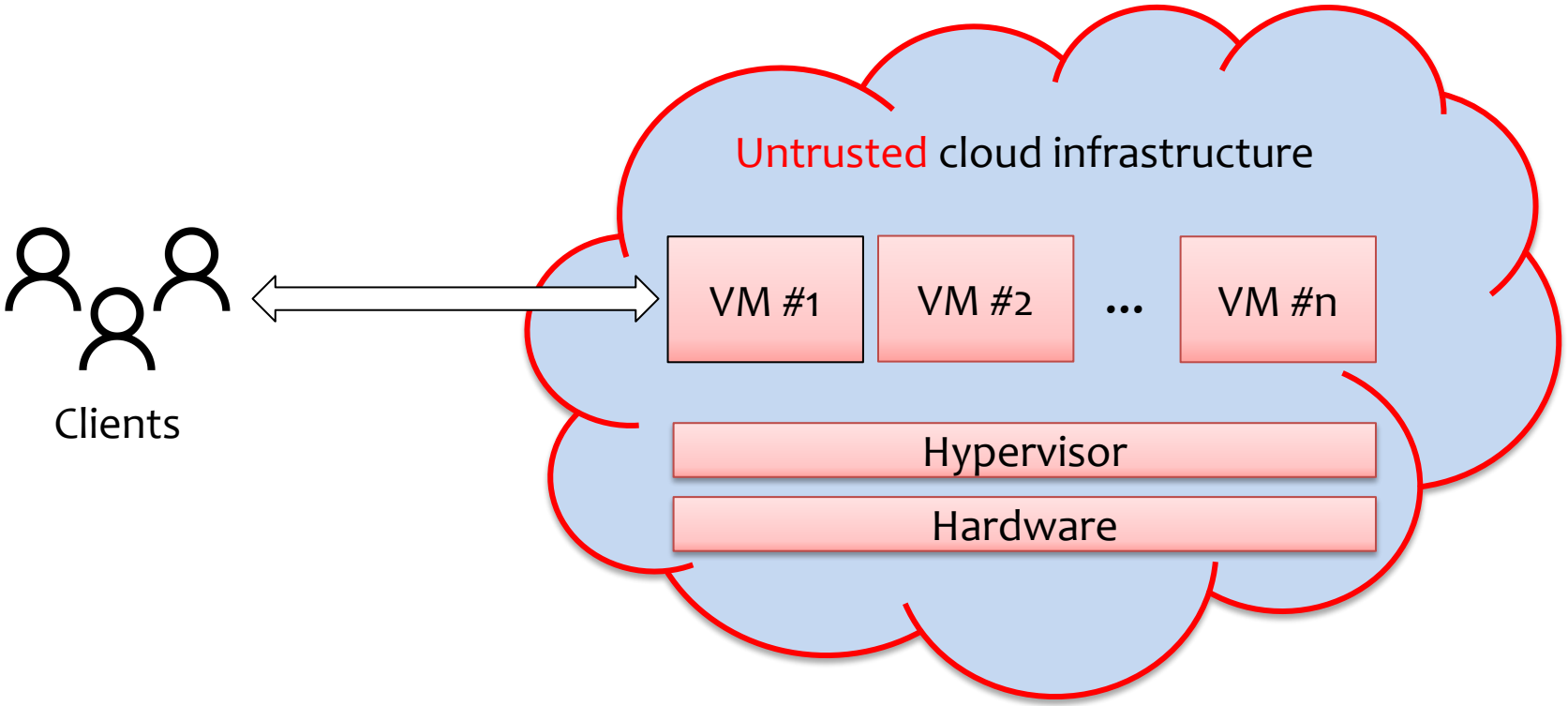
Gramine-TDX

A Lightweight OS Kernel for Confidential VMs

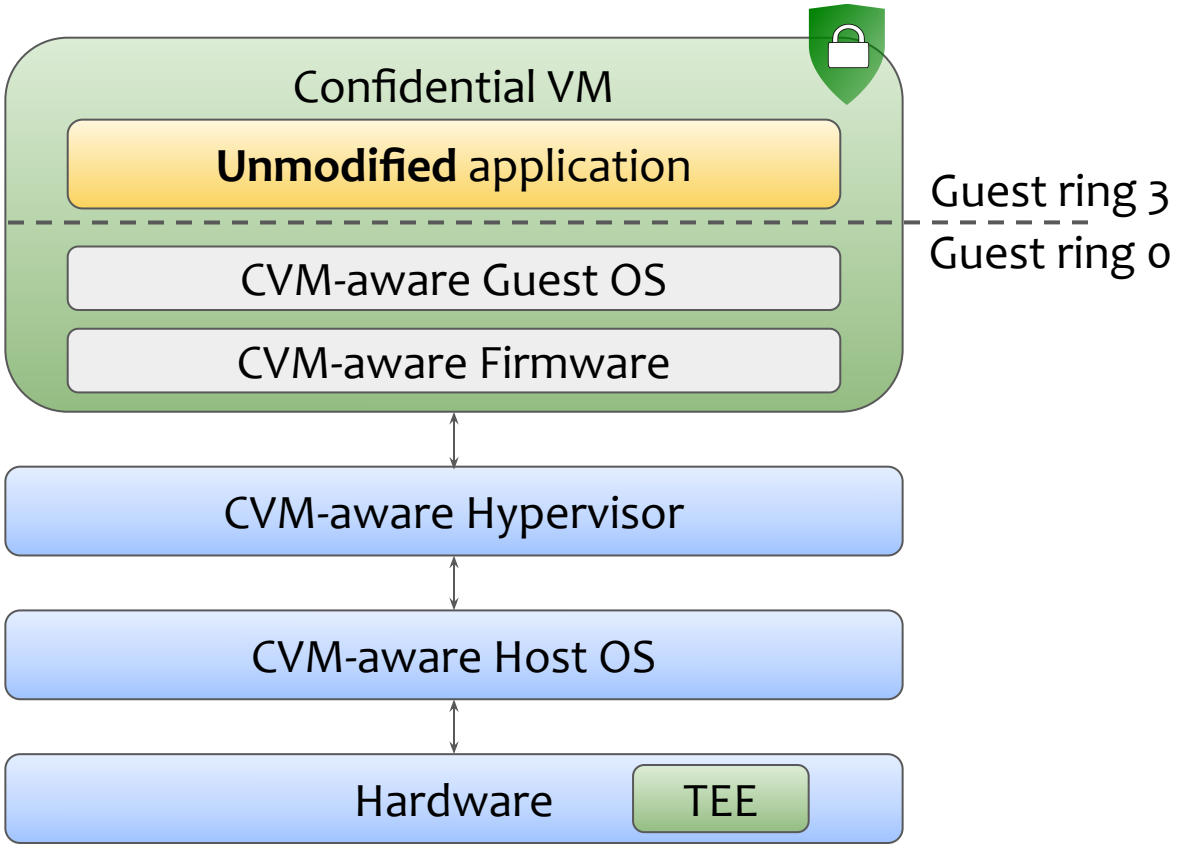
Dmitrii Kuvaiskii, **Dimitrios Stavrakakis**, Kailun Qin,
Cedric Xing, Pramod Bhatotia, Mona Vij

ACM CCS 2024

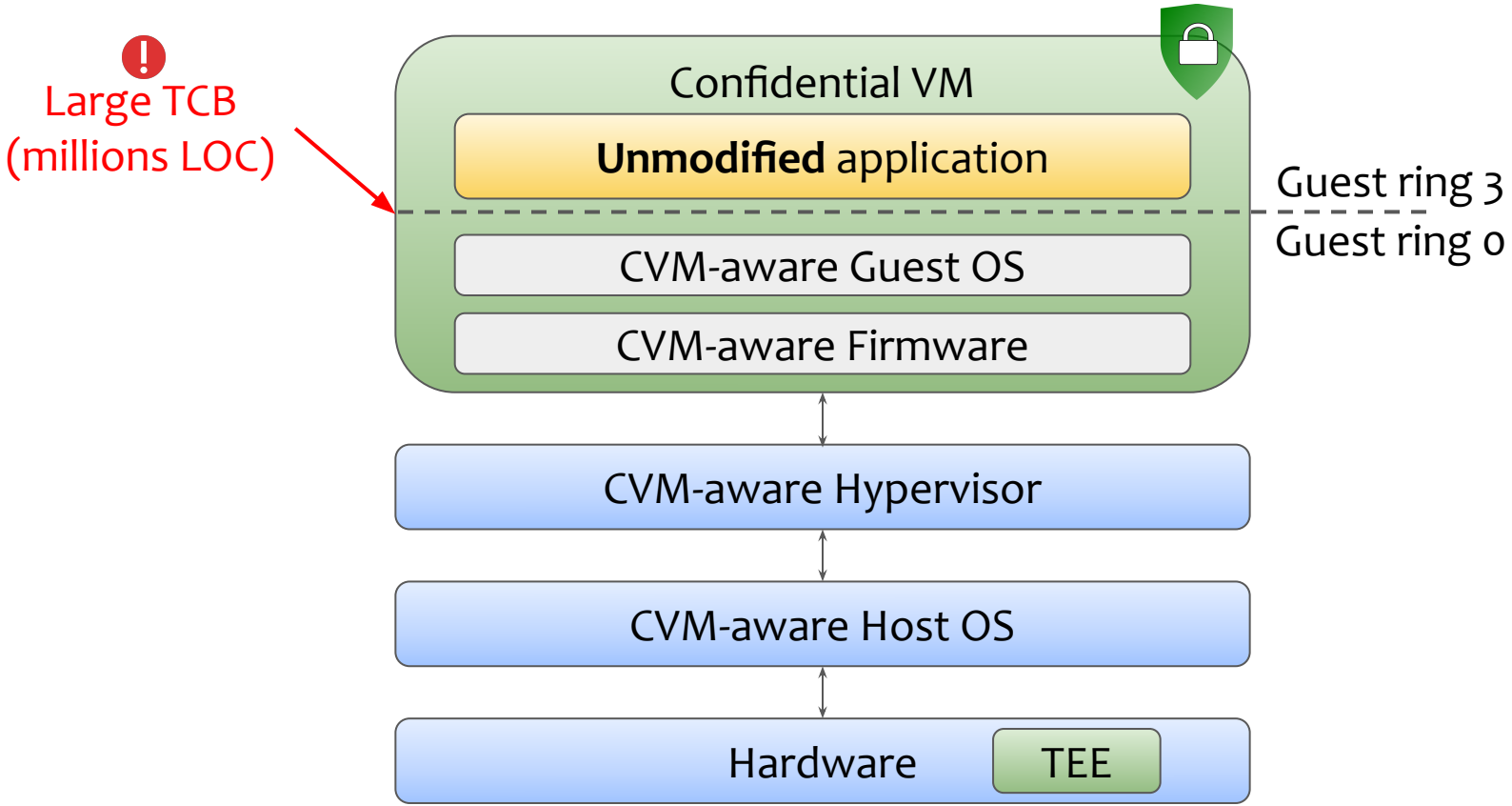
Security threats in the cloud



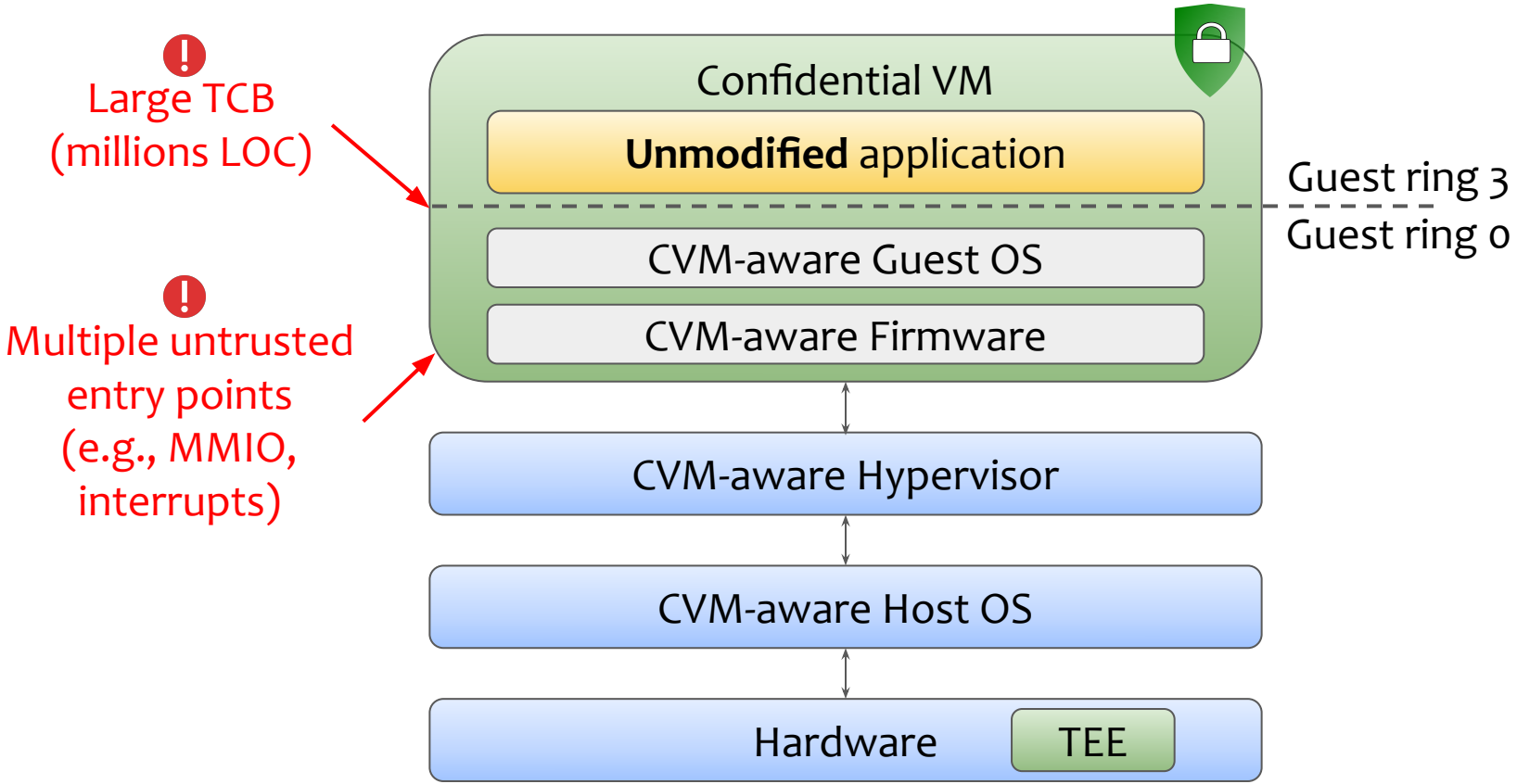
Confidential Virtual Machines (CVMs)



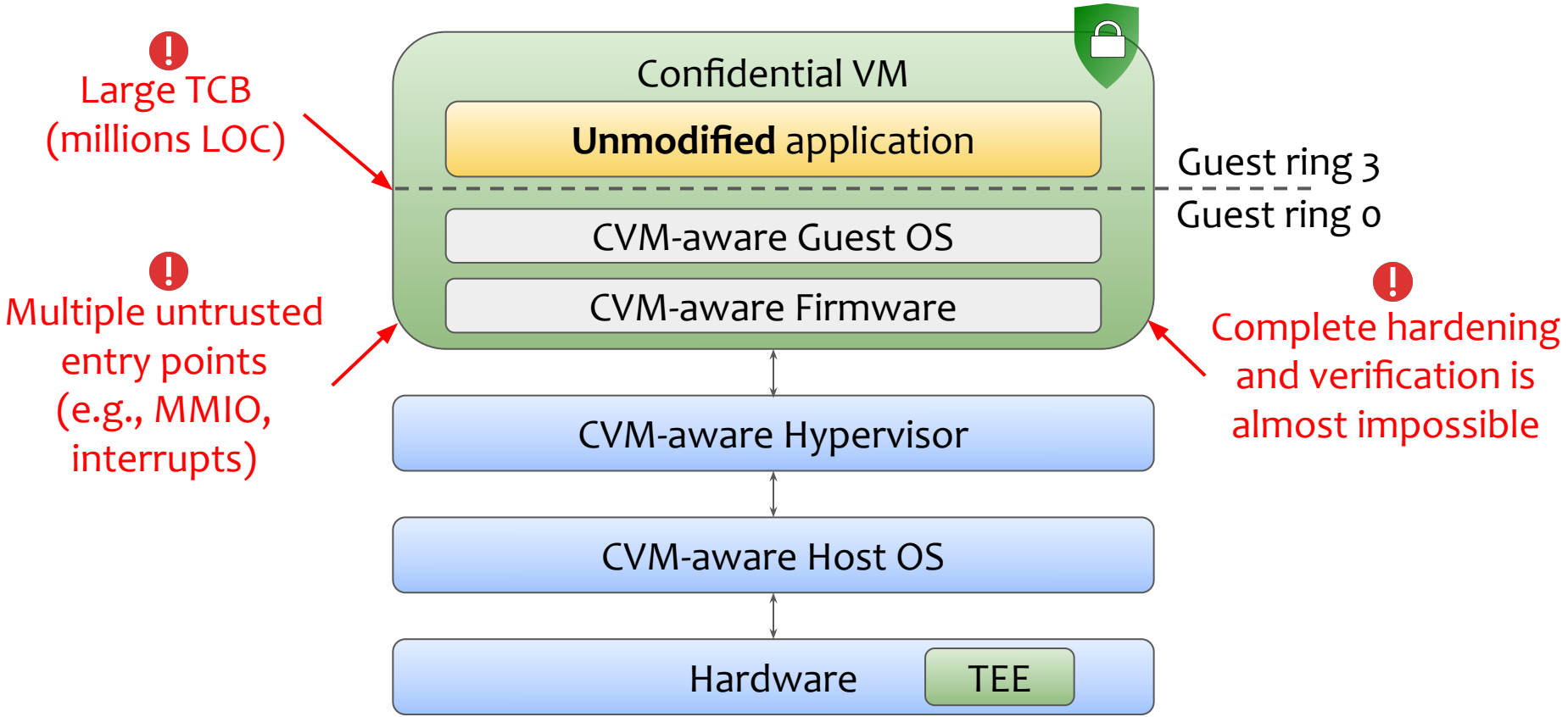
Confidential Virtual Machines (CVMs)



Confidential Virtual Machines (CVMs)



Confidential Virtual Machines (CVMs)



Problem statement

How to design a **generic, minimal, security-first** kernel for confidential VMs with a **small attack surface**?

Gramine-TDX

A Lightweight OS Kernel for Confidential VMs

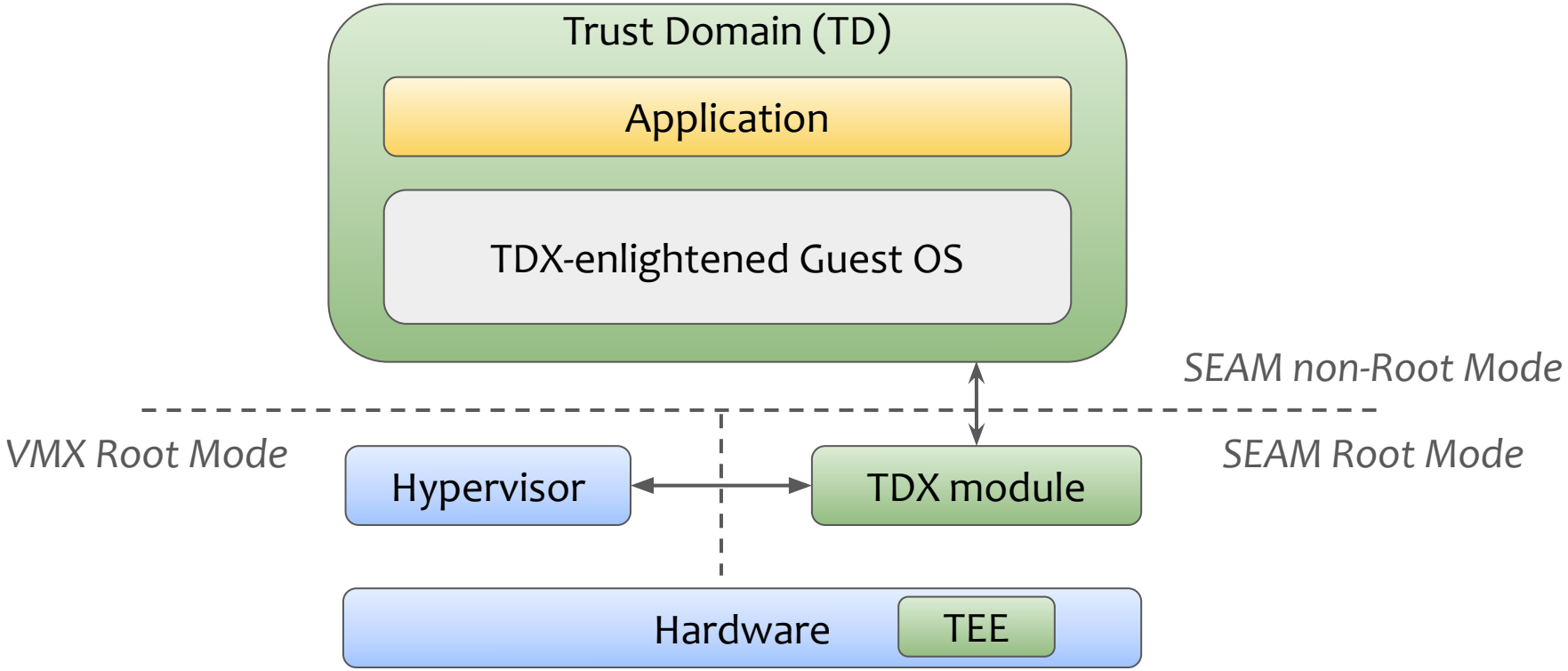
Design goals:

- Minimal attack surface
- Small Trusted Computing Base (TCB)
- Support of diverse applications, frameworks and languages
- Ease of use and deployment

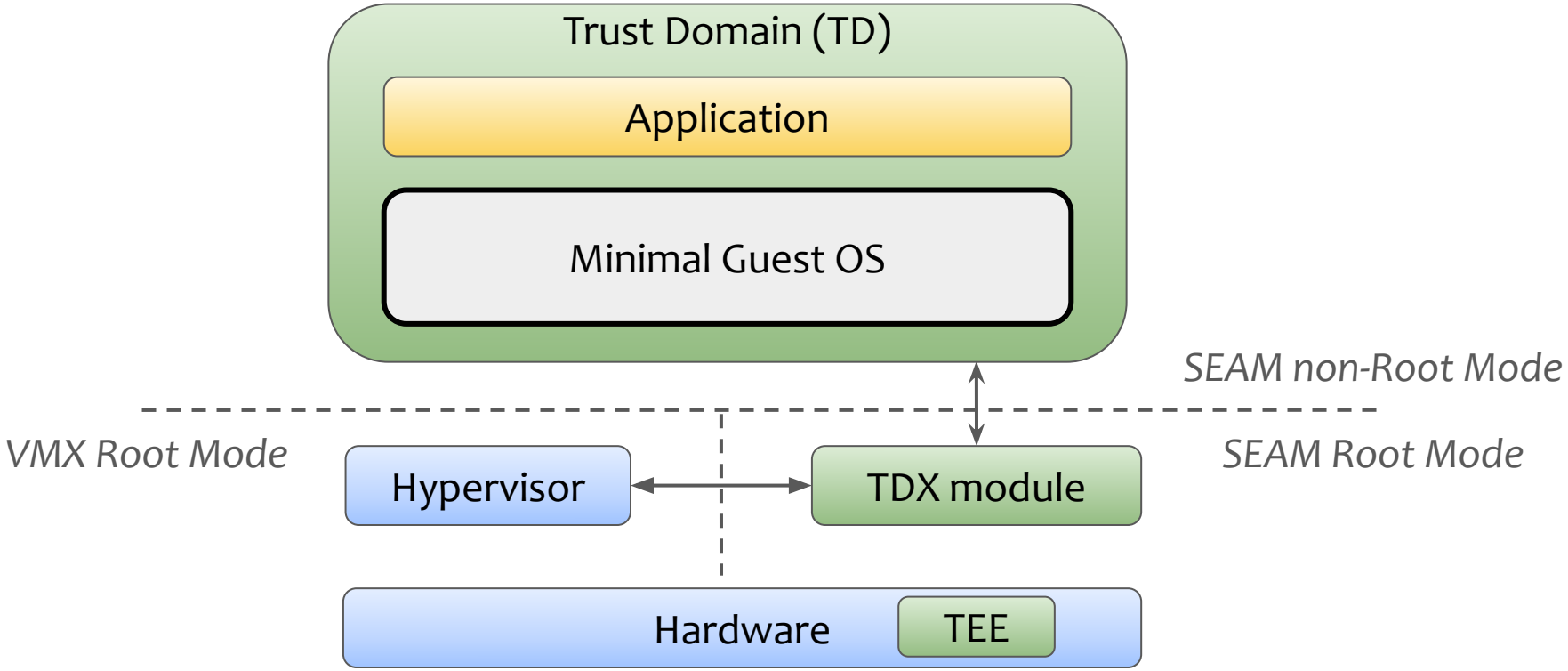
Outline

- ~~Motivation~~
- Design
- Implementation
- Evaluation

Design – Intel TDX

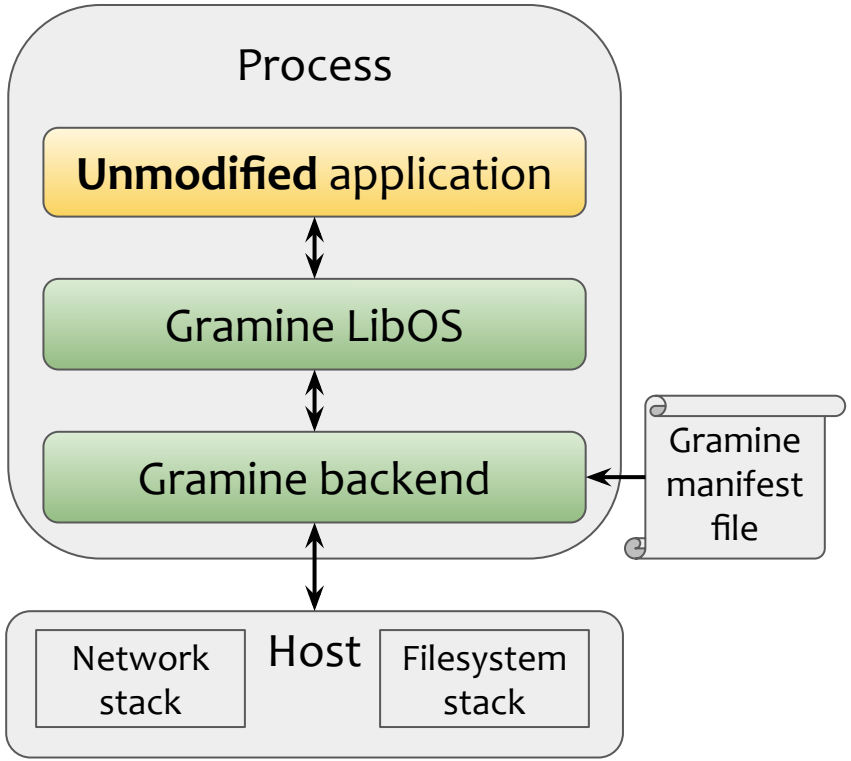


Design – Strawman design



Design – Gramine LibOS

- Mature confidential computing project
- Active development for the last ~10 years
- Modular design
- Offloads functionalities to the host
- Manifest file for security configuration



Design – Challenges

#1 Preserve a small TCB

#2 Minimal interface & hardening

#3 Practical cloud deployment

Design – Challenges

#1 Preserve a small TCB

Simplistic implementation of OS primitives
Include **the bare minimum** virtio drivers

#2 Minimal interface & hardening

#3 Practical cloud deployment

Design – Challenges

#1 Preserve a small TCB

Simplistic implementation of OS primitives
Include **the bare minimum** virtio drivers

#2 Minimal interface & hardening

Expose **limited** entry points
Verify against known hashes/values

#3 Practical cloud deployment

Design – Challenges

#1 Preserve a small TCB

Simplistic implementation of OS primitives
Include **the bare minimum** virtio drivers

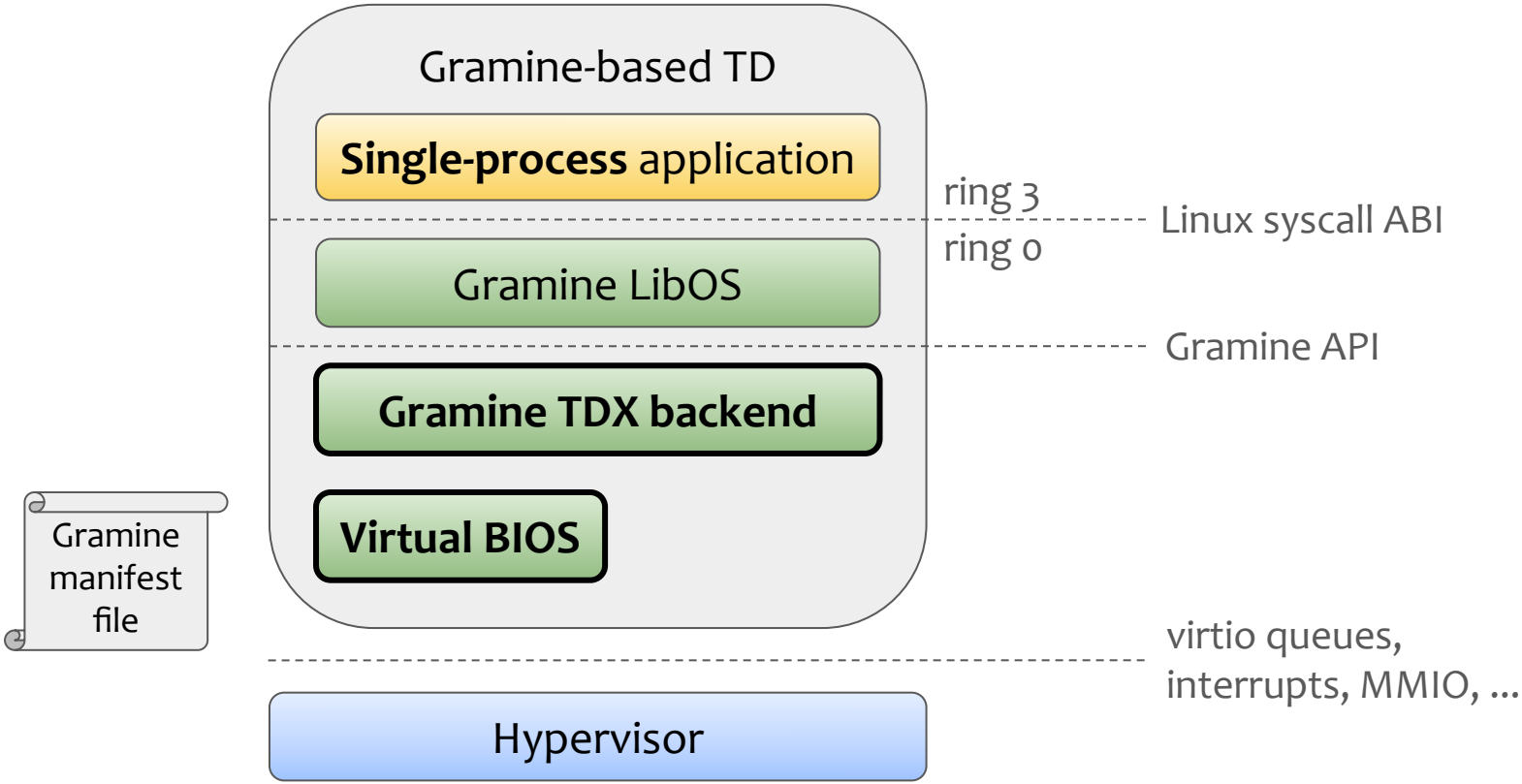
#2 Minimal interface & hardening

Expose **limited** entry points
Verify against known hashes/values

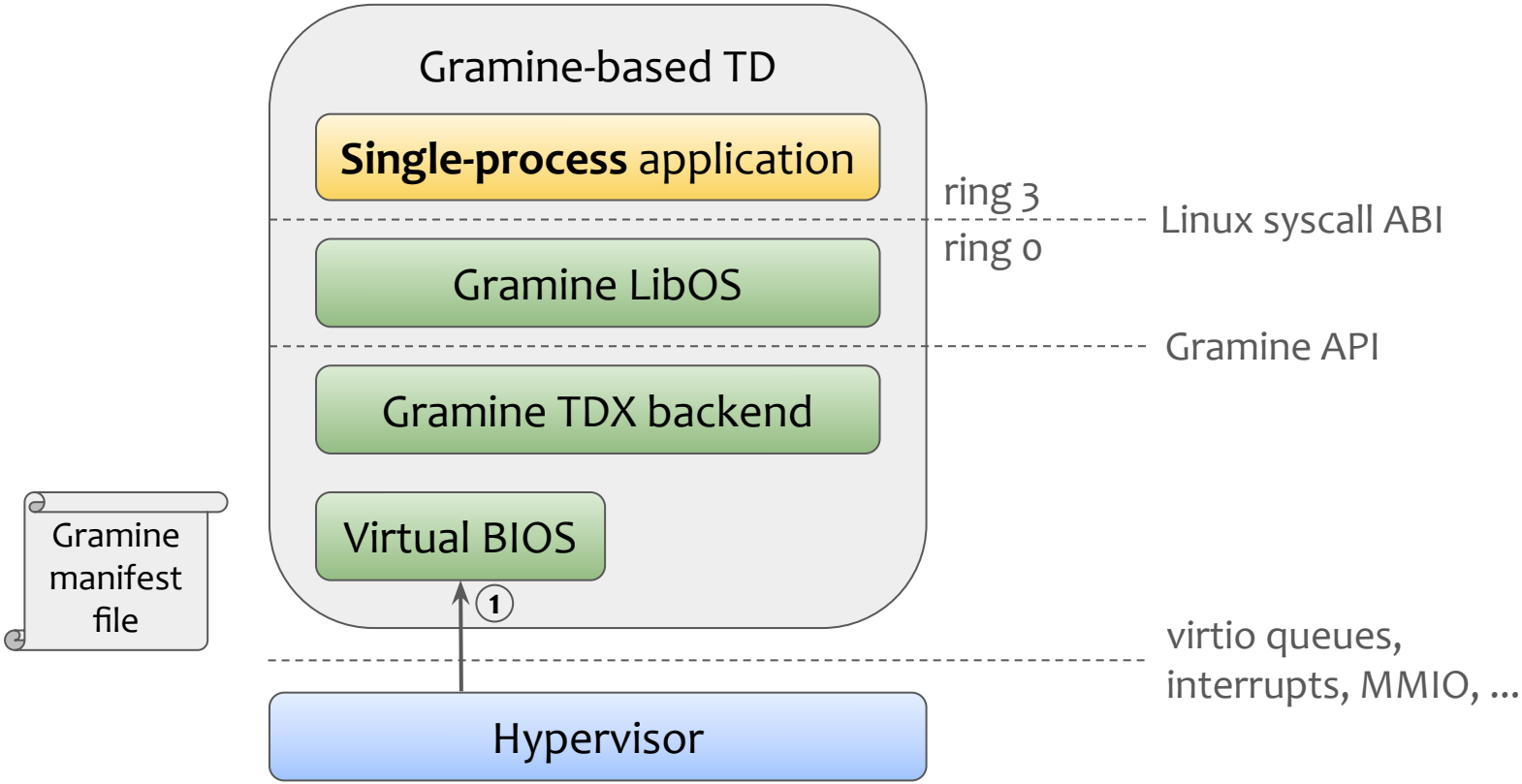
#3 Practical cloud deployment

Hypervisor-agnostic design
Standardized virtio devices and VM techniques

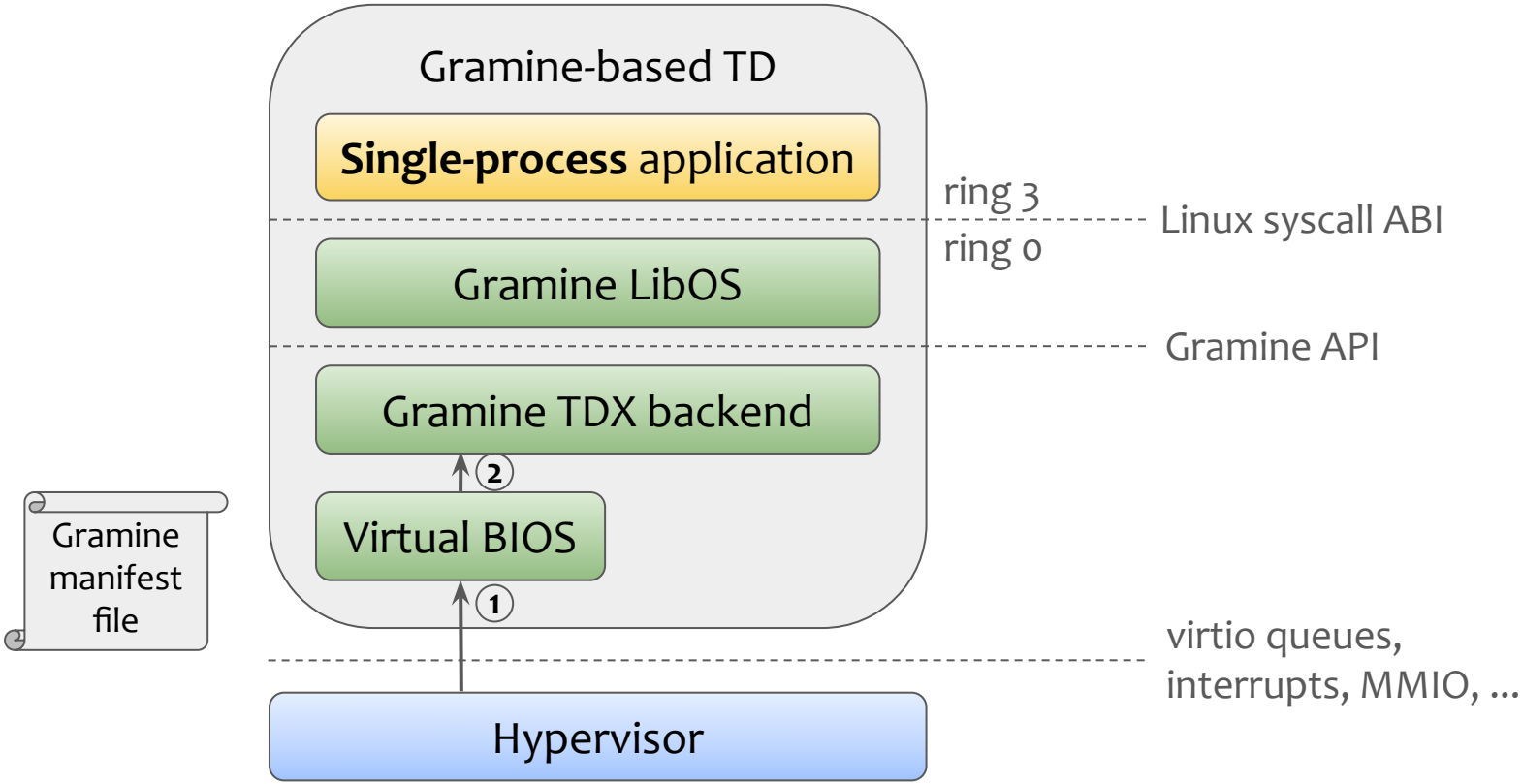
Design – Gramine-TDX overview



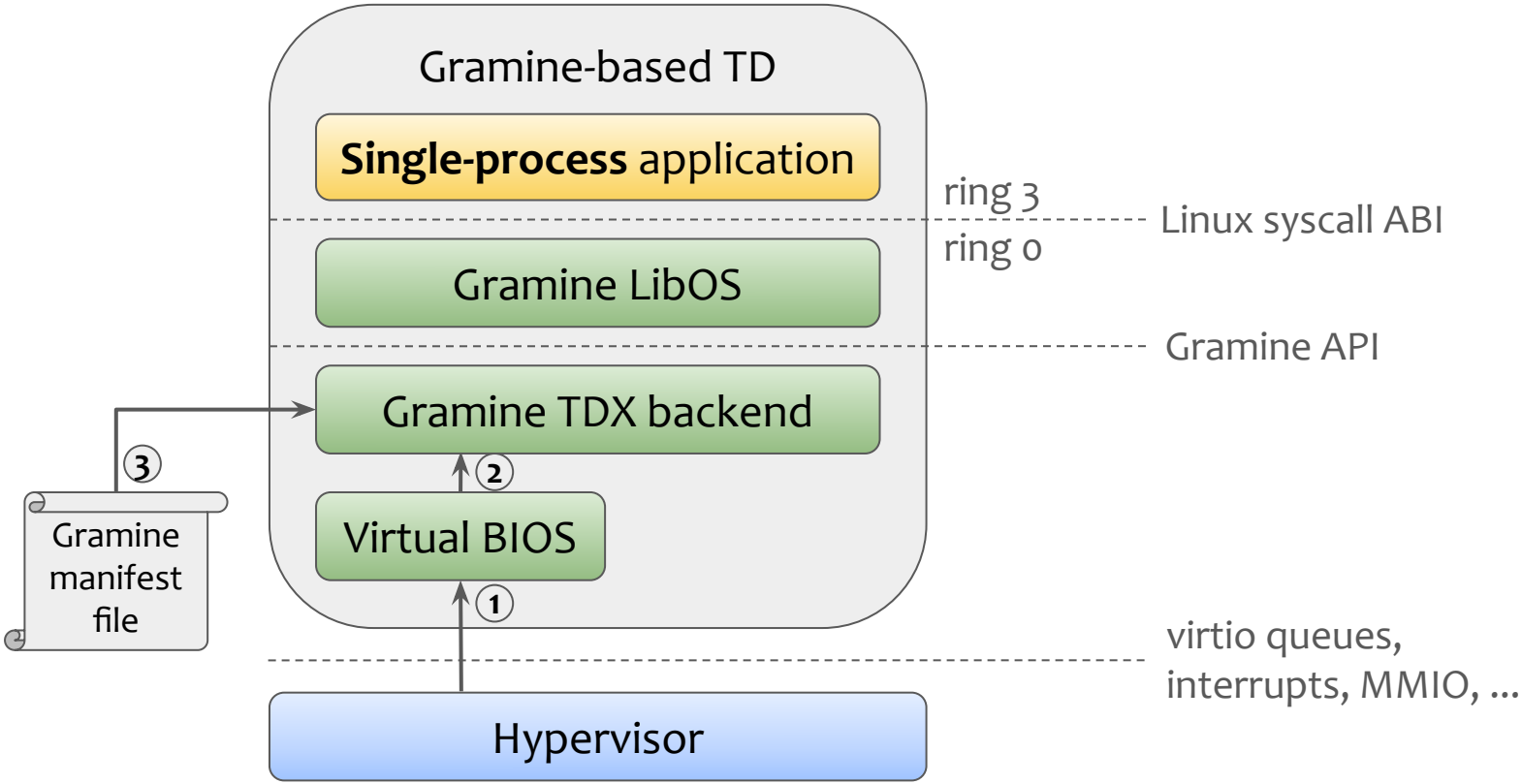
Design – Gramine-TDX overview



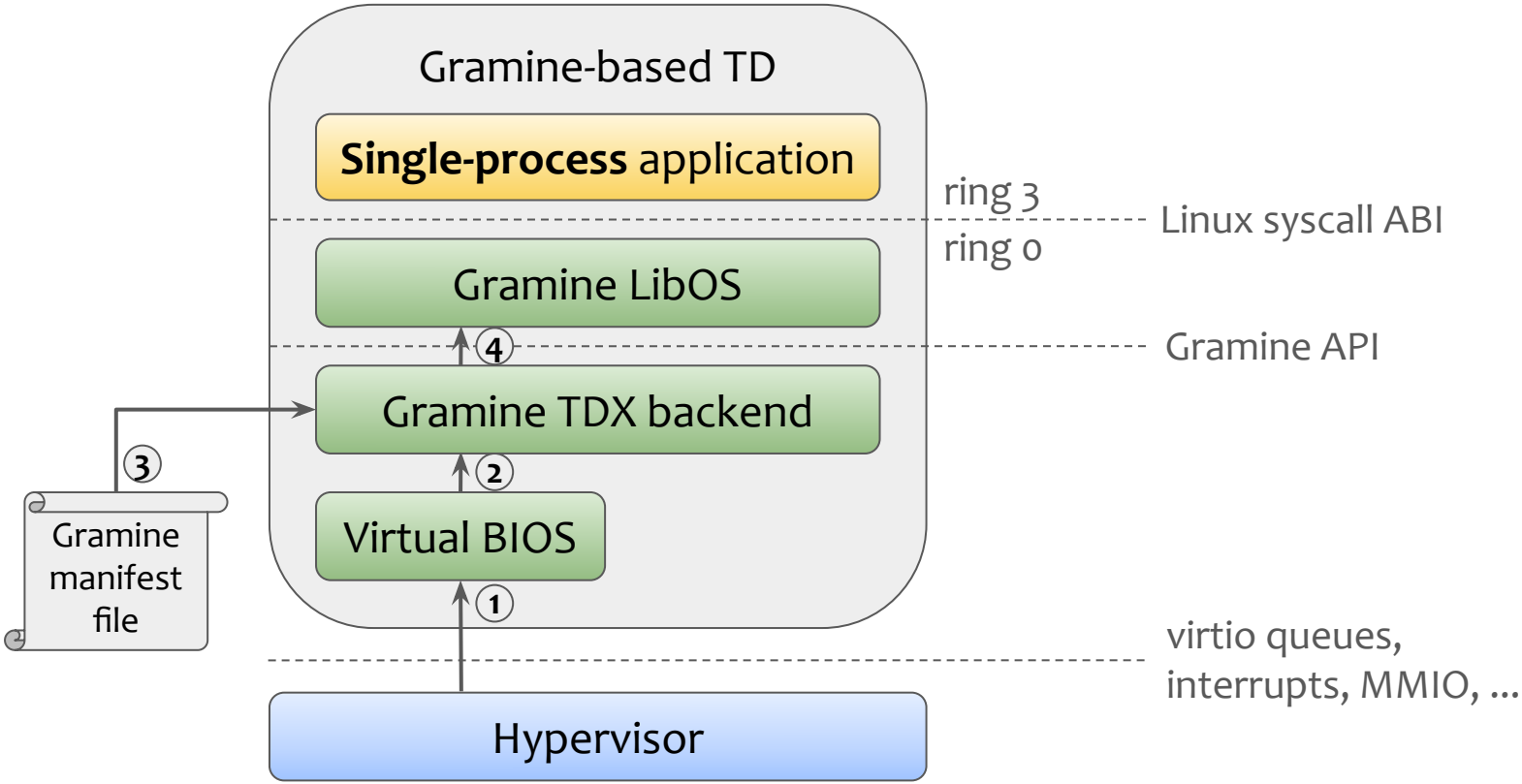
Design – Gramine-TDX overview



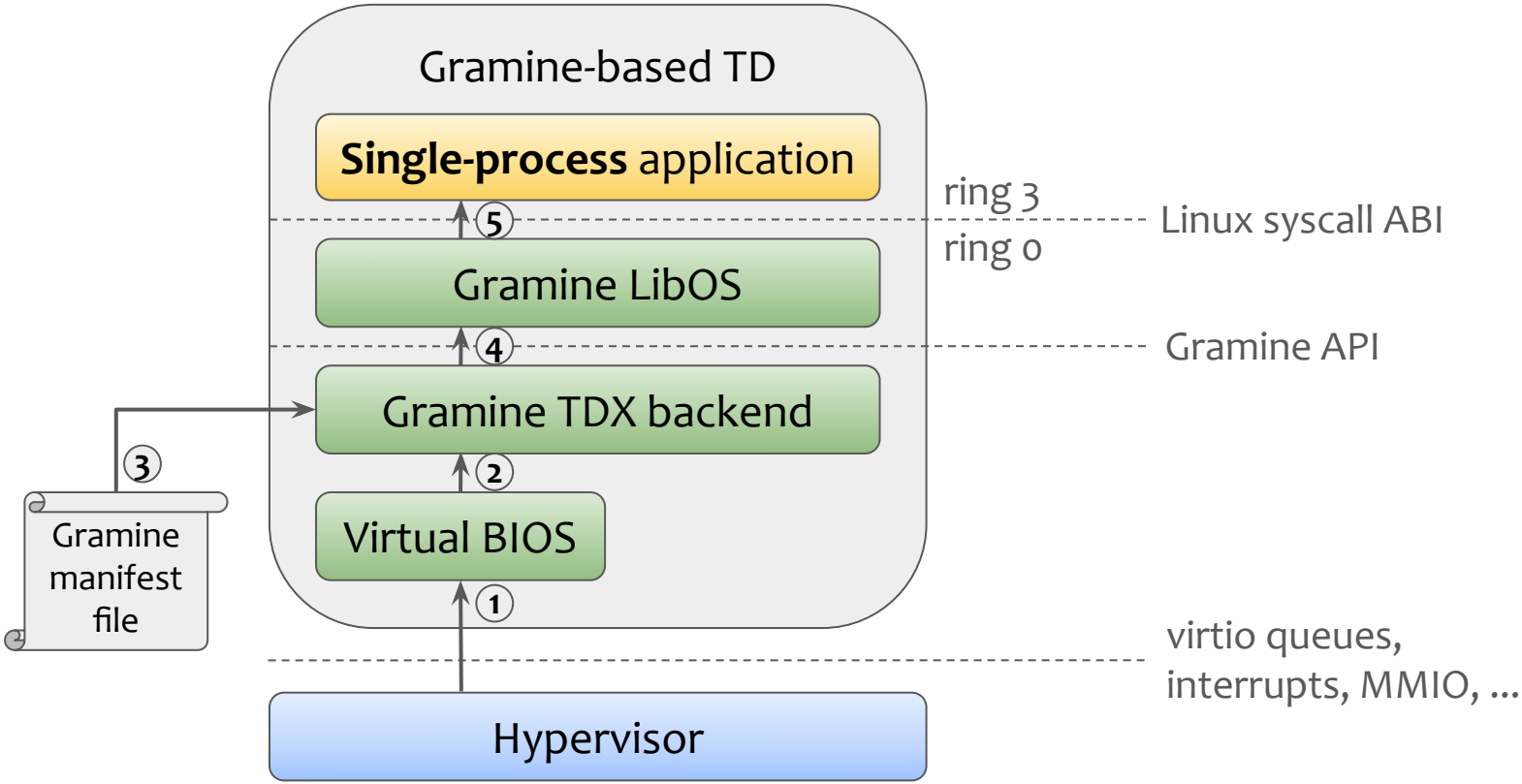
Design – Gramine-TDX overview



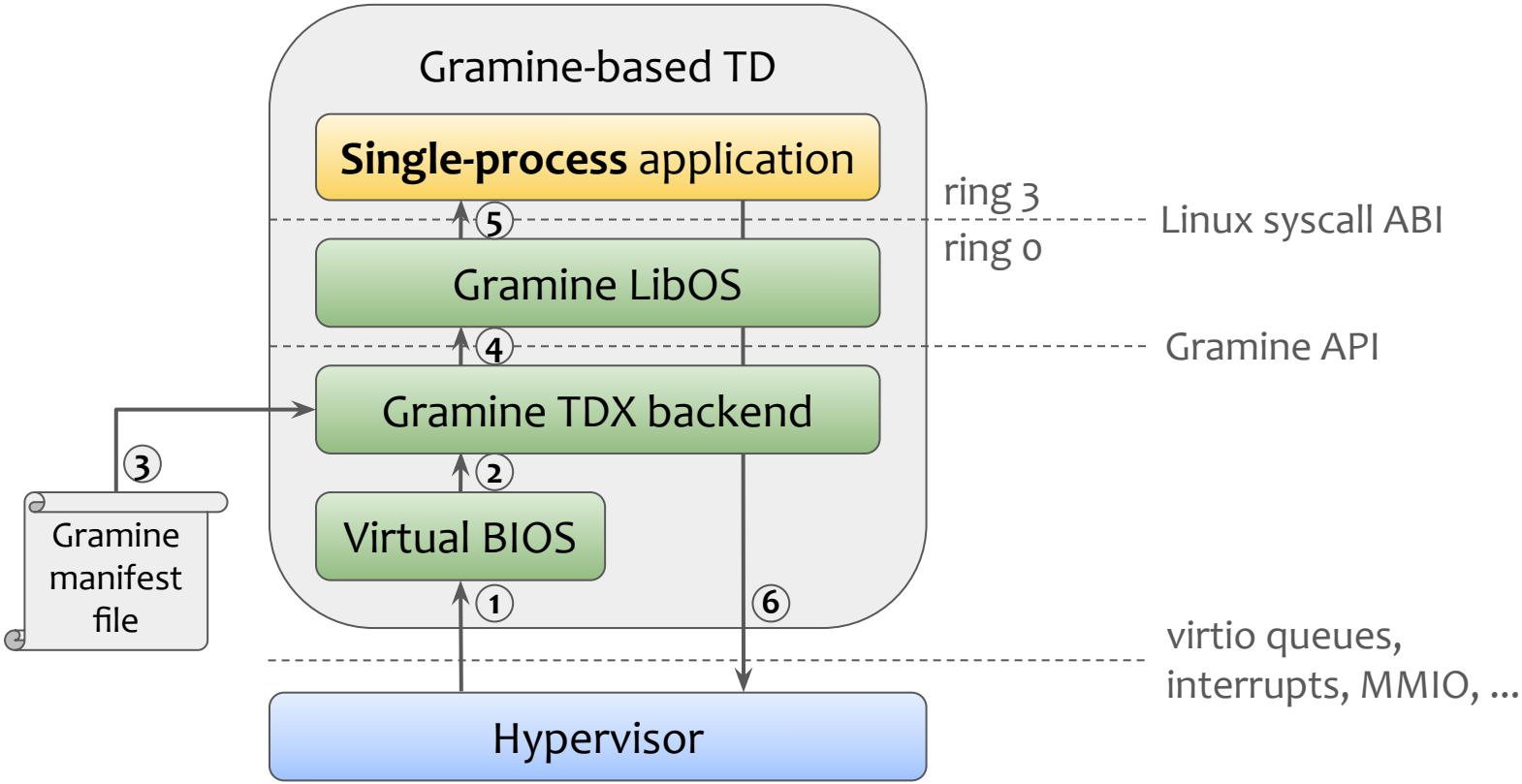
Design – Gramine-TDX overview



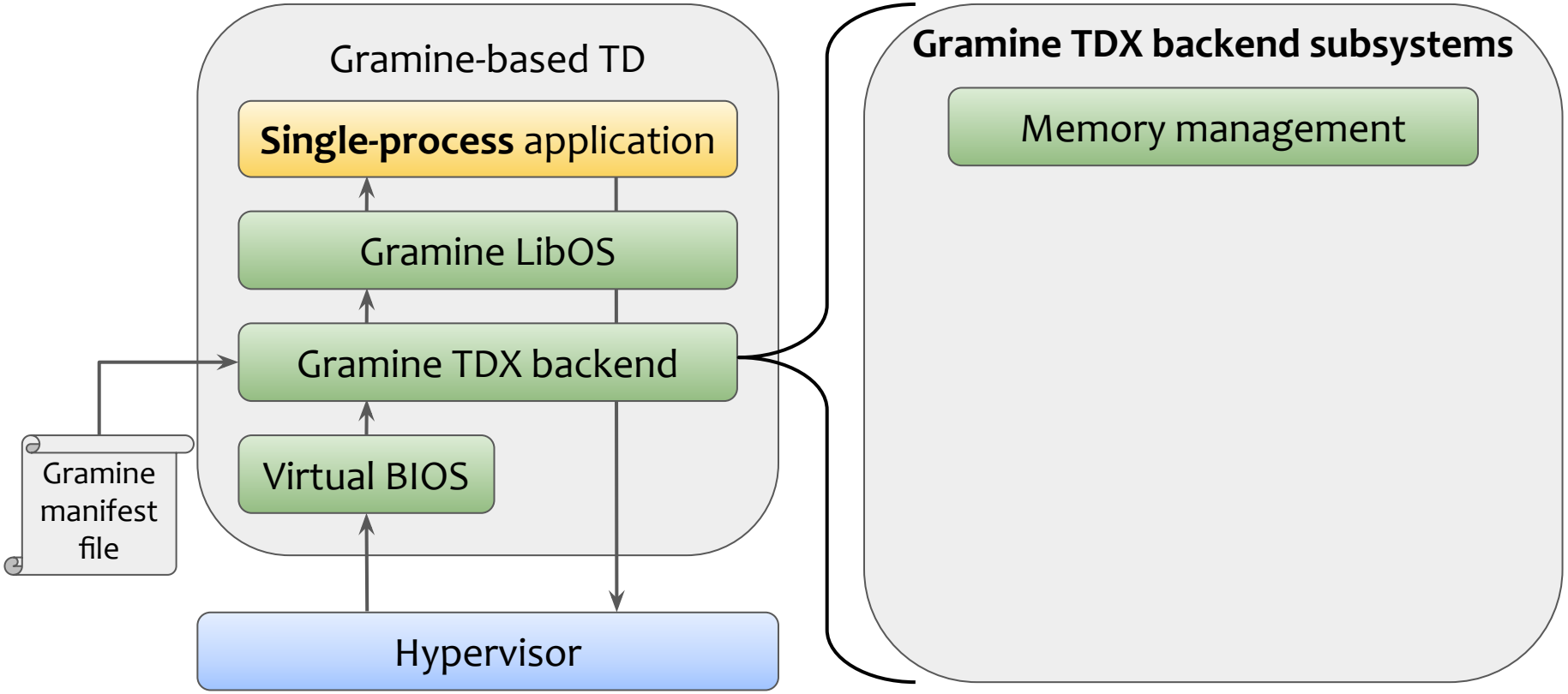
Design – Gramine-TDX overview



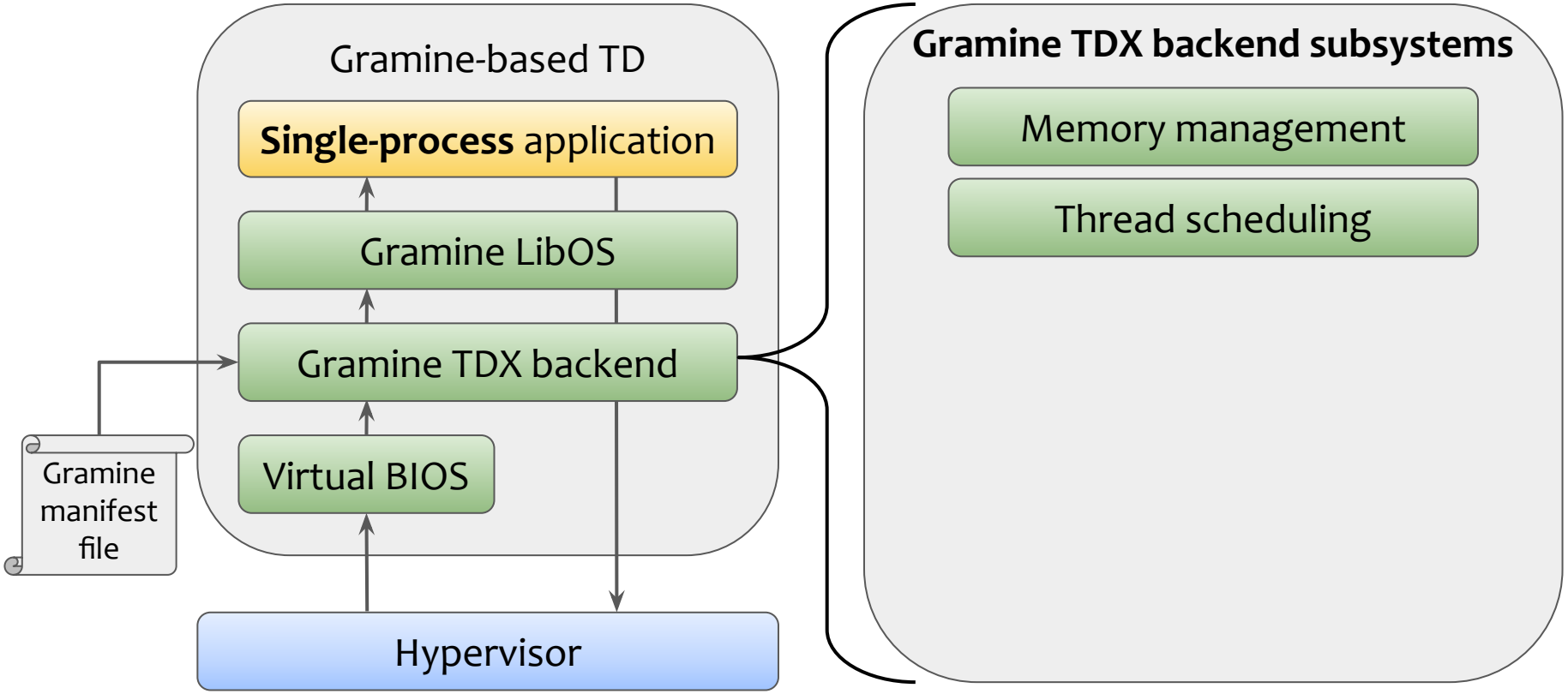
Design – Gramine-TDX overview



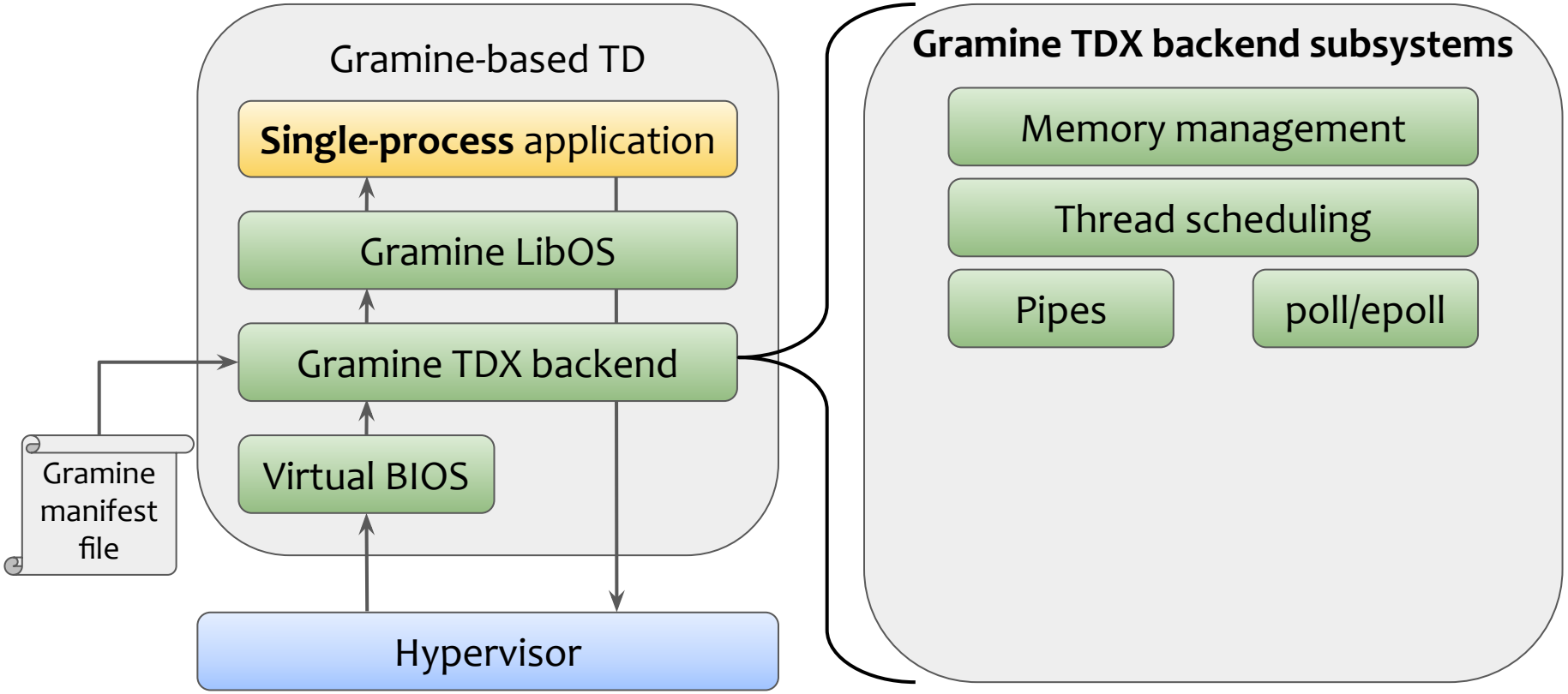
Design – Gramine-TDX deep-dive



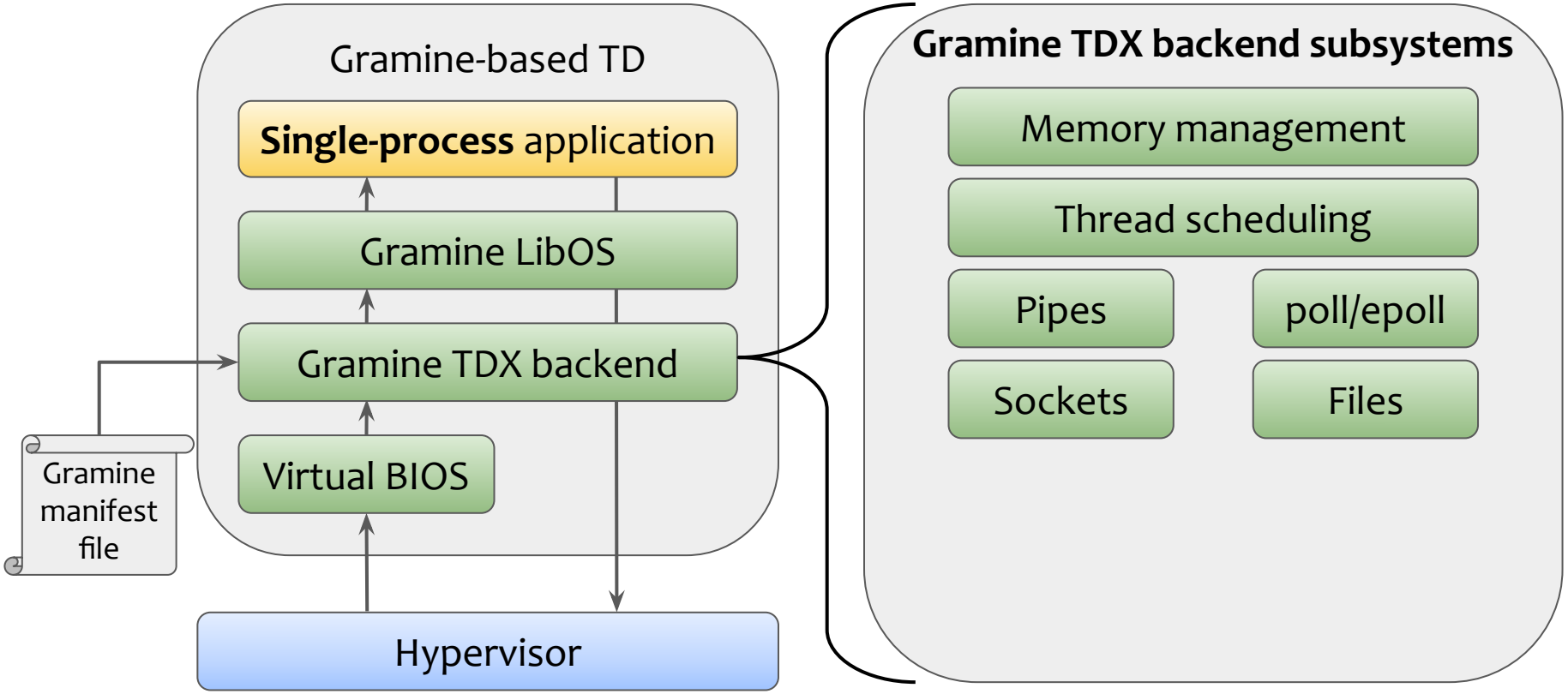
Design – Gramine-TDX deep-dive



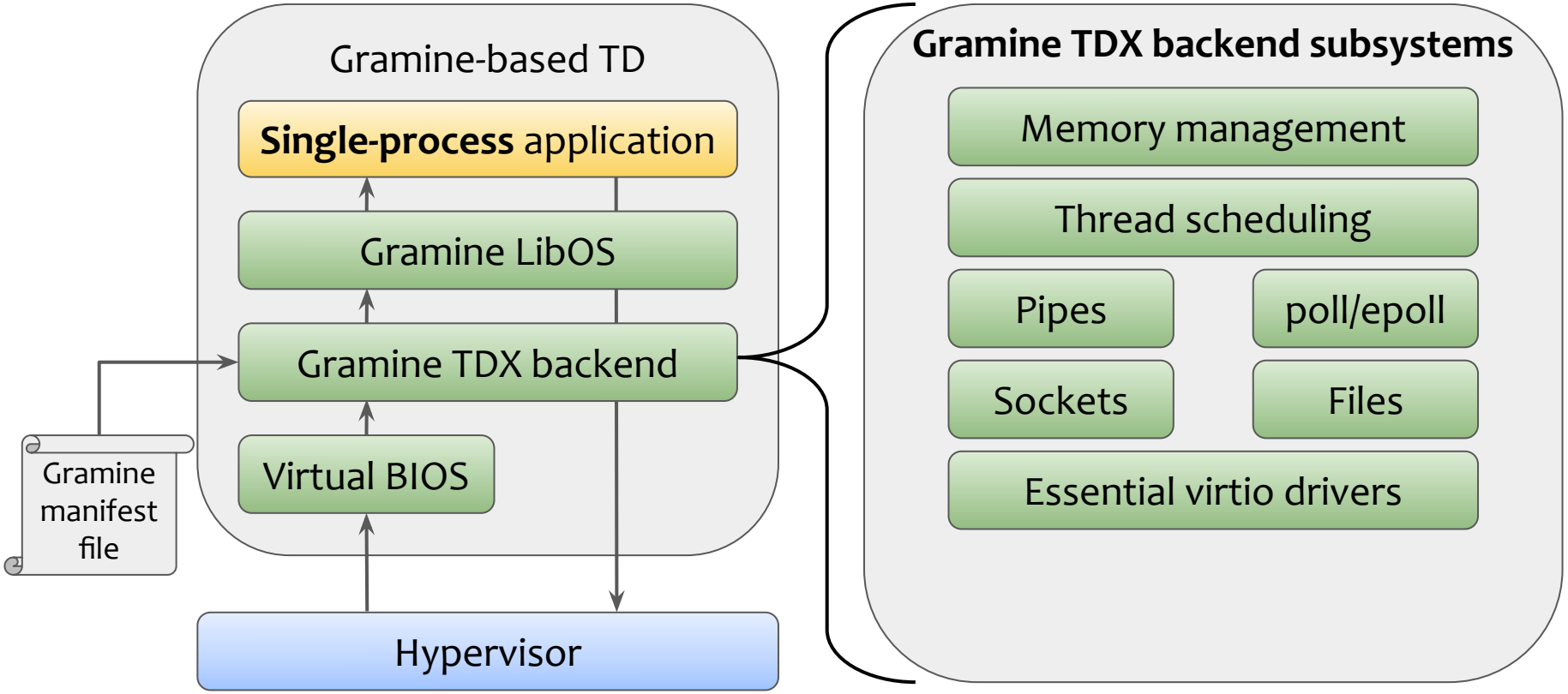
Design – Gramine-TDX deep-dive



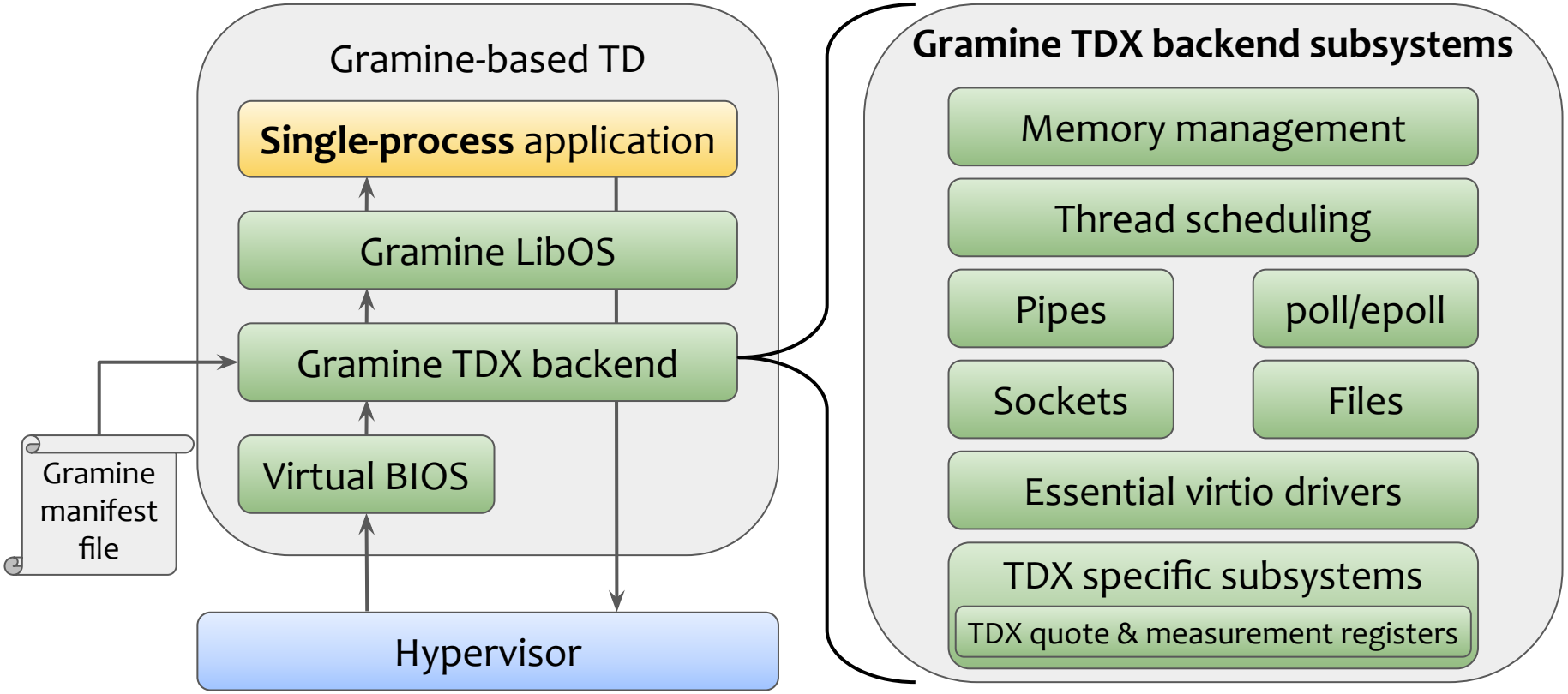
Design – Gramine-TDX deep-dive



Design – Gramine-TDX deep-dive



Design – Gramine-TDX deep-dive

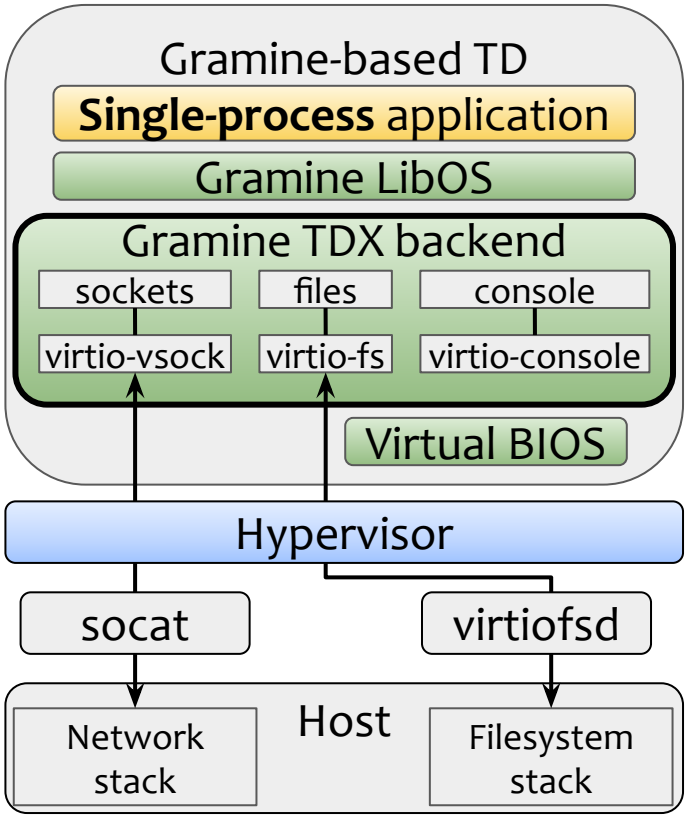


Outline

- ~~Motivation~~
- ~~Design~~
- Implementation
- Evaluation

Implementation

- TDX-Backend written from scratch (**~17K LoC**)
- Reuse of **70%** of Gramine code
- Minimal implementation of only **3 drivers**
- Use **TD-Shim** as the Virtual BIOS



Implementation – Security analysis

- Minimized attack surface
  - Limited entry points for untrusted inputs & use of trusted HW primitives (e.g., CPUID)
- Input validation and hardening
  - Inputs are verified against expected ranges/values and known safe hashes
- Confidentiality and integrity
  - Protection of code and data via the TEE, files via the manifest and networking via TLS
- Secure measurement of software stack
  - Intel TDX remote attestation ensures trust in the running environment

Outline

- ~~Motivation~~
- ~~Design~~
- ~~Implementation~~
- Evaluation

Security evaluation

Kernel	Binary (MB)	LoC	#inputs
Ubuntu 22.04 v5.19	68	~2M	15628
Intel TDX v5.19	56	~2M	1098
Firecracker v6.1	27	~1M	911
Gramine-TDX	1.2	57K	177

Security evaluation

Kernel	Binary (MB)	LoC	#inputs
Ubuntu 22.04 v5.19	68	~2M	15628
Intel TDX v5.19	56	~2M	1098
Firecracker v6.1	27	~1M	911
Gramine-TDX	1.2	57K	177

Security evaluation

Kernel	Binary (MB)	LoC	#inputs
Ubuntu 22.04 v5.19	68	~2M	15628
Intel TDX v5.19	56	~2M	1098
Firecracker v6.1	27	~1M	911
Gramine-TDX	1.2	57K	177

Gramine-TDX is ~50× smaller than the Linux kernel and has a minimal attack surface

Performance evaluation

What is the performance impact of Gramine-TDX in:

- CPU-intensive applications
  - PyTorch, OpenVINO, TensorFlow, candle, Blender, Image processing apps
- Storage & network I/O intensive applications
  - SQLite, Redis, Memcached, lighttpd
- System operations
  - UnixBench
- Boot time
  - Microbenchmarks

Performance evaluation

What is the performance impact of Gramine-TDX in:

- **CPU-intensive applications**
  - PyTorch, OpenVINO, TensorFlow, candle, Blender, Image processing apps
- **Storage & network I/O intensive applications**
  - SQLite, Redis, Memcached, lighttpd
- System operations
  - UnixBench
- Boot time
  - Microbenchmarks

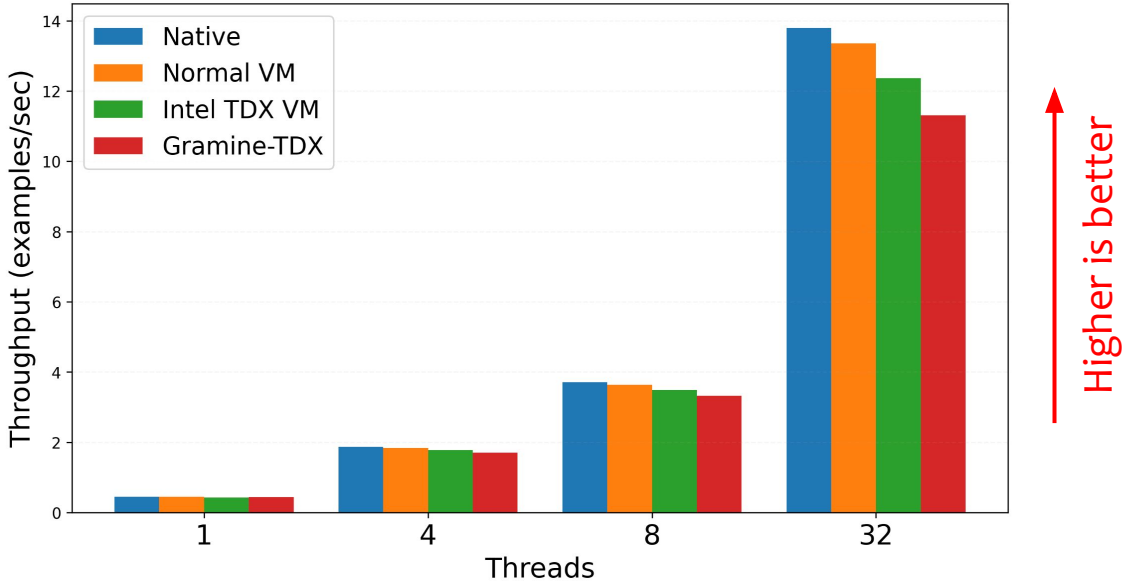
Performance evaluation

- Experimental setup:
  - Intel Xeon Platinum 8570 PU (3.60GHz, 56 cores)
  - 1 TB (16 channels x 64 GB) DRAM
  - Intel TDX Module v1.5
  - Host & Guest: Intel TDX-enabled Linux kernel v6.8

- Variants:
  - *Native* → Bare-metal execution
  - *Normal VM* → Execution in standard VM with Linux kernel
  - *Intel TDX VM* → Execution in Intel TDX VM with Linux kernel
  - *Gramine-TDX* →Execution in Intel TDX VM with Gramine-TDX kernel

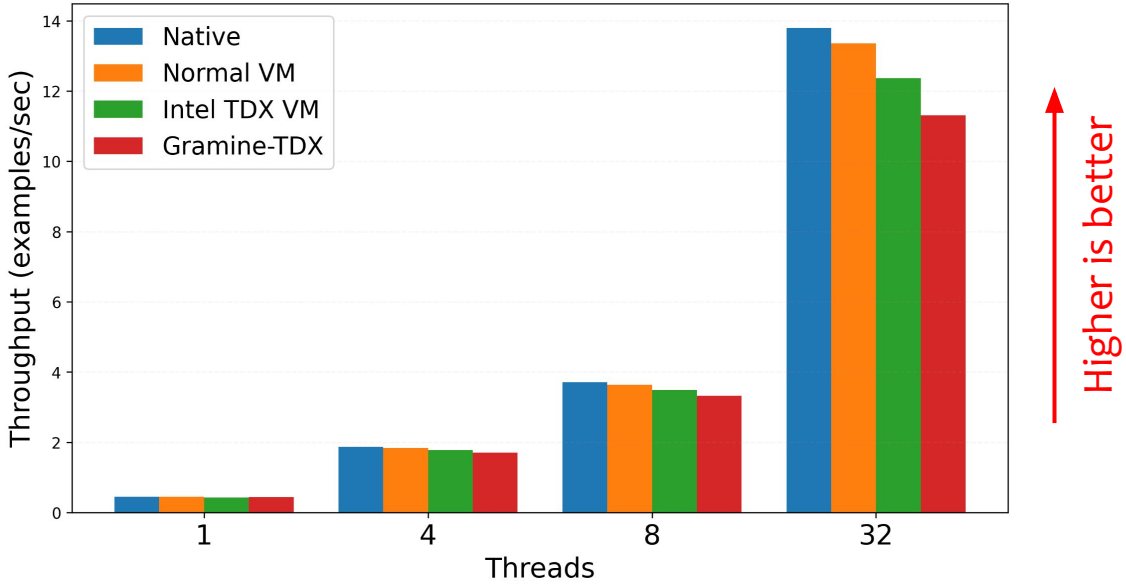
CPU intensive applications

Tensorflow app using the BERT Large model



CPU intensive applications

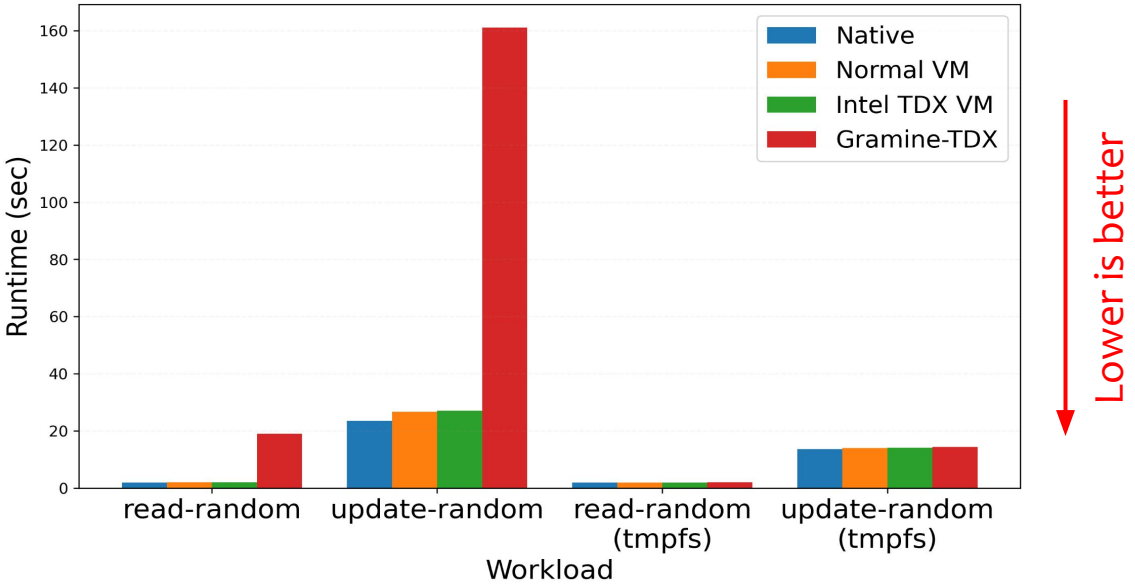
Tensorflow app using the BERT Large model



Gramine-TDX incurs minimal overheads in CPU intensive applications

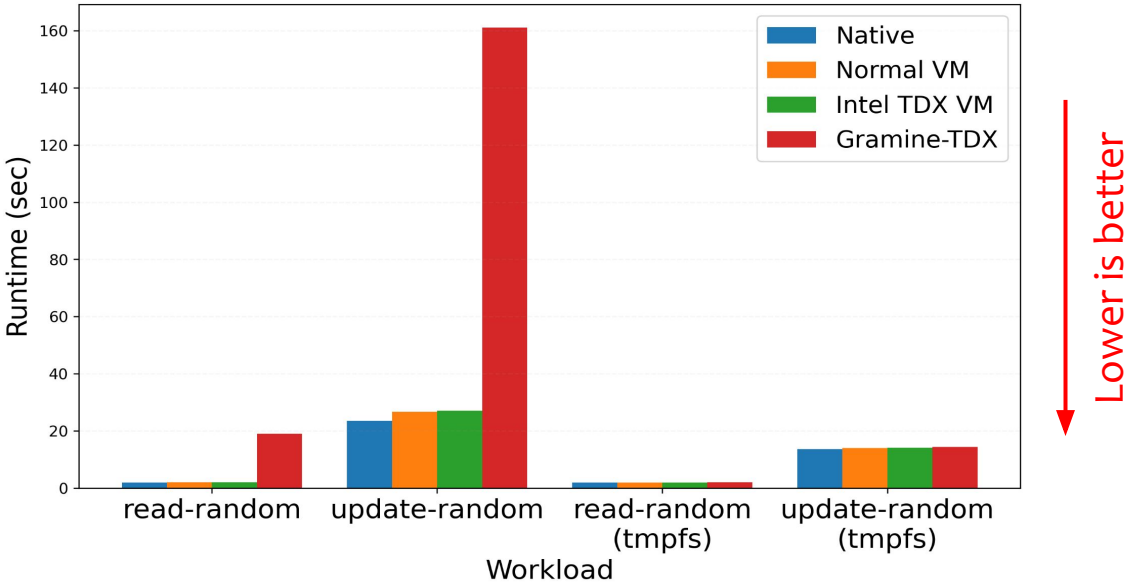
Storage I/O intensive applications

SQLite *kvtest* workloads using files backed by different filesystems (*ext4*, *tmpfs*)



Storage I/O intensive applications

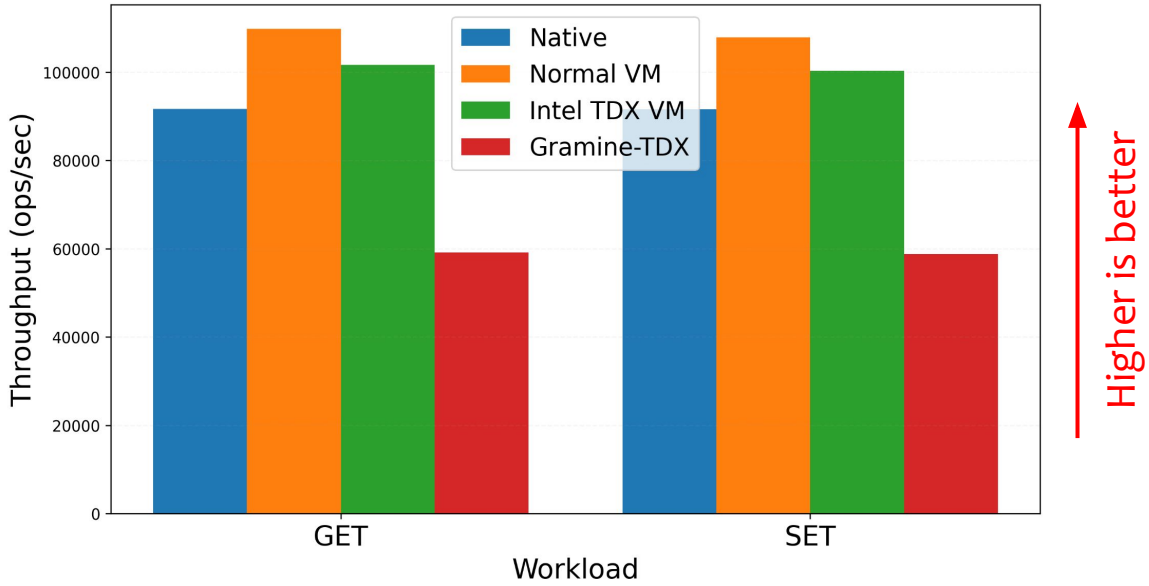
SQLite *kvtest* workloads using files backed by different filesystems (*ext4*, *tmpfs*)



Gramine-TDX pays a high performance tax for file I/O done through its virtio-fs driver

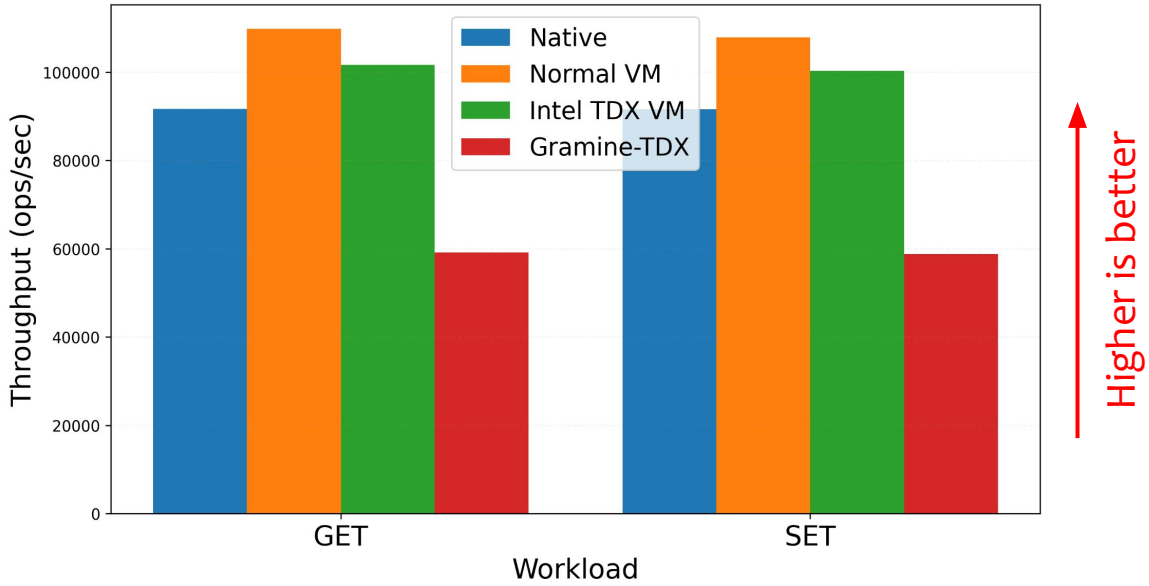
Network I/O intensive applications

Redis server throughput (redis-benchmark, default settings)



Network I/O intensive applications

Redis server throughput (redis-benchmark, default settings)



Gramine-TDX can incur considerable overheads in network I/O intensive applications

Summary

How to design a **minimal, security-first** kernel for confidential VMs with a **small attack surface**?

Gramine-TDX: A Lightweight OS Kernel for Confidential VMs

- Minimal attack surface
- Small Trusted Computing Base (TCB)
- Compatibility with diverse applications, frameworks, and languages
- Easy to use and deploy

Code

Try it out!

https://github.com/gramineproject/gramine-tdx