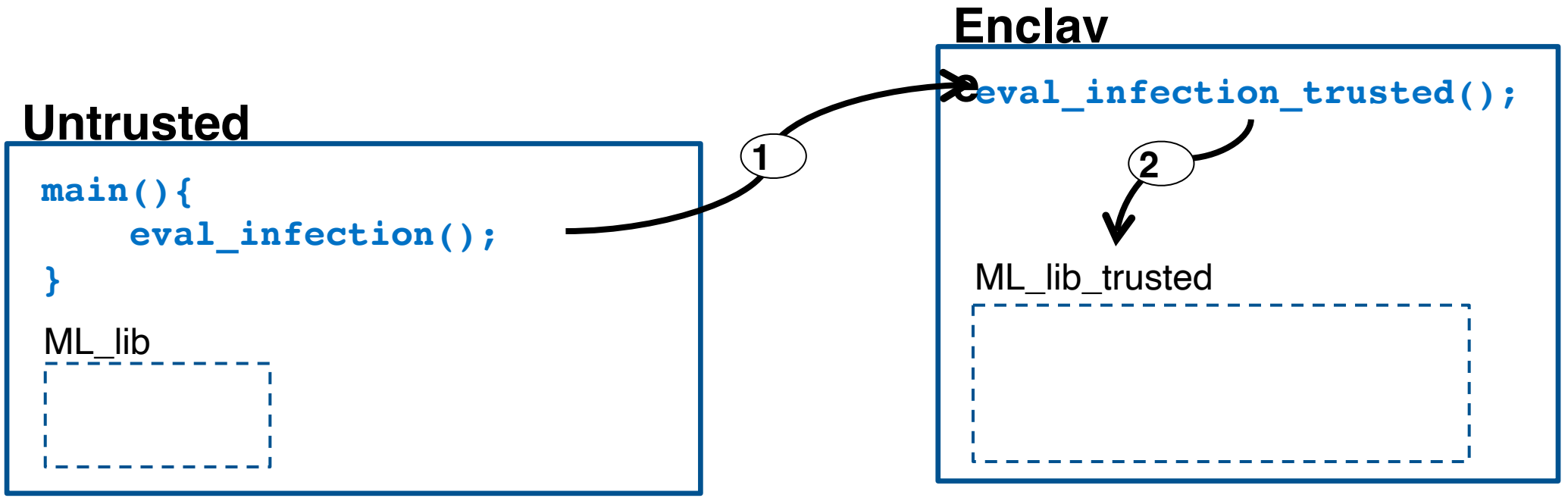
SGX in theory: Sensitive apps, minimal TCB



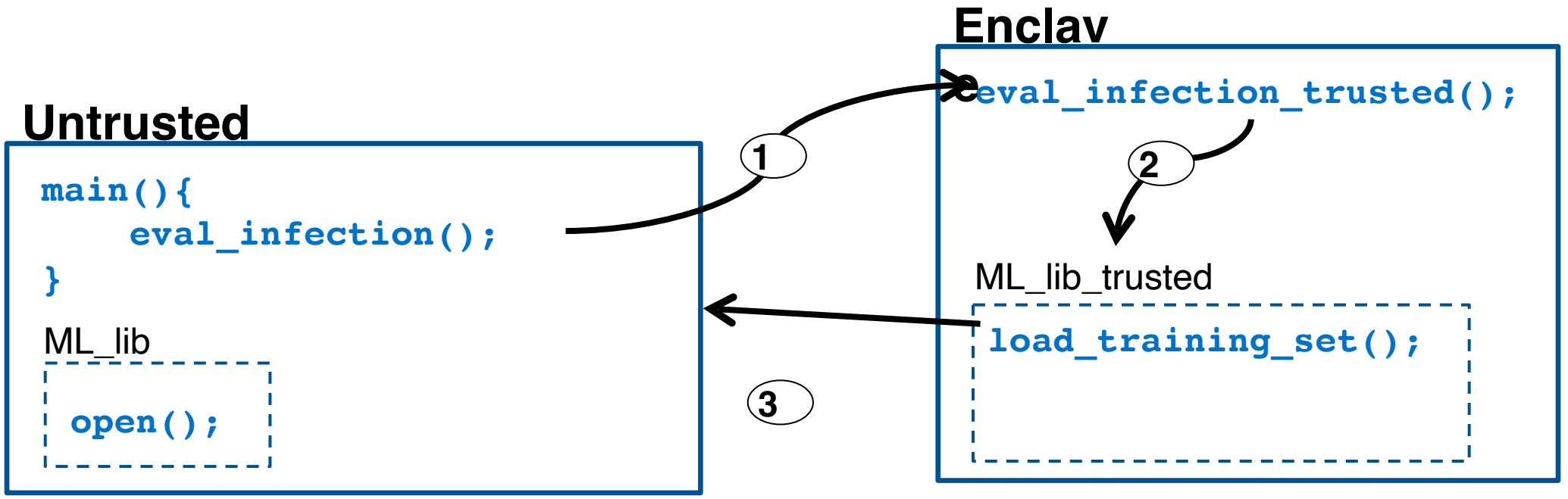
SGX in theory: Sensitive apps, minimal TCB



SGX in theory: Sensitive apps, minimal TCB

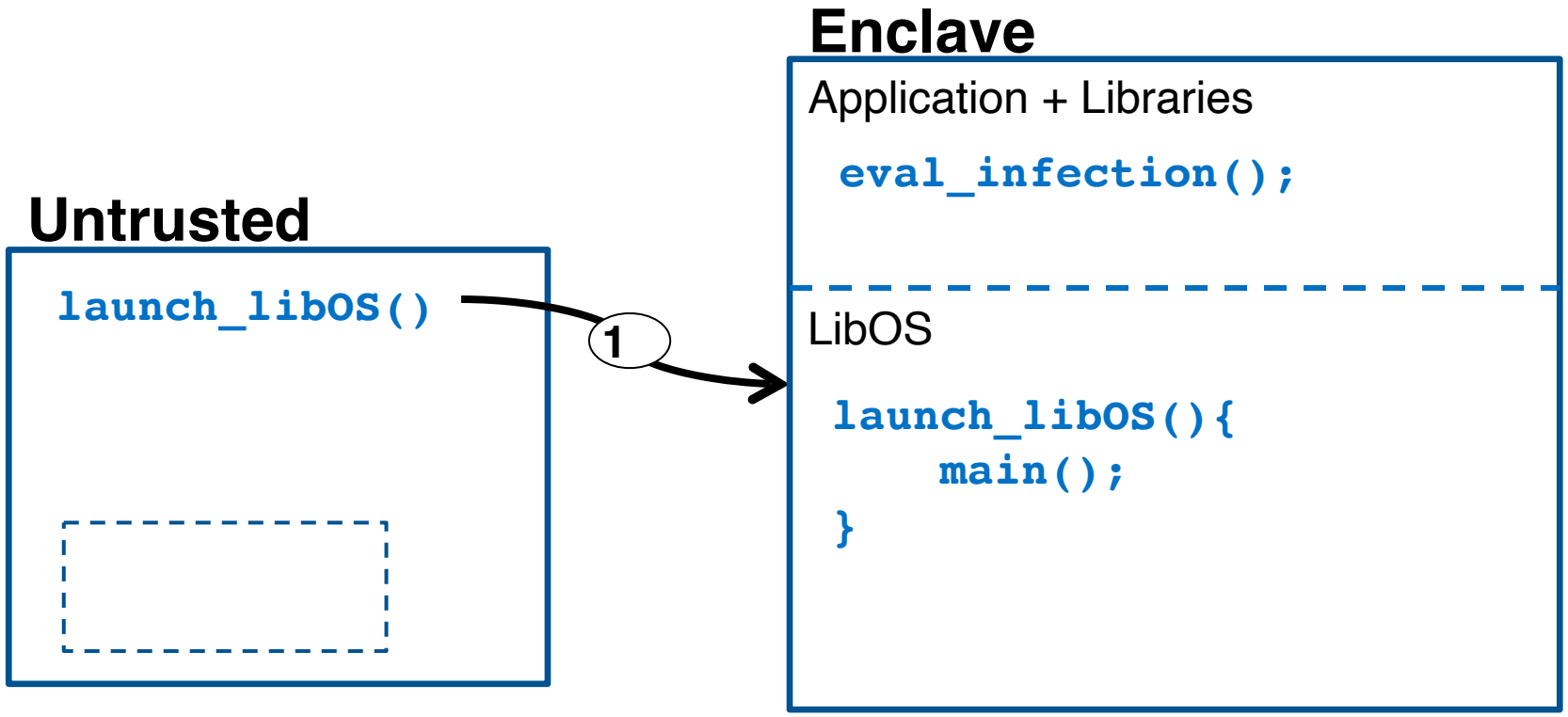


SGX in theory: Sensitive apps, minimal TCB

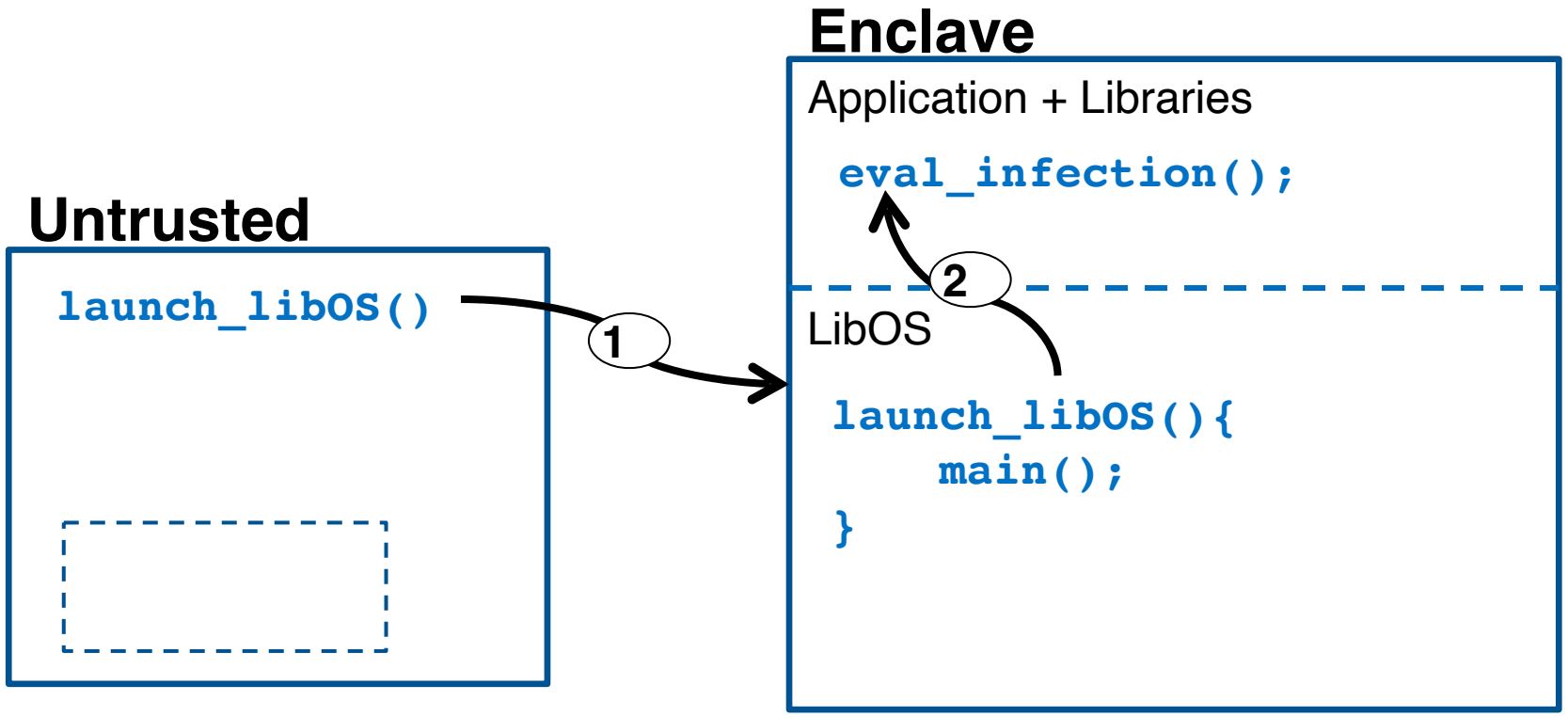


Native TEE programming requires a lot of integration efforts.

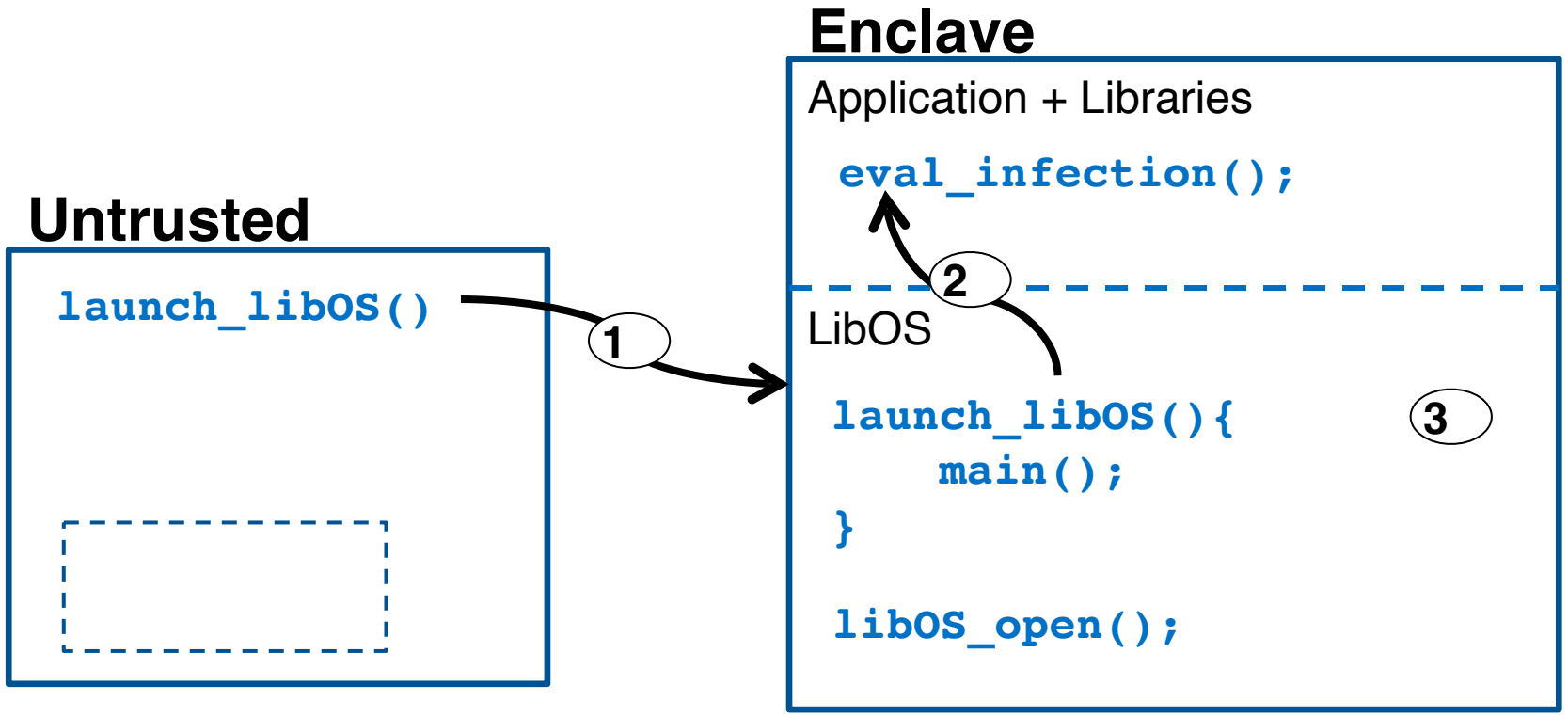
SGX in practice: Library OSes



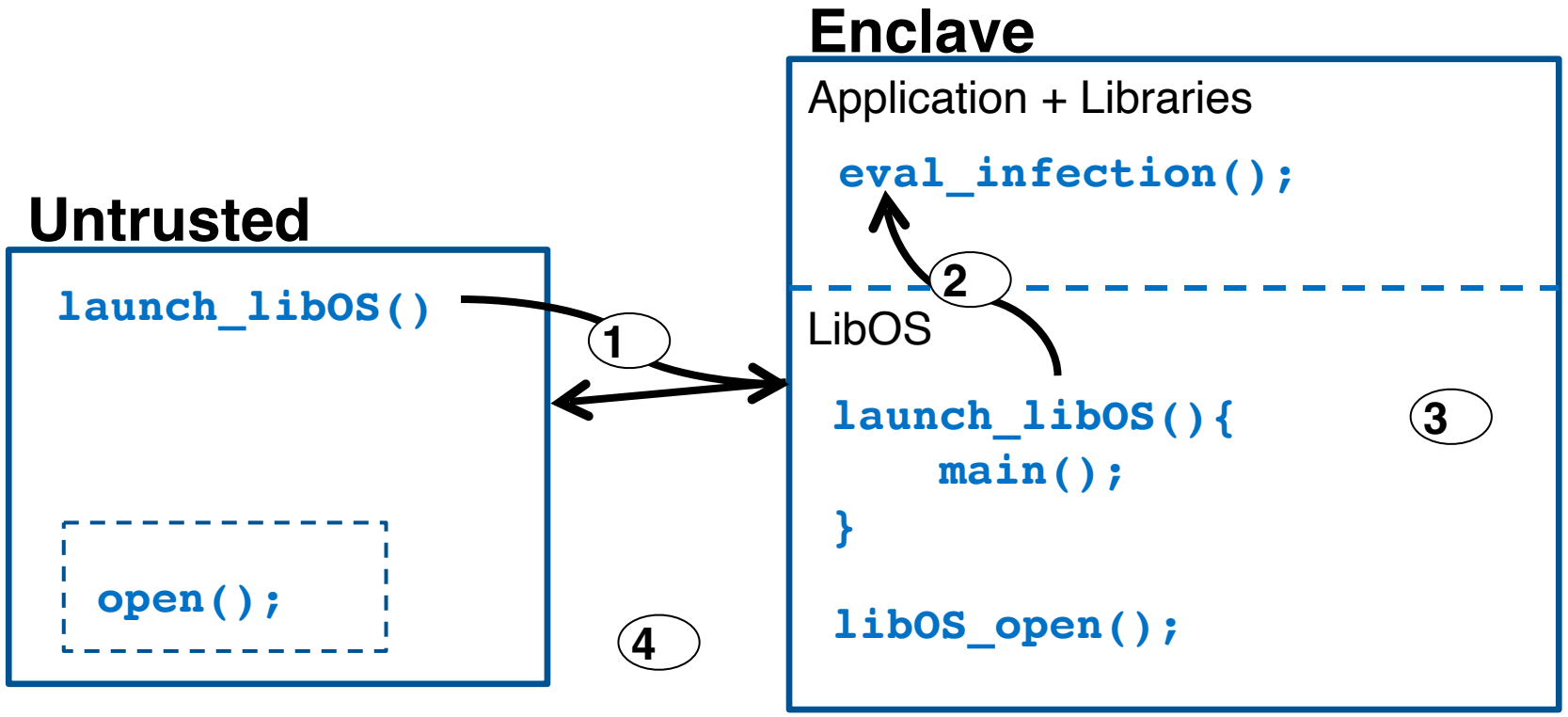
SGX in practice: Library OSes



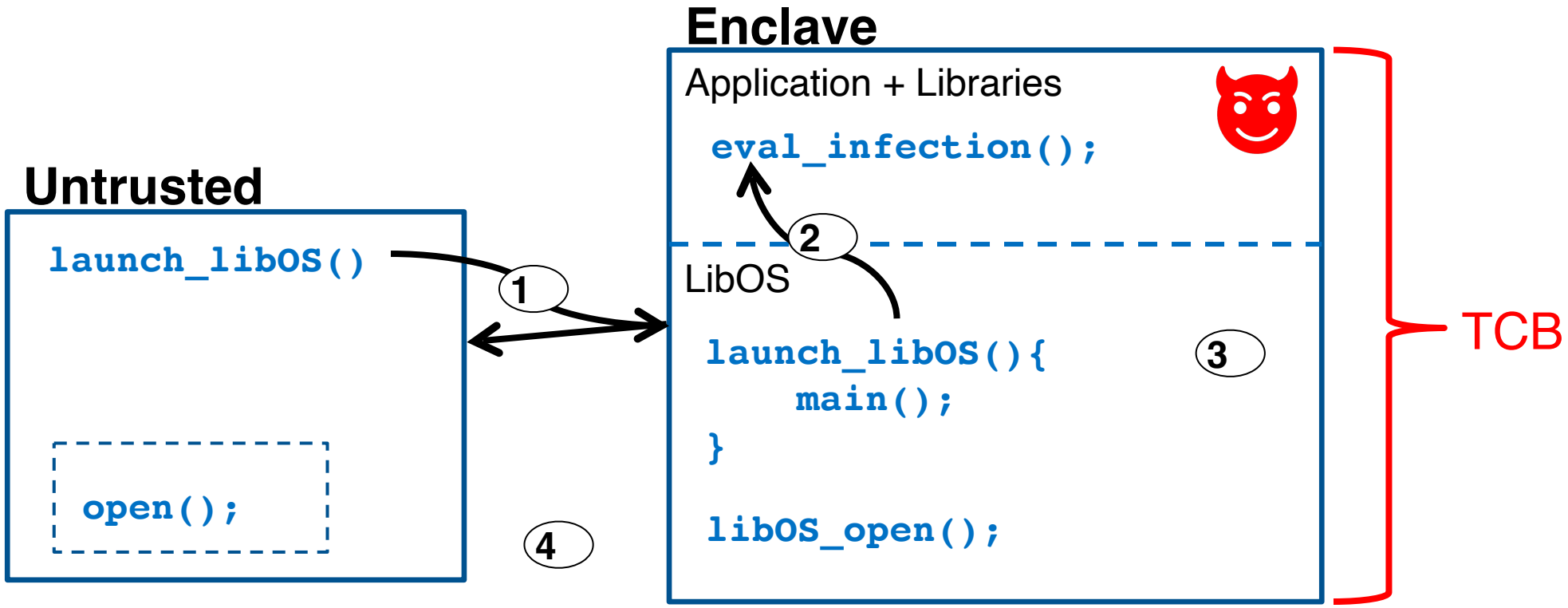
SGX in practice: Library OSes



SGX in practice: Library OSes



Library OSes bring untrusted Libs into TCB!



Attack Vector: LibOS data structures

```
eval_infection(){




}
```

LibOS file descriptor table

FD	Path
0	stdin
1	stdout
2	stderr

Attack vector: LibOS data structures

```
eval_infection(){
    ML_lib.load_training_set();



}
```

LibOS file descriptor table

FD	Path
0	stdin
1	stdout
2	stderr
3	model.csv

Attack vectors: LibOS data structures

```
eval_infection{
    ML_lib.load_training_set();

    data_file = open("x-ray.jpg");


}
```

LibOS file descriptor table

FD	Path
0	stdin
1	stdout
2	stderr
3	model.csv
4	x-ray.jpg

Attack vector: LibOS data structures

```
eval_infection(){
    ML_lib.load_training_set();

    data_file = open("x-ray.jpg");

    result = ML_lib.classify(data_file);
}
```

LibOS file descriptor table

FD	Path
0	stdin
1	stdout
2	stderr
3	model.csv
4	x-ray.jpg

Attack vector: LibOS data structures

```
eval_infection(){
    ML_lib.load_training_set();

    data_file = open("x-ray.jpg");

    result = ML_lib.classify(data_file);
}
```

LibOS file descriptor table

FD	Path
0	stdin
1	stdout
2	stderr
3	model.csv
4	healthy.jpg

Attack vector: LibOS data structures

```
eval_infection(){
    ML_lib.load_training_set();

    data_file = open("x-ray.jpg");

    result = ML_lib.classify(data_file);
}
```

LibOS file descriptor table

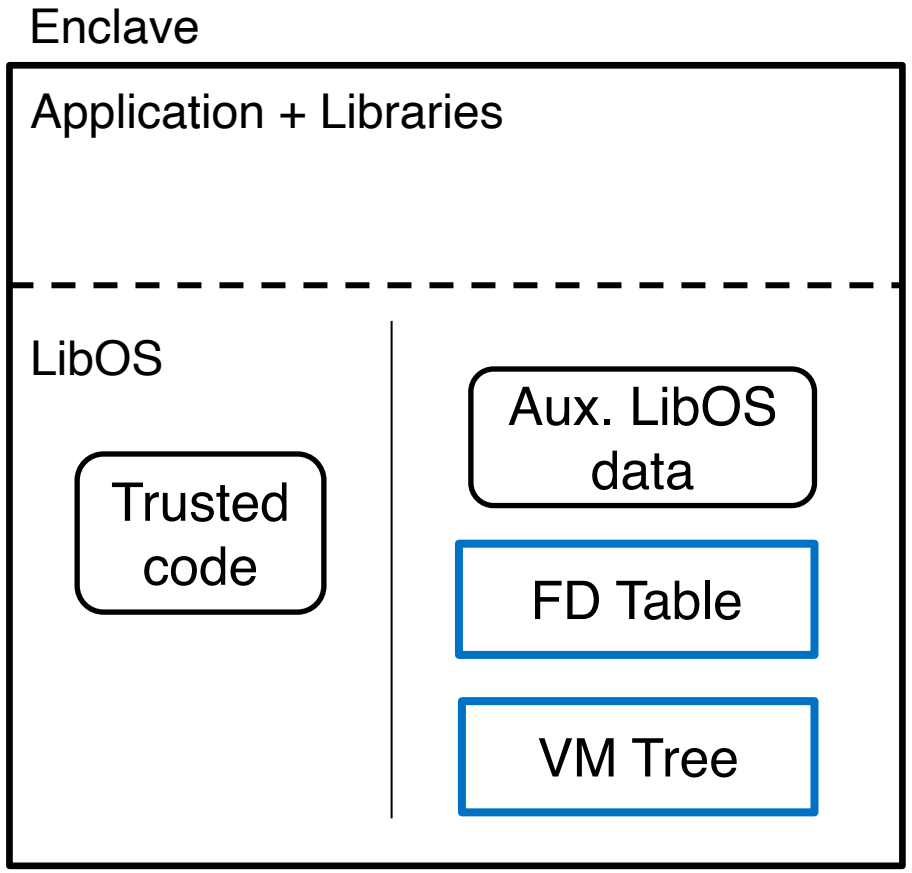
FD	Path
0	stdin
1	stdout
2	stderr
3	model.csv
4	healthy.jpg

Traditional OS: Prevented via privilege ring isolation!

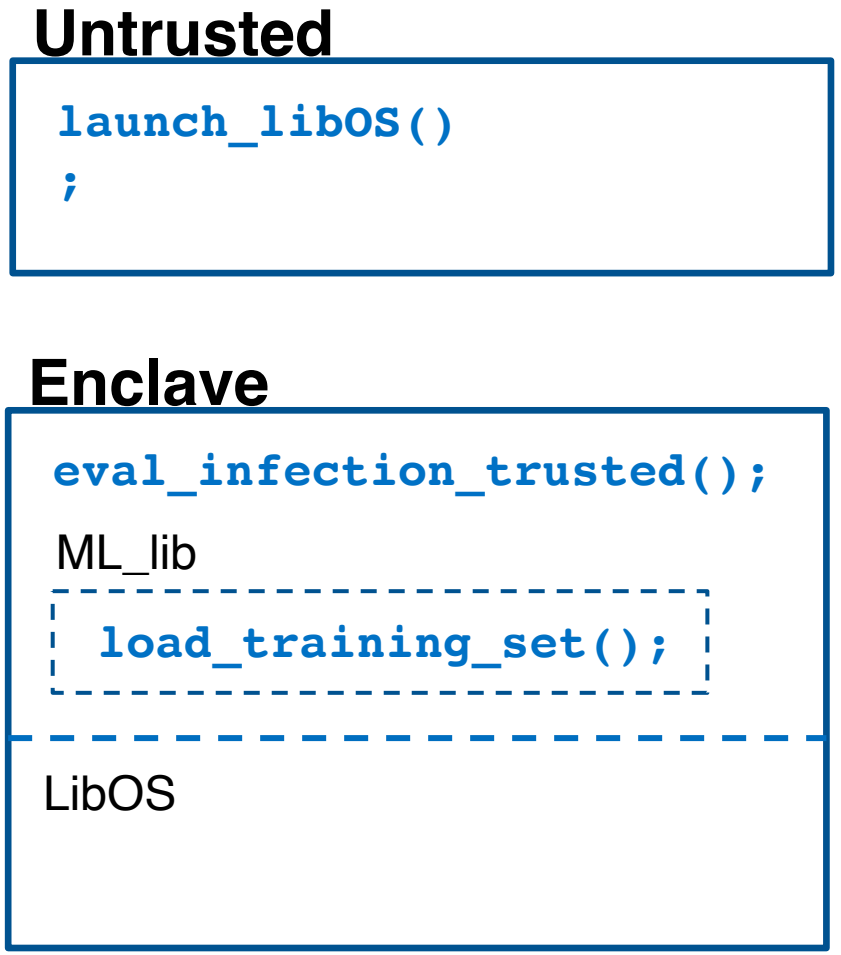
Problem:

No memory access restrictions *within* an enclave. Untrusted third-party code has unfettered access to all in-enclave data.

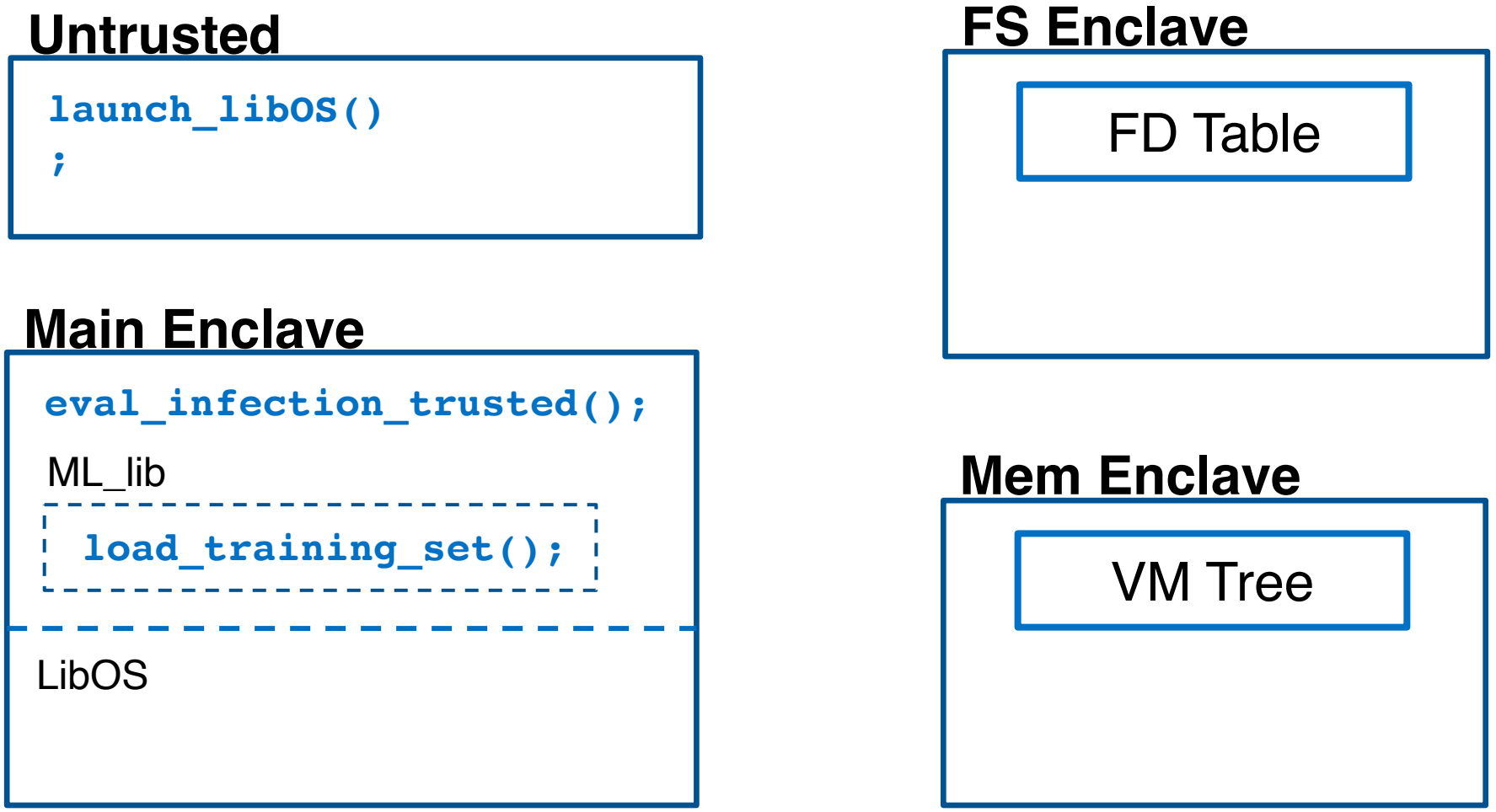
EnclaveDom: Privilege Separation in HW Enclaves



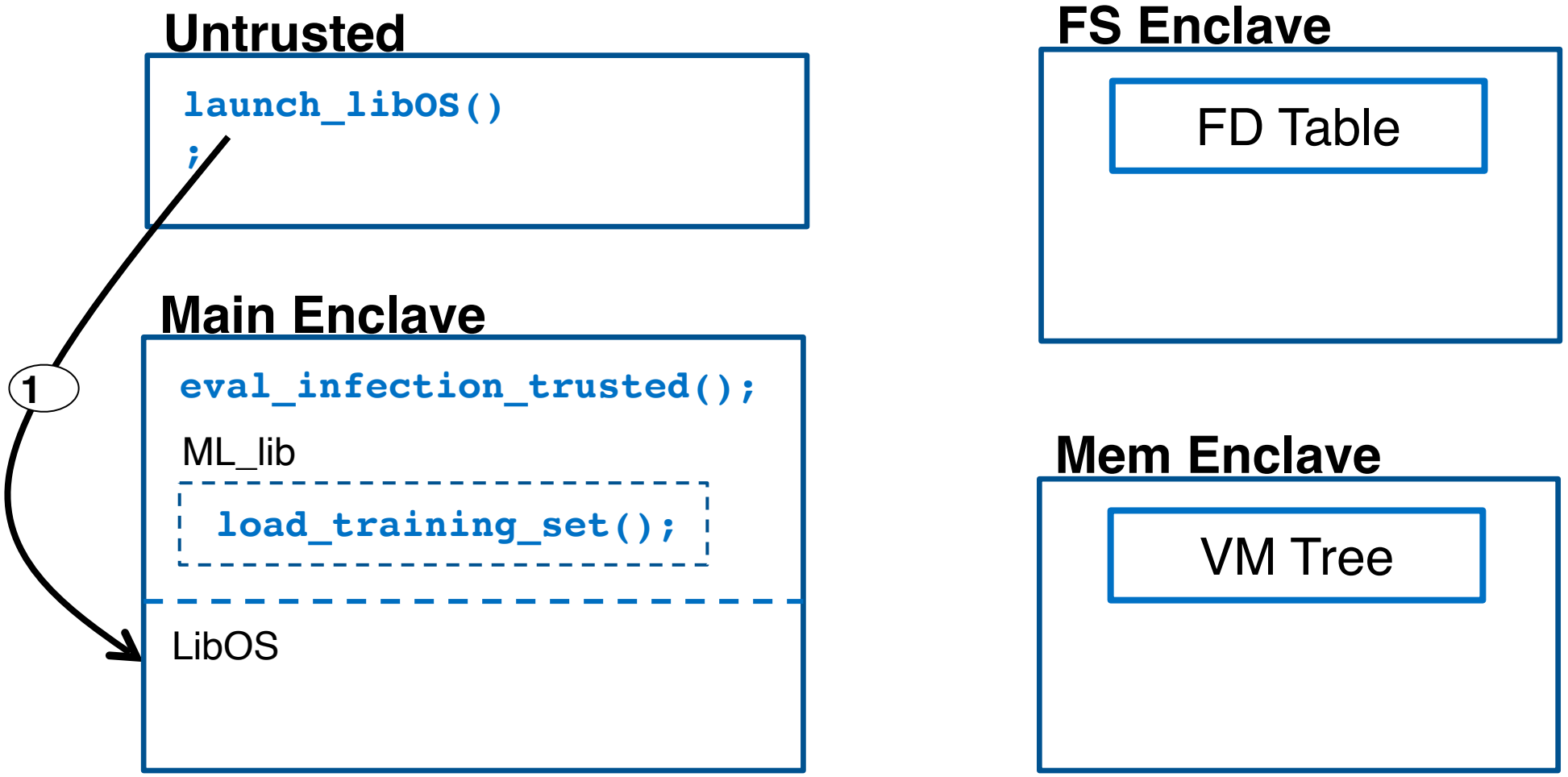
Strawman: Per-Domain Enclaves



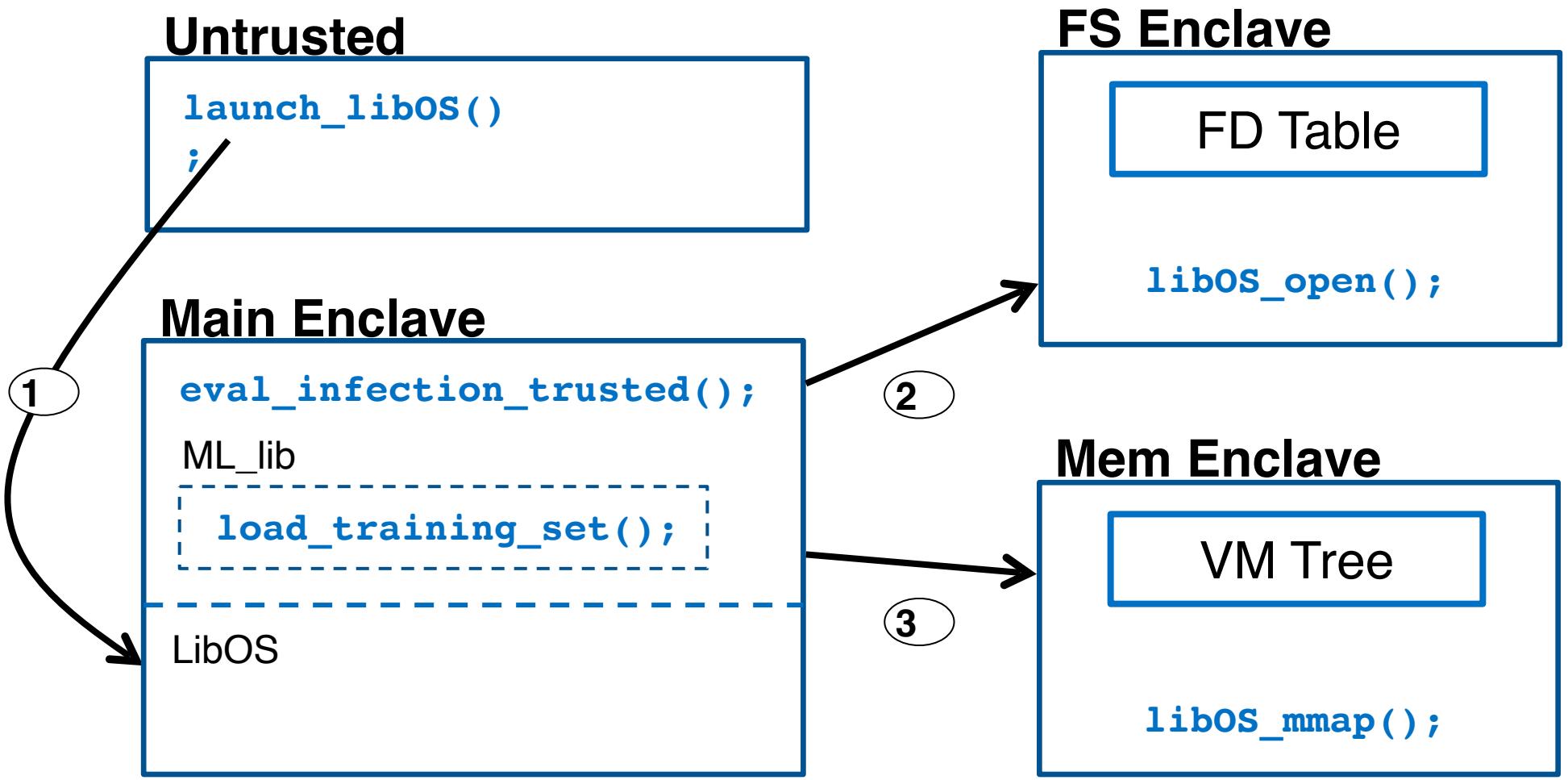
Strawman: Per-Domain Enclaves



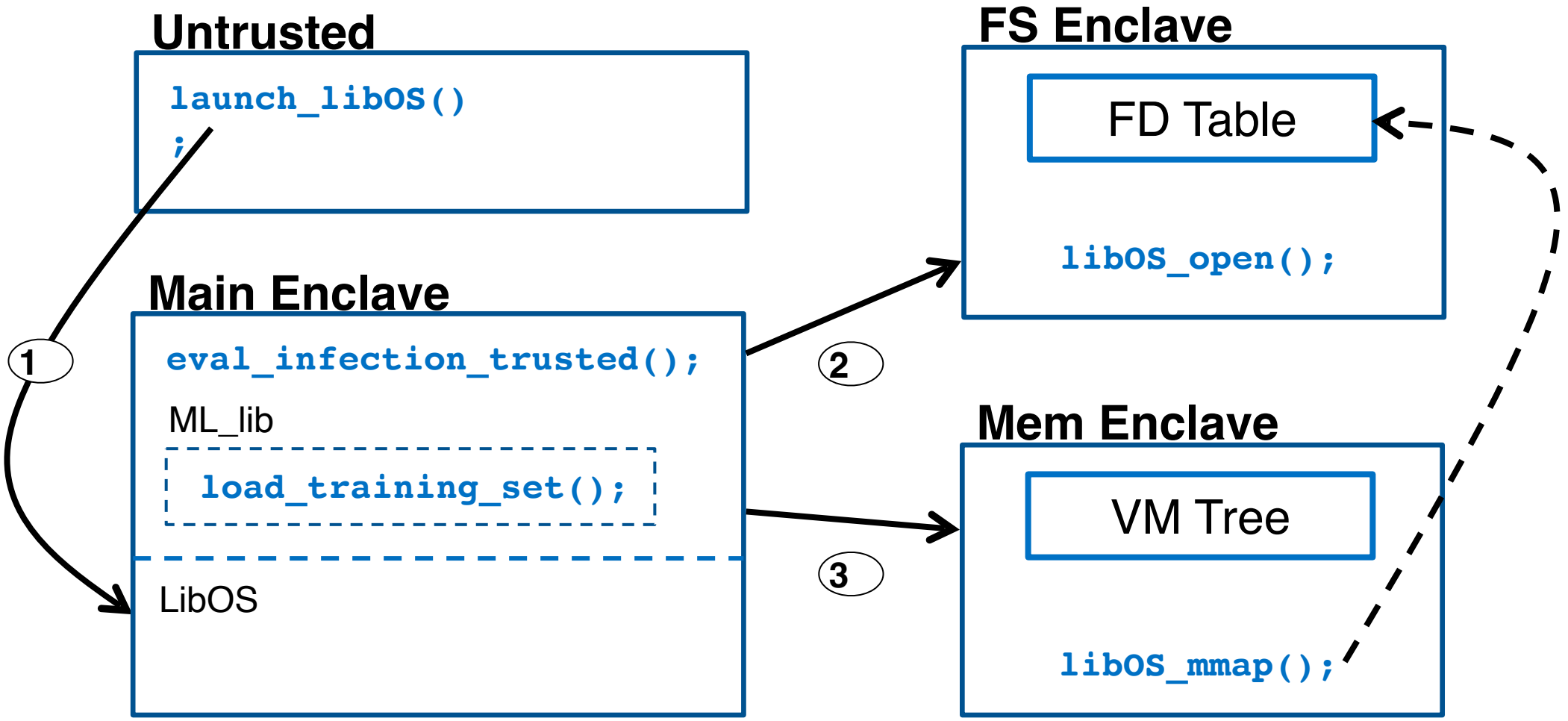
Strawman: Per-Domain Enclaves



Strawman: Per-Domain Enclaves

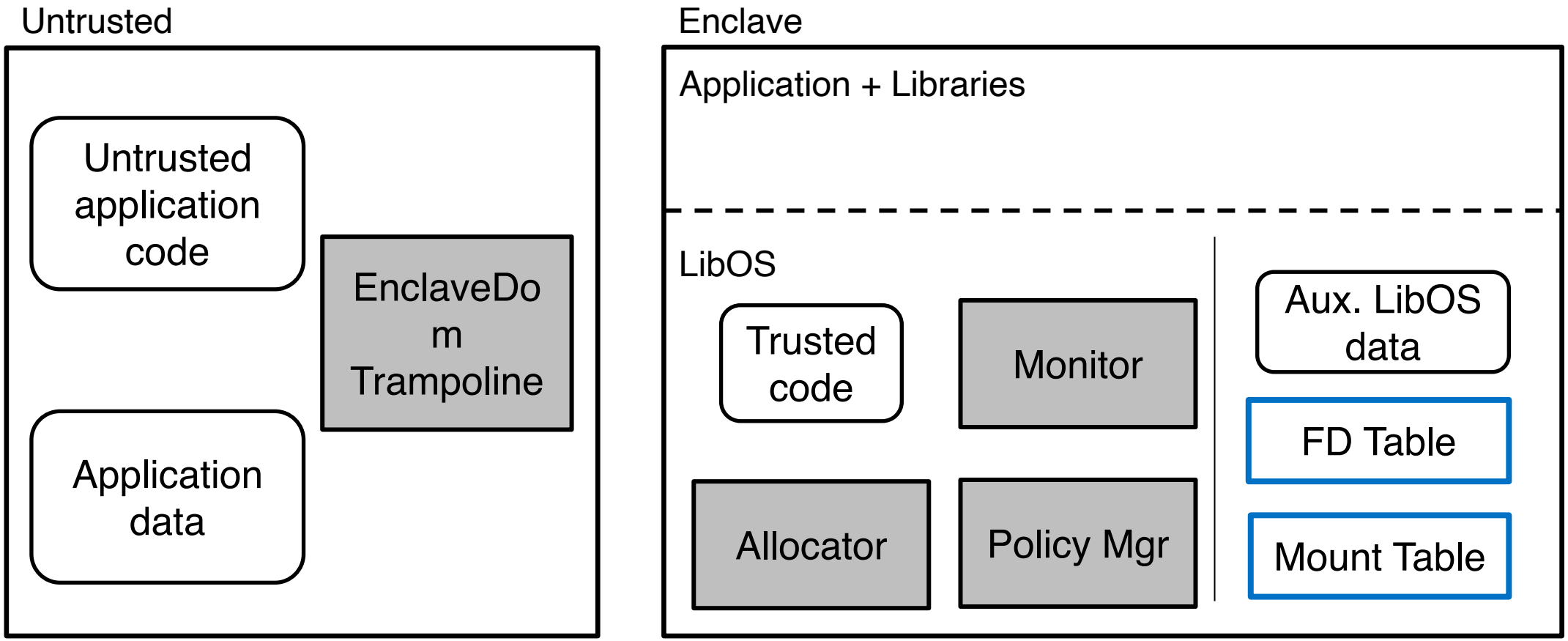


Strawman: Per-Domain Enclaves



Isolation
vs
Complexity & Performance

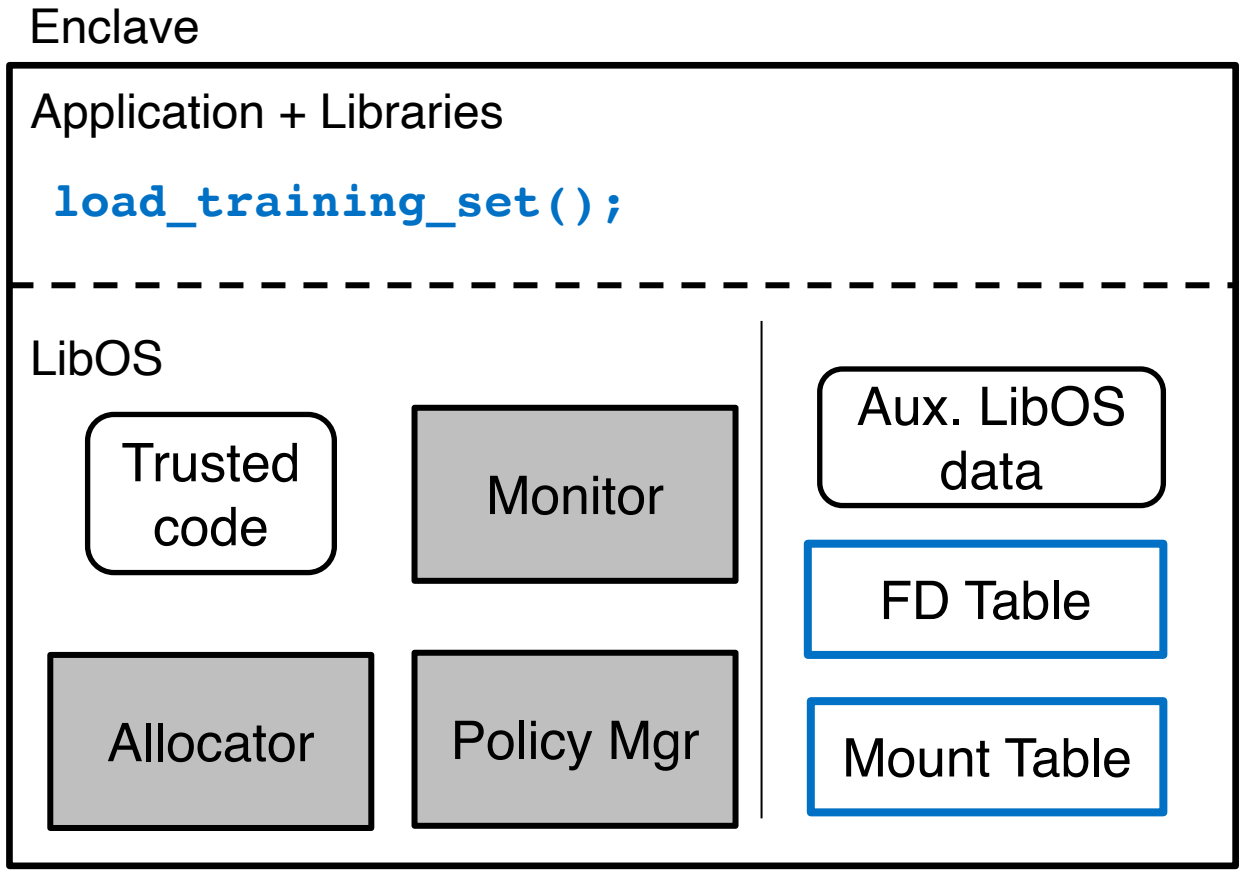
EnclaveDom LibOS Architecture



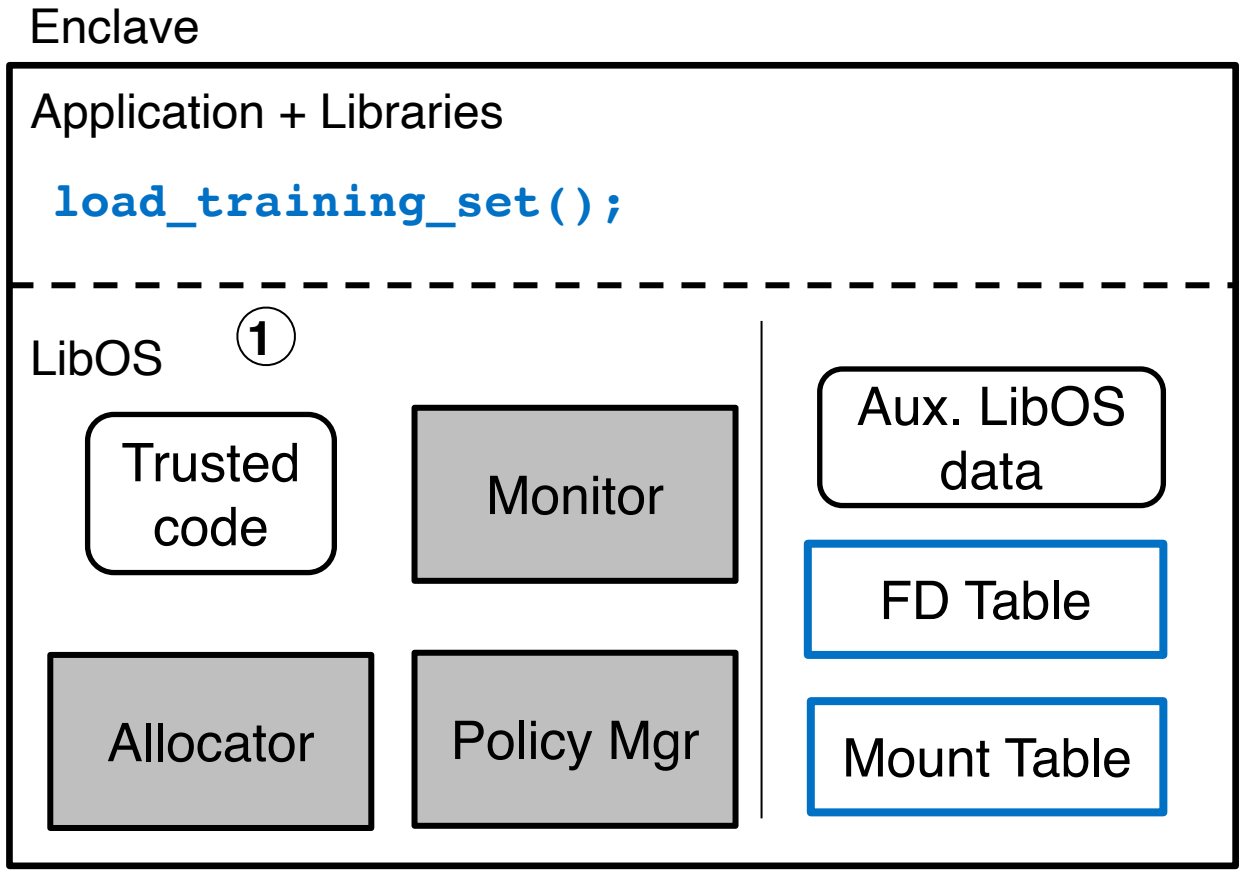
EnclaveDom Memory Domains: Intel MPK

- Memory protection keys: hardware memory page tagging
- Per-CPU register stores R/W access privileges to 16 tags
- Domain access bits in register check on every memory access

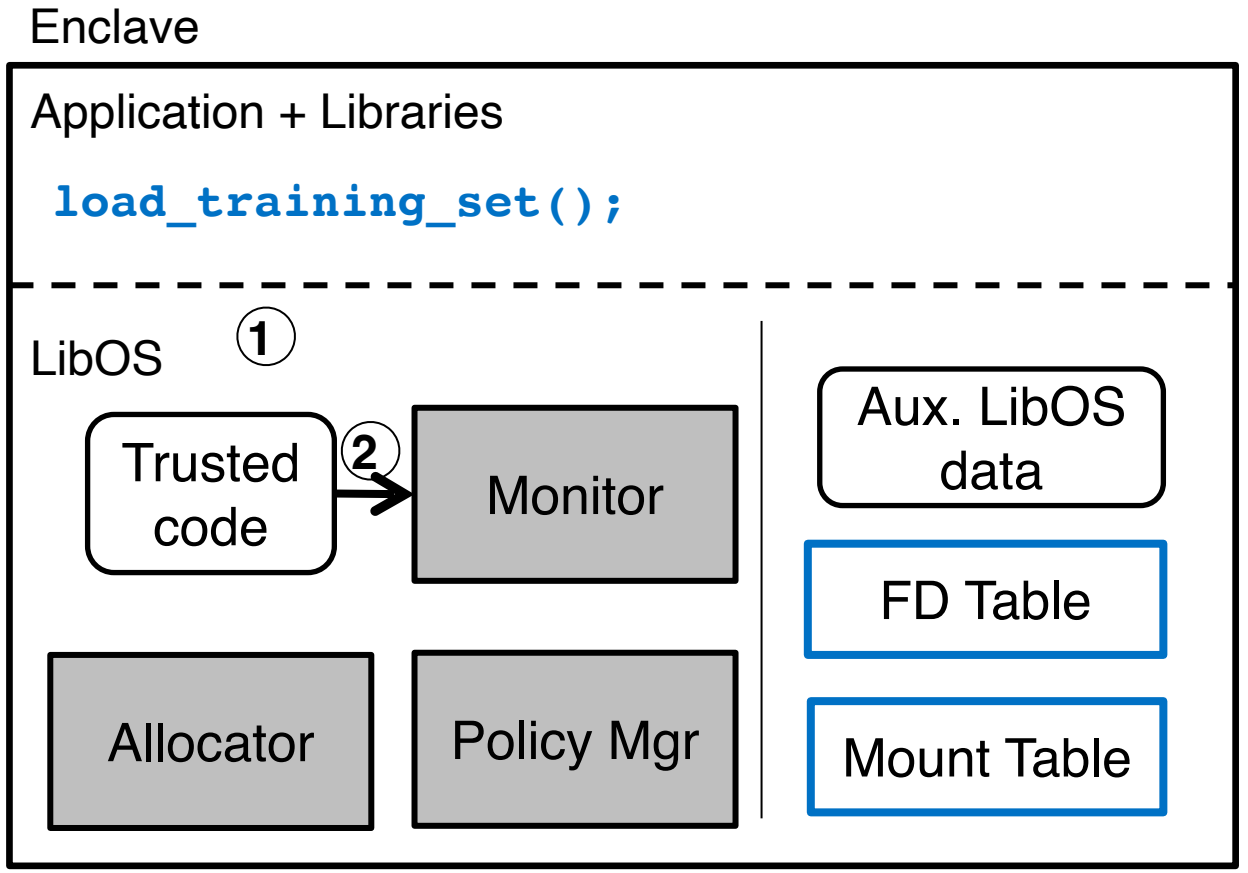
EnclaveDom Access Checks



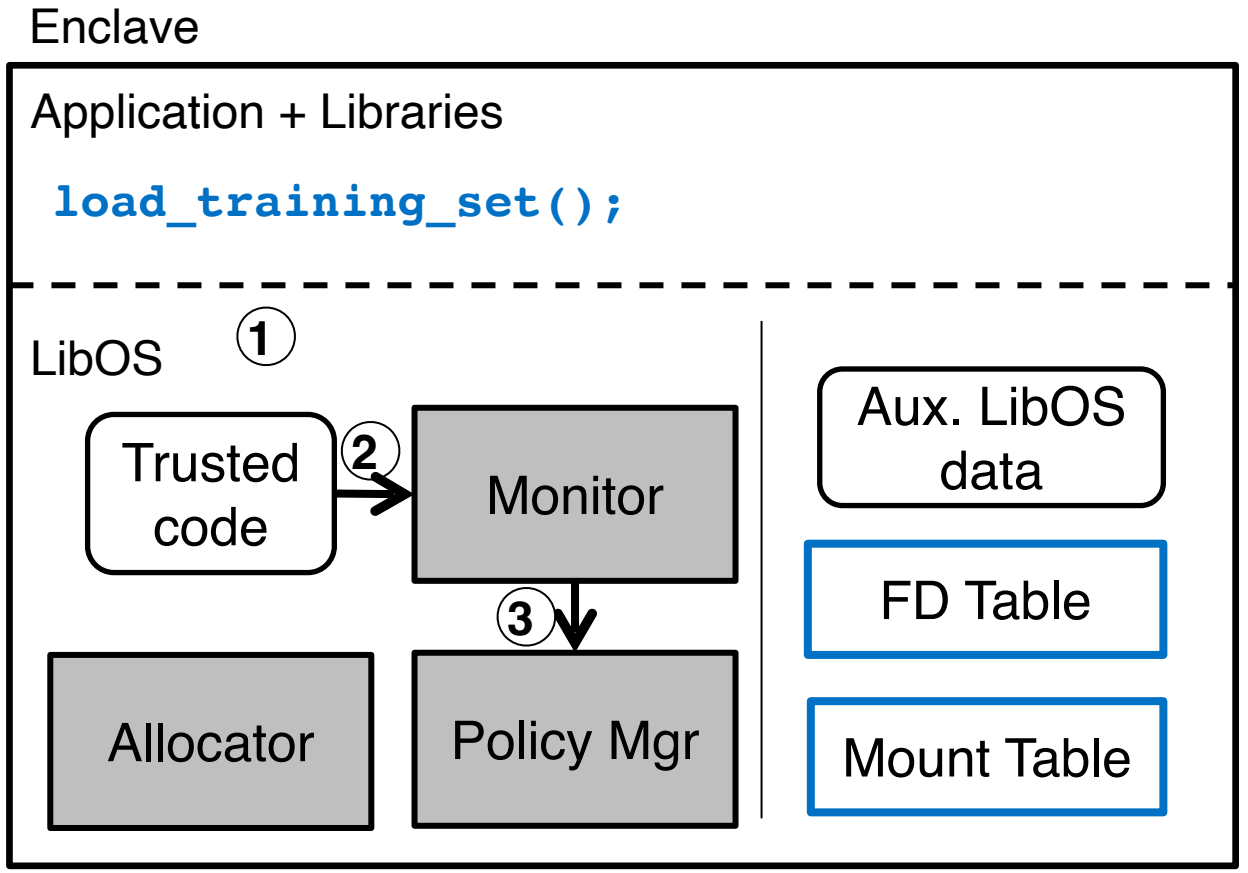
EnclaveDom Access Checks



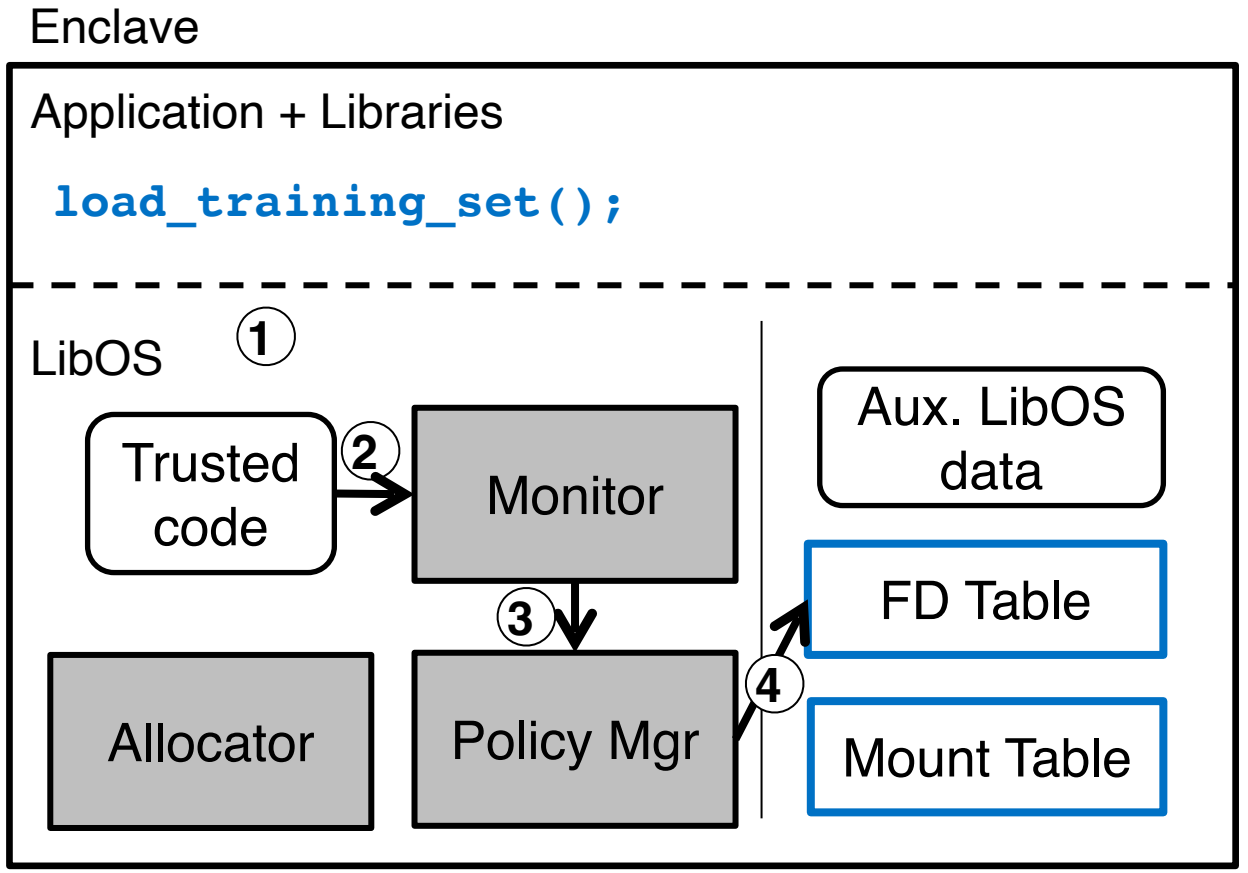
EnclaveDom Access Checks



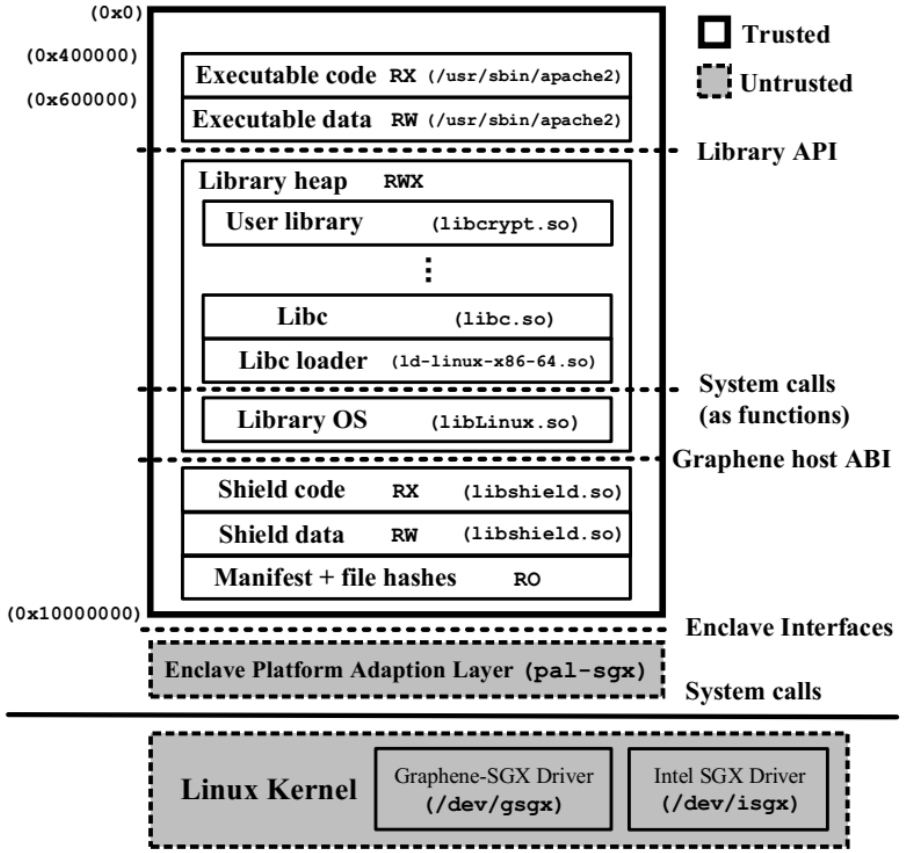
EnclaveDom Access Checks



EnclaveDom Access Checks

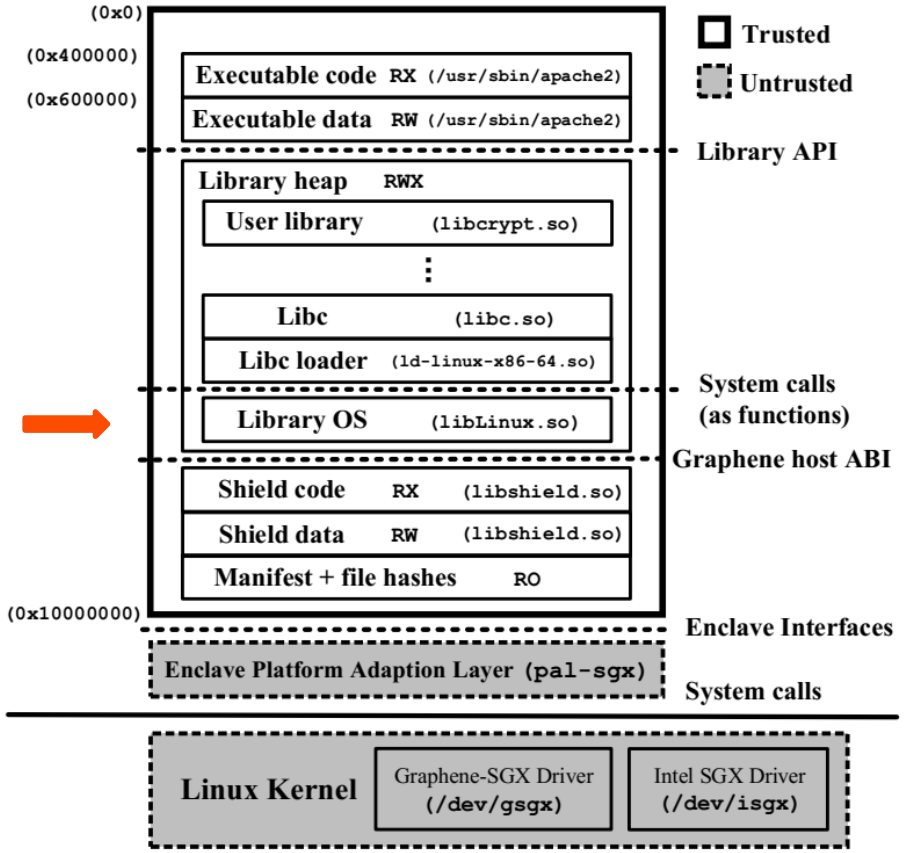


Porting EnclaveDom to Graphene



Che Tsai, C., Porter, D. E., and Vij, M. Graphene-SGX: A Practical Library OS for Unmodified Applications on SGX. In *USENIX ATC, 2017*.

Porting EnclaveDom to Graphene



Che Tsai, C., Porter, D. E., and Vij, M. Graphene-SGX: A Practical Library OS for Unmodified Applications on SGX. In *USENIX ATC, 2017*.

EnclaveDom Testing Platform

- Intel 10th generation CPU
- Linux kernel 5.3 (Ubuntu 18.04)
- Userspace EnclaveDom API for domain management in C++
- Graphene-SGX March 2019 pre-release

EnclaveDom Performance Evaluation

	% time in EnclaveDom	accessed memdom(s)
`open`	6.4	handle, fs
`close`	49.1	handle
`stat`	49.9	fs
`fstat`	50.1	handle, fs
`mmap`	0.8	handle

Graphene shim layer microbenchmarks

System configuration: Intel 10th gen CPU, Linux kernel 5.3 (Ubuntu 18.04), userspace EnclaveDom API in C++, Graphene SGX March 2019 pre-release

EnclaveDom Memory Usage

	memory usage (in bytes)
handle_dom	98
fs_dom	1030
Total	1200

Graphene shim layer microbenchmarks

System configuration: Intel 10th gen CPU, Linux kernel 5.3 (Ubuntu 18.04), userspace EnclaveDom API in C++, Graphene SGX March 2019 pre-release

EnclaveDom Open Research Questions

- How can EnclaveDom provide fine-grained isolation within the app itself?

EnclaveDom Open Research Questions

- How can EnclaveDom provide fine-grained isolation within the app itself?
- How to protect OS-level MPK interface against misuse?

EnclaveDom Open Research Questions

- How to provide fine-grained isolation within the app itself?
- How to protect MPK interface against compromised host OS?
- Scalability of MPK: How to support more than 16 domains?

EnclaveDom Recap

Target Large-TCB Apps

SGX + MPK

Privilege Isolation for LibOSes

Thank you!

Contact: marcela.melara@intel.com

Twitter: @mas0mel

Github: @masomel