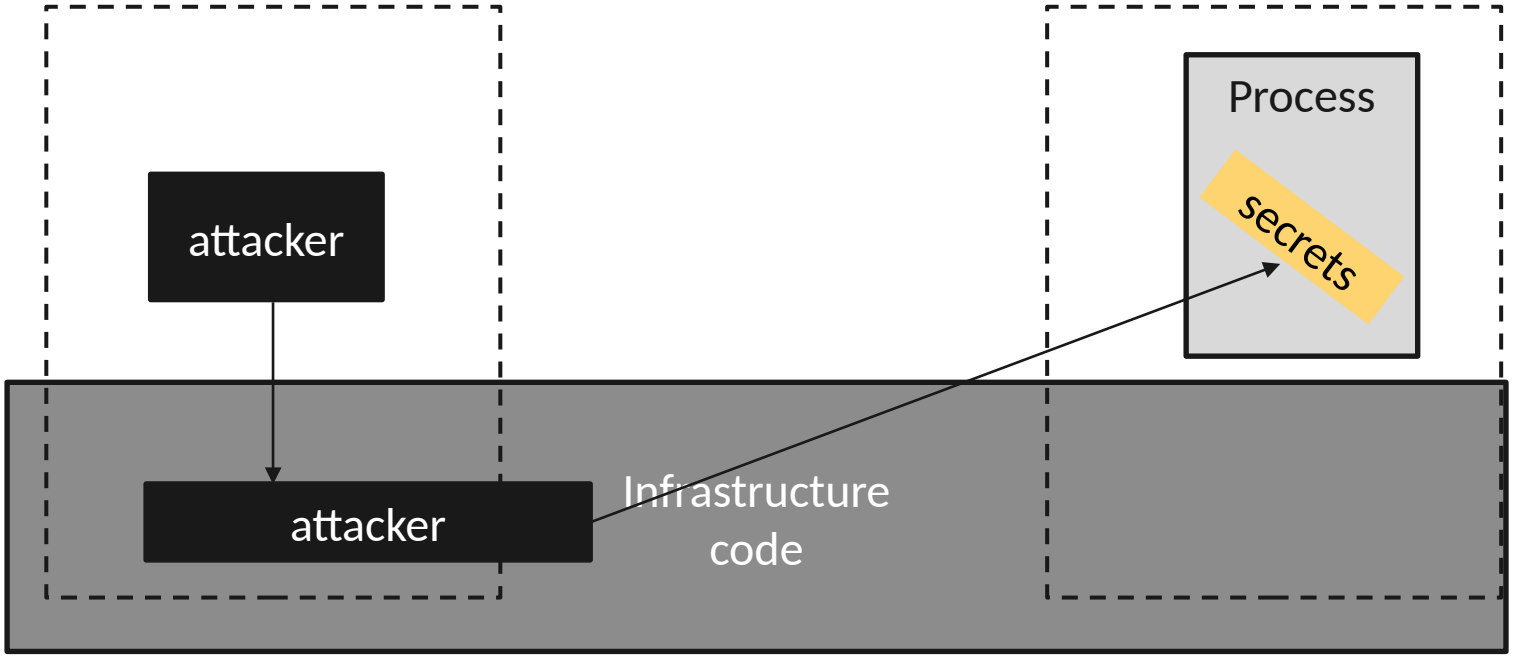
Say Goodbye to Virtualization for a Safer Cloud

Dan Williams, Ricardo Koller, Brandon Lum
IBM T.J. Watson Research Center

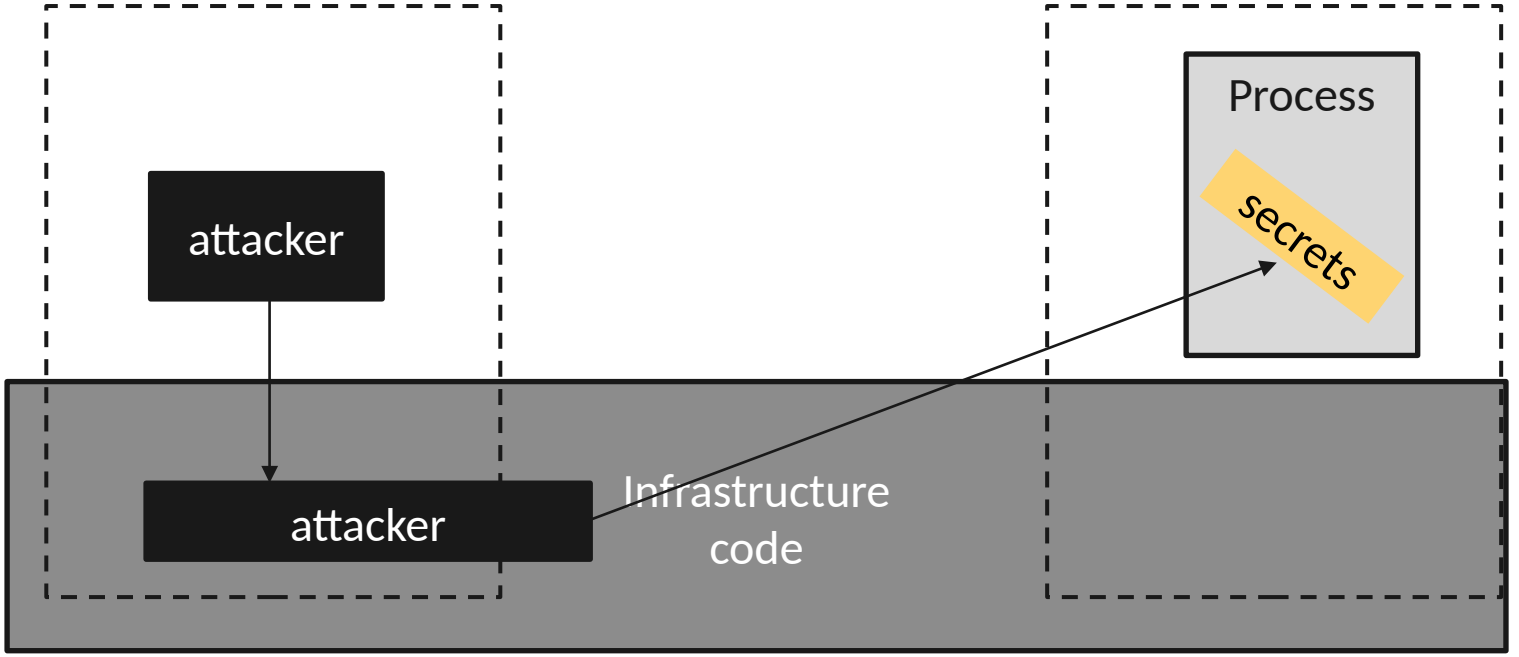
Hotcloud18

IBM

Isolation is important for the cloud

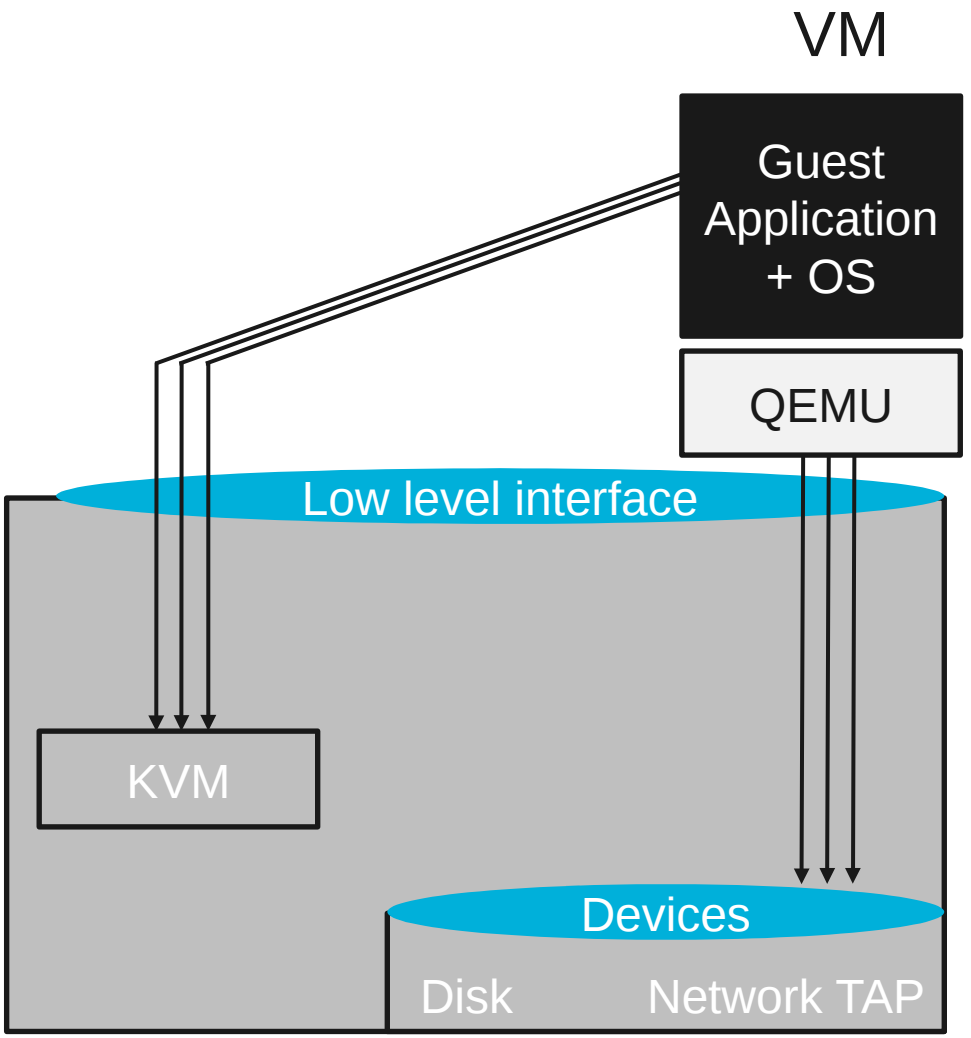


Isolation is important for the cloud



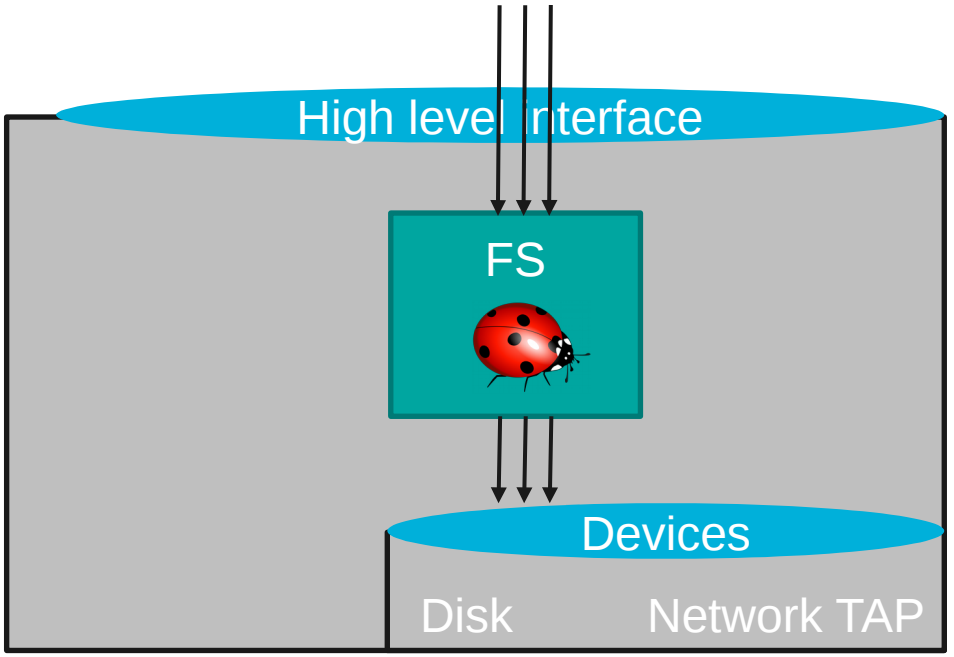
- Virtualization is the gold standard for isolation

Why are VMs the gold standard for isolation?



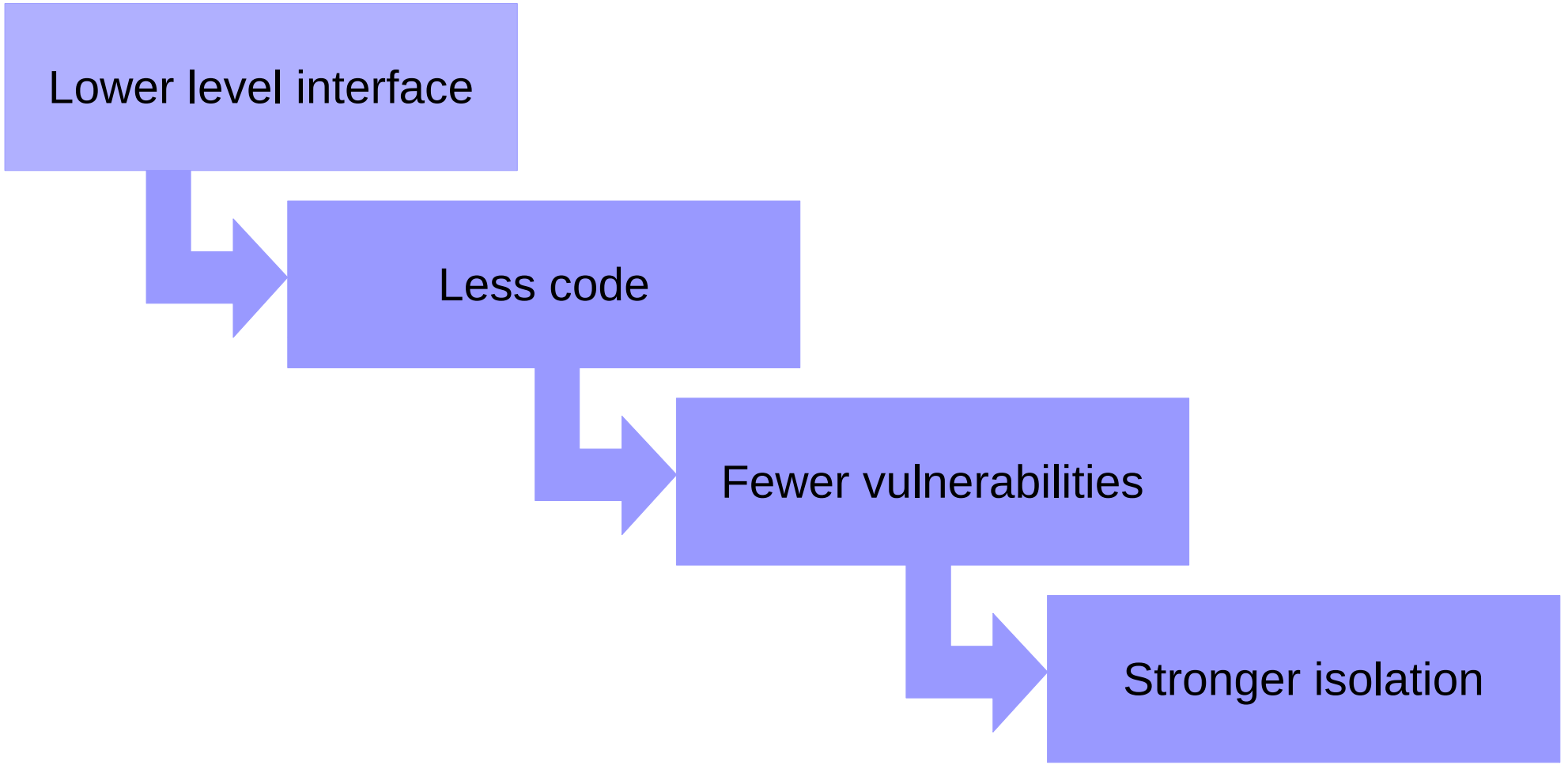
- Virtual machines use a low level interface to the host

Why are VMs the gold standard for isolation?



"Dirty COW"
(CVE-2016-5195)

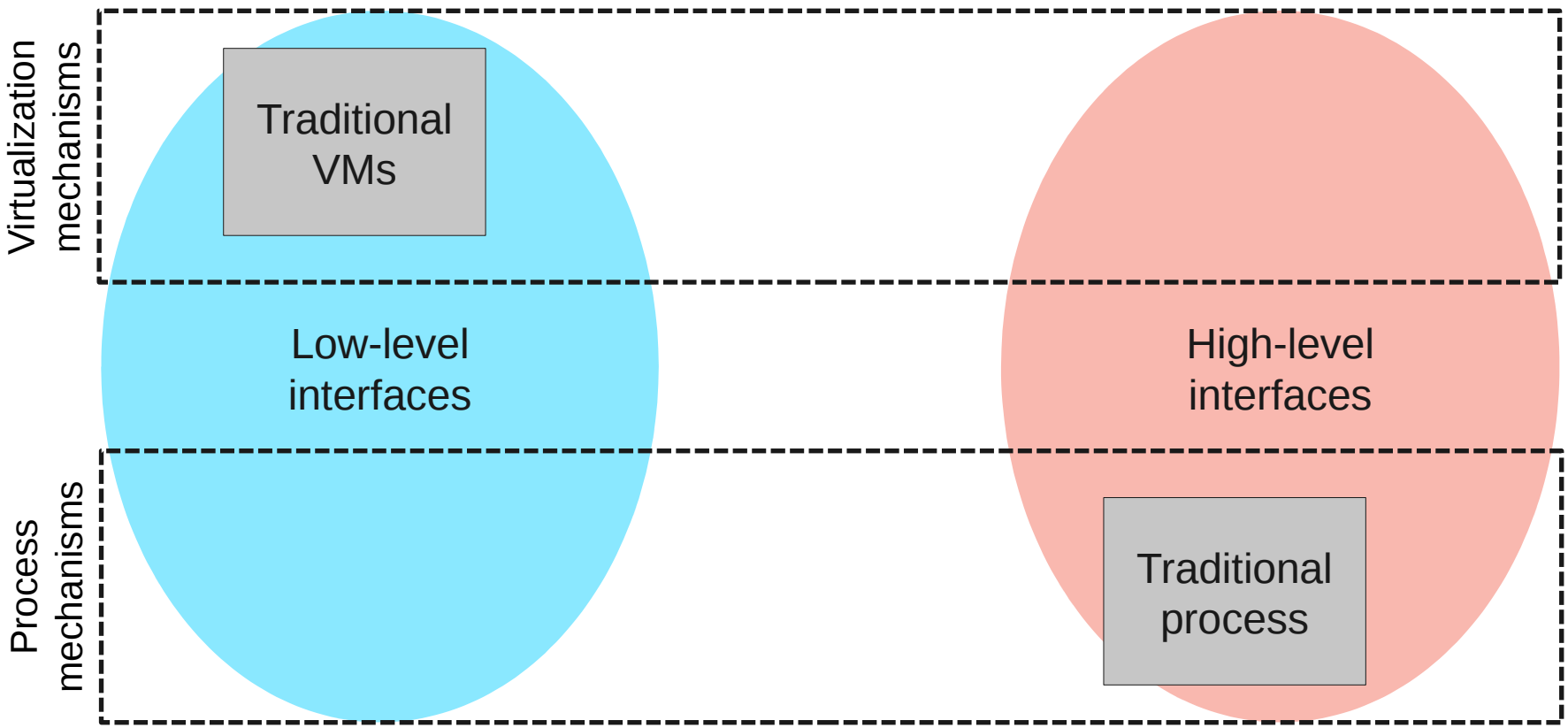
- A high level interface increases the chances of hitting a bug like the one above


Lower level interface
Less code
Fewer vulnerabilities
Stronger isolation

- The level of interface has nothing to do with virtualization

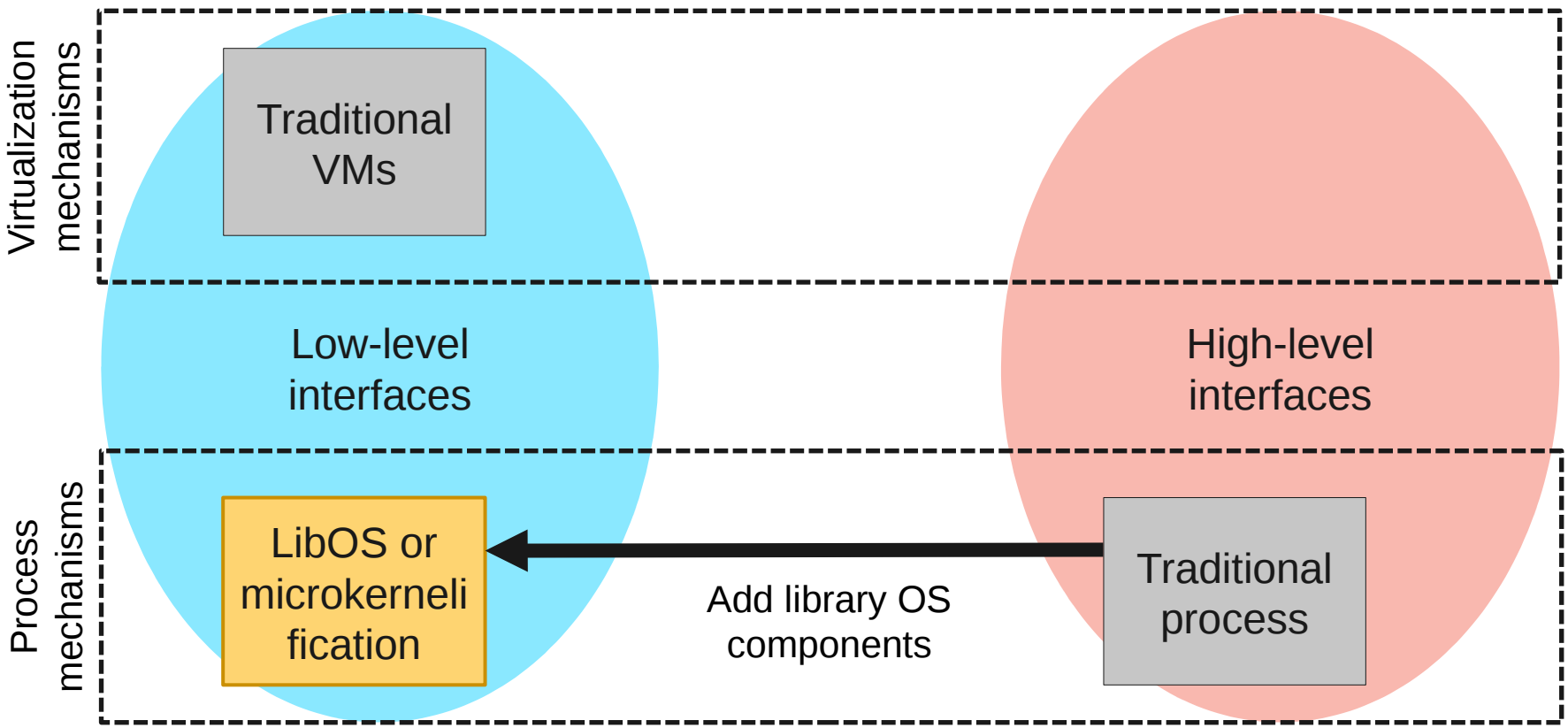
- Virtualization makes things worse with regards to isolation

Interface level ≠ mechanism



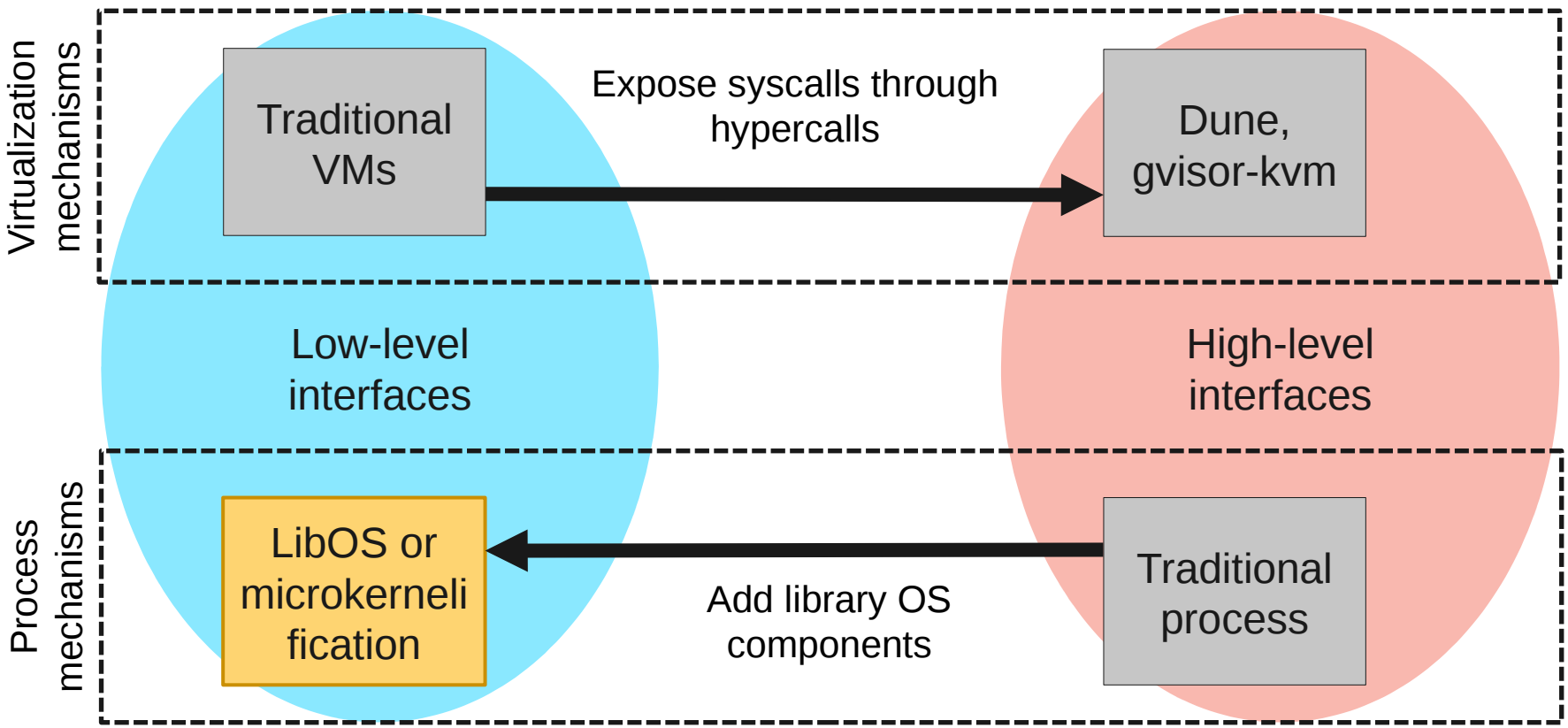
- For historic reasons we tend to equate the interface mechanism with what it was created for

Interface level ≠ mechanism

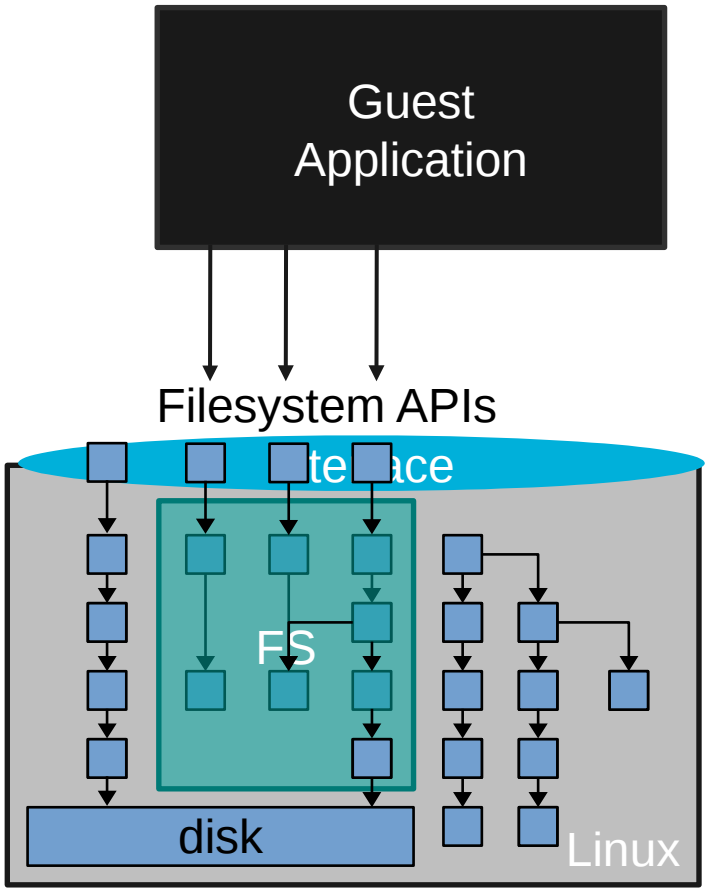


- We will show how to construct a system with a low level interface that uses process mechanisms

Interface level ≠ mechanism

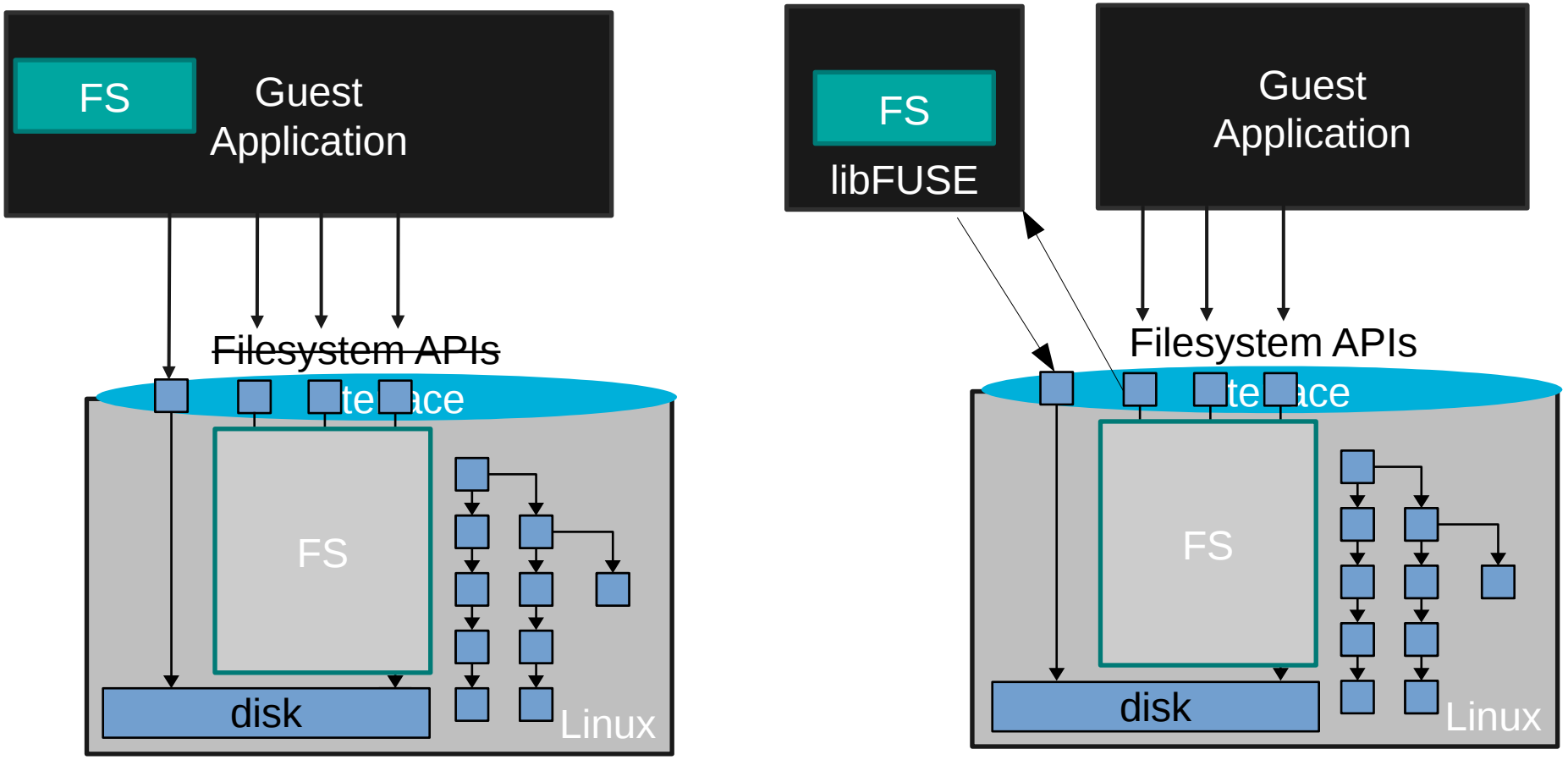


Adjust the interface level



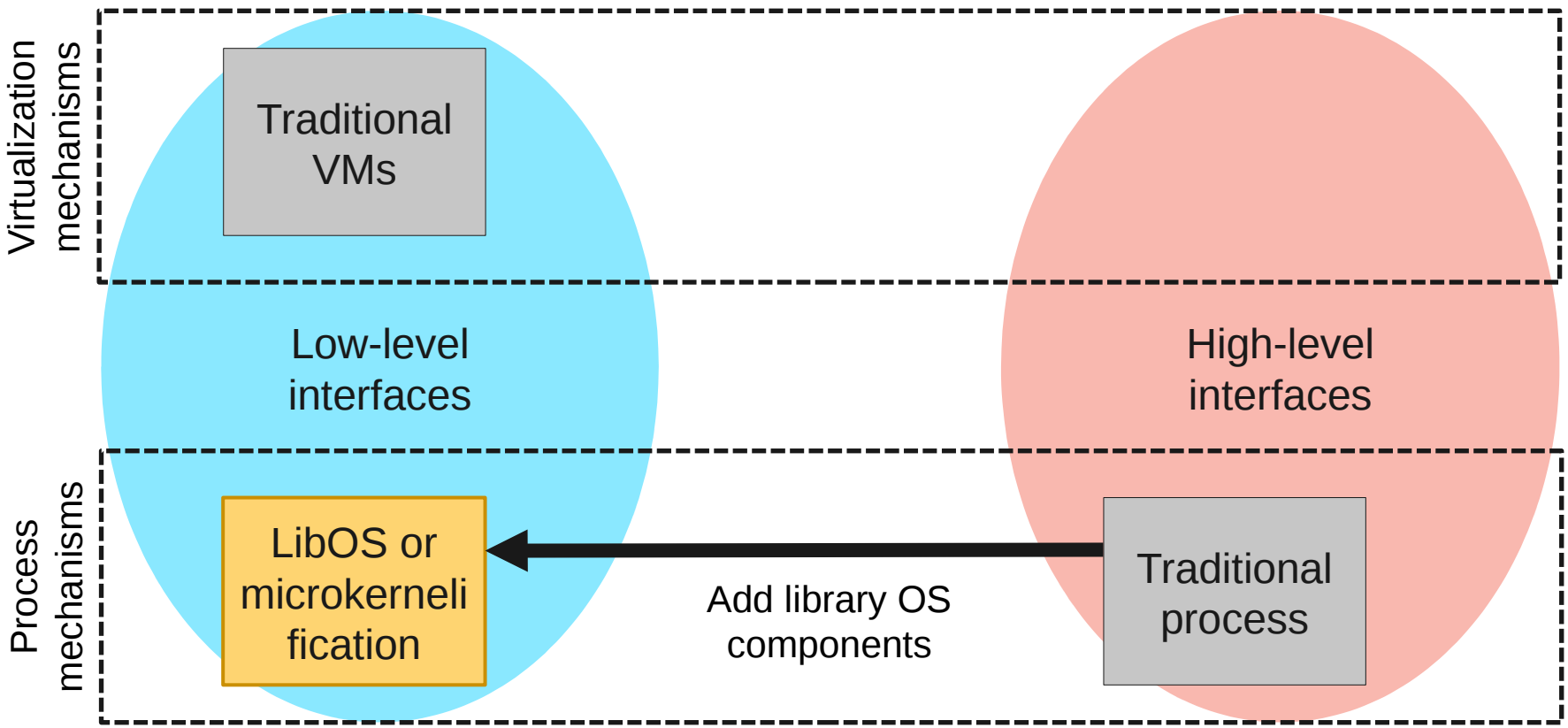
- An application using files uses a file system provided by the host

Adjust the interface level

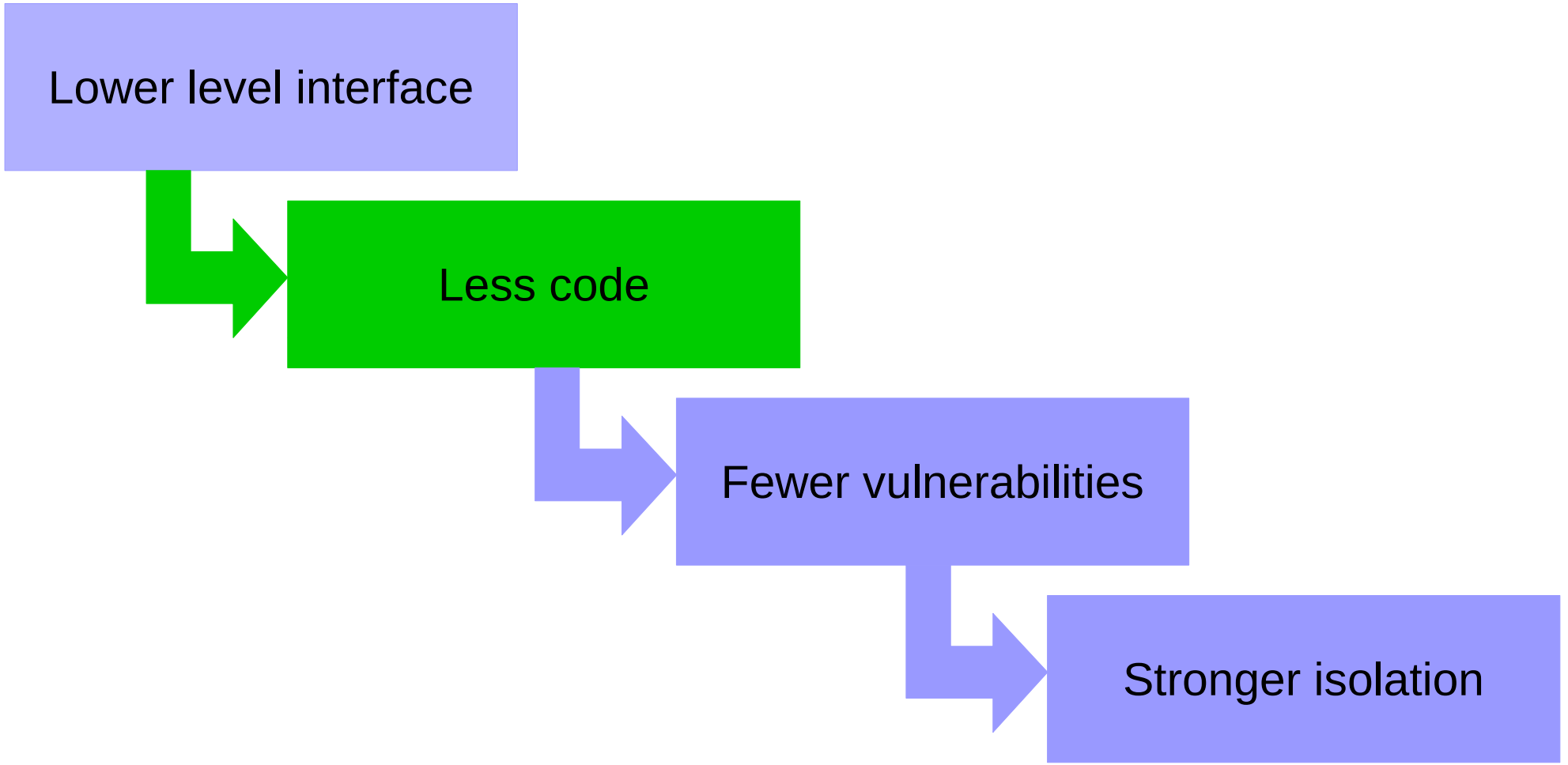


- Can adjust the level of interface by moving the implementation to user level

Interface level ≠ mechanism



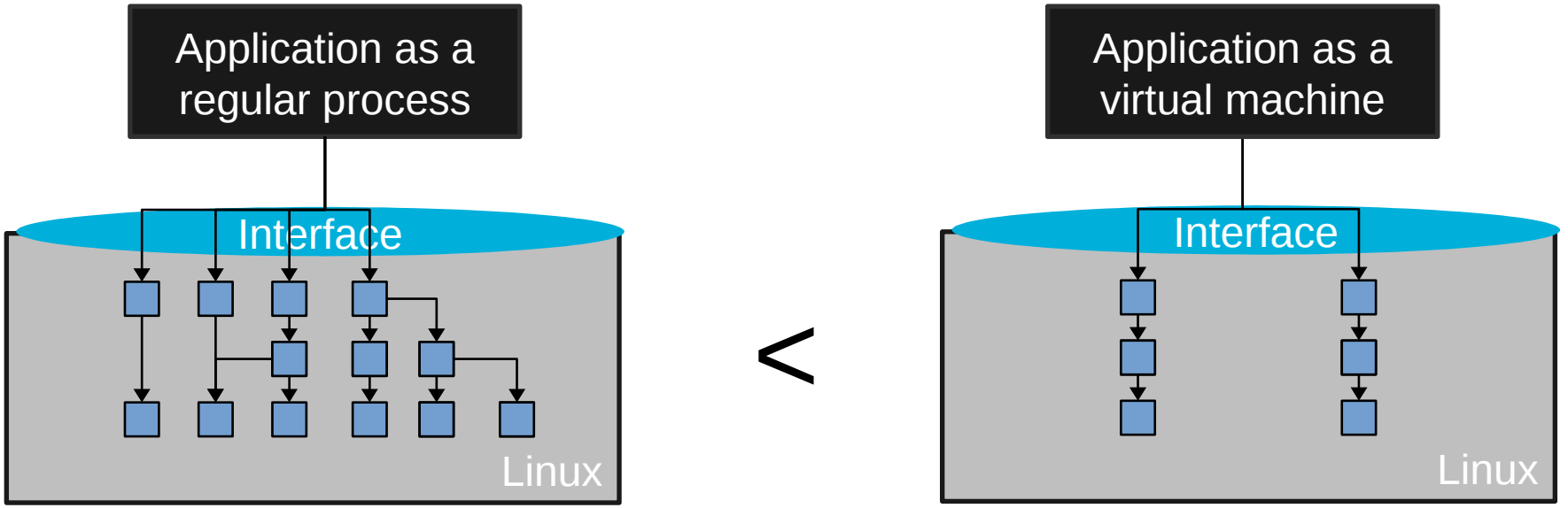
- How “much” isolation are we really gaining by going left?
- What’s the isolation “cost” of virtualization?



- Measure the amount of code used as a proxy for measuring isolation

Metric for Isolation

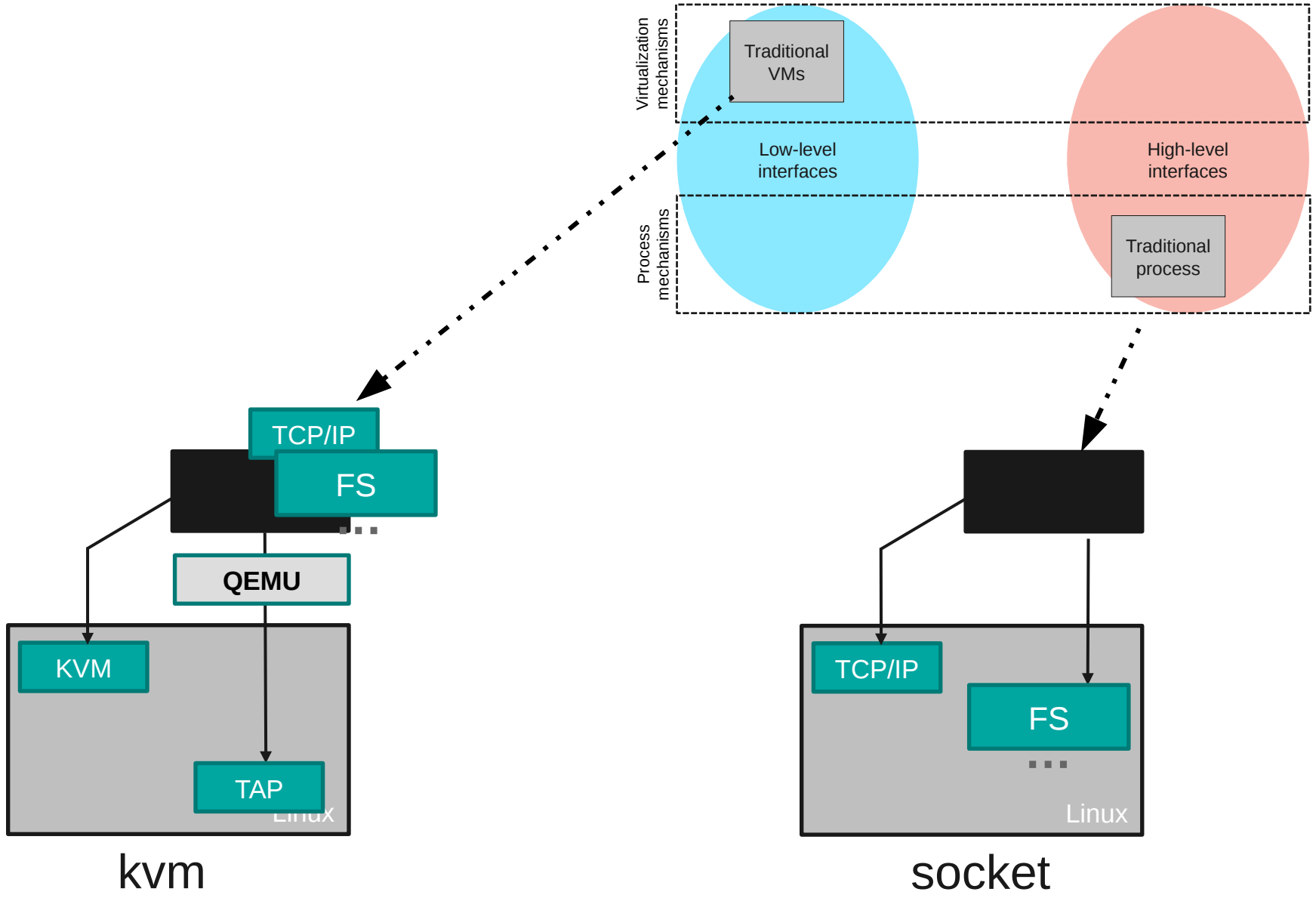
- Want to measure how much of the kernel is exposed
  - Kernel function tracing (ftrace)



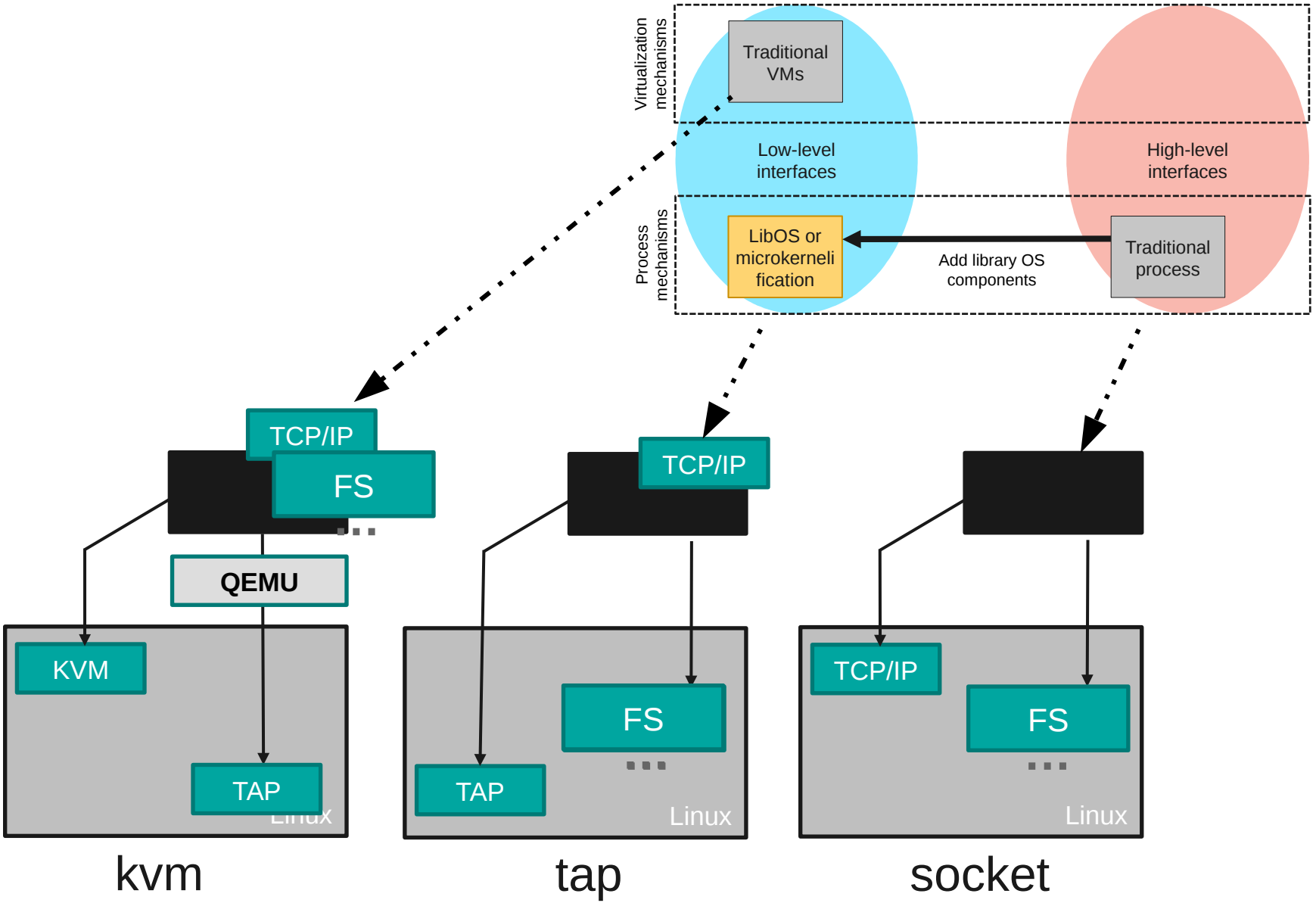
12 kernel functions

6 kernel functions

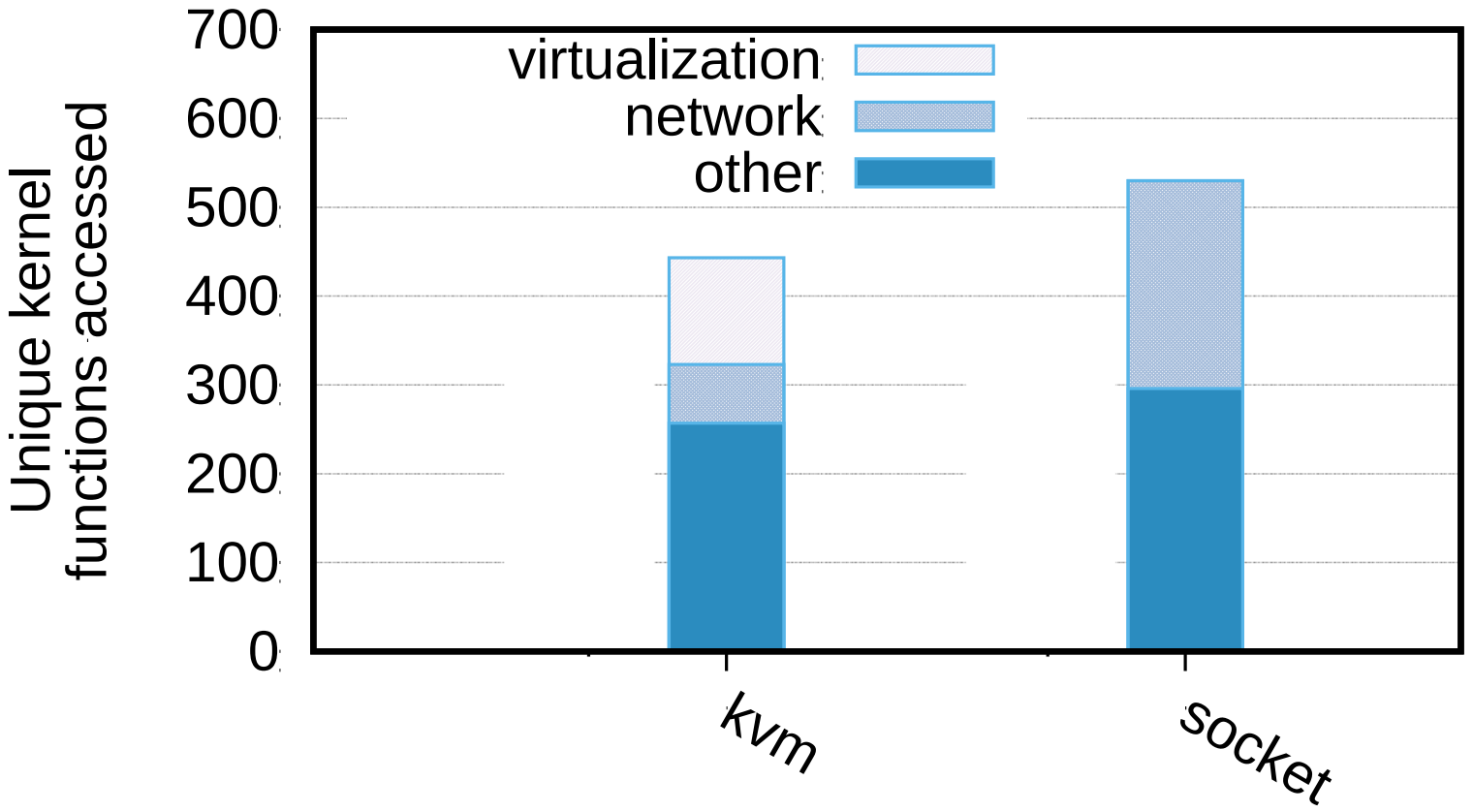
Comparing all options



Comparing all options



Dangers of virtualization mechanisms



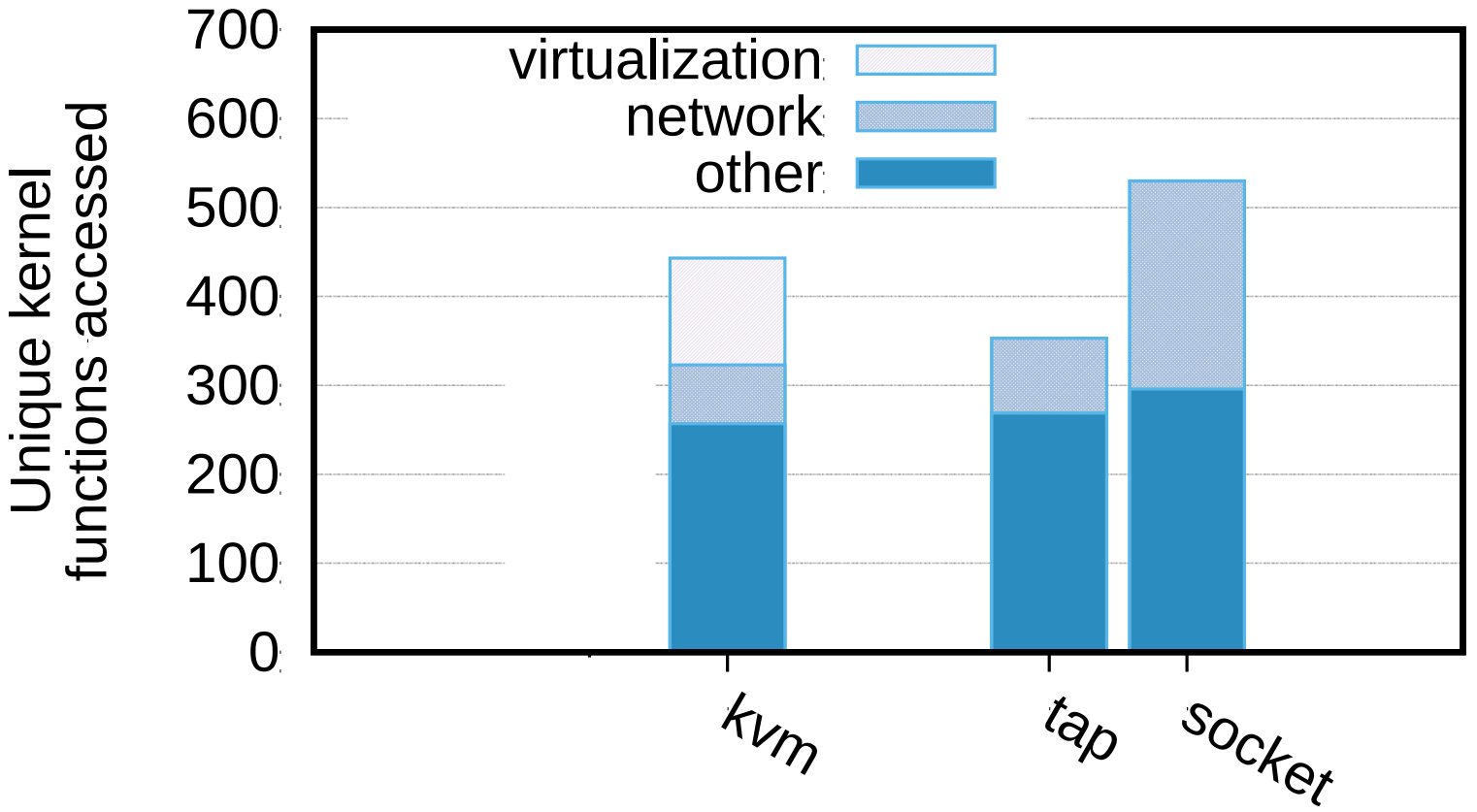
- kvm is better than socket
- The white area in kvm represents the cost of managing virtualization hardware

What’s the catch?

- How do we design systems to achieve better isolation in the cloud? Considering these:
  - Generality
  - Performance
  - Maintenance
- What isolation metric to use?

Nabla containers: https://nabla-containers.github.io/

Dangers of virtualization mechanisms



- tap reduces the amount ot networking needed as it's being pushed up
- tap is even lower than kvm

Discussion

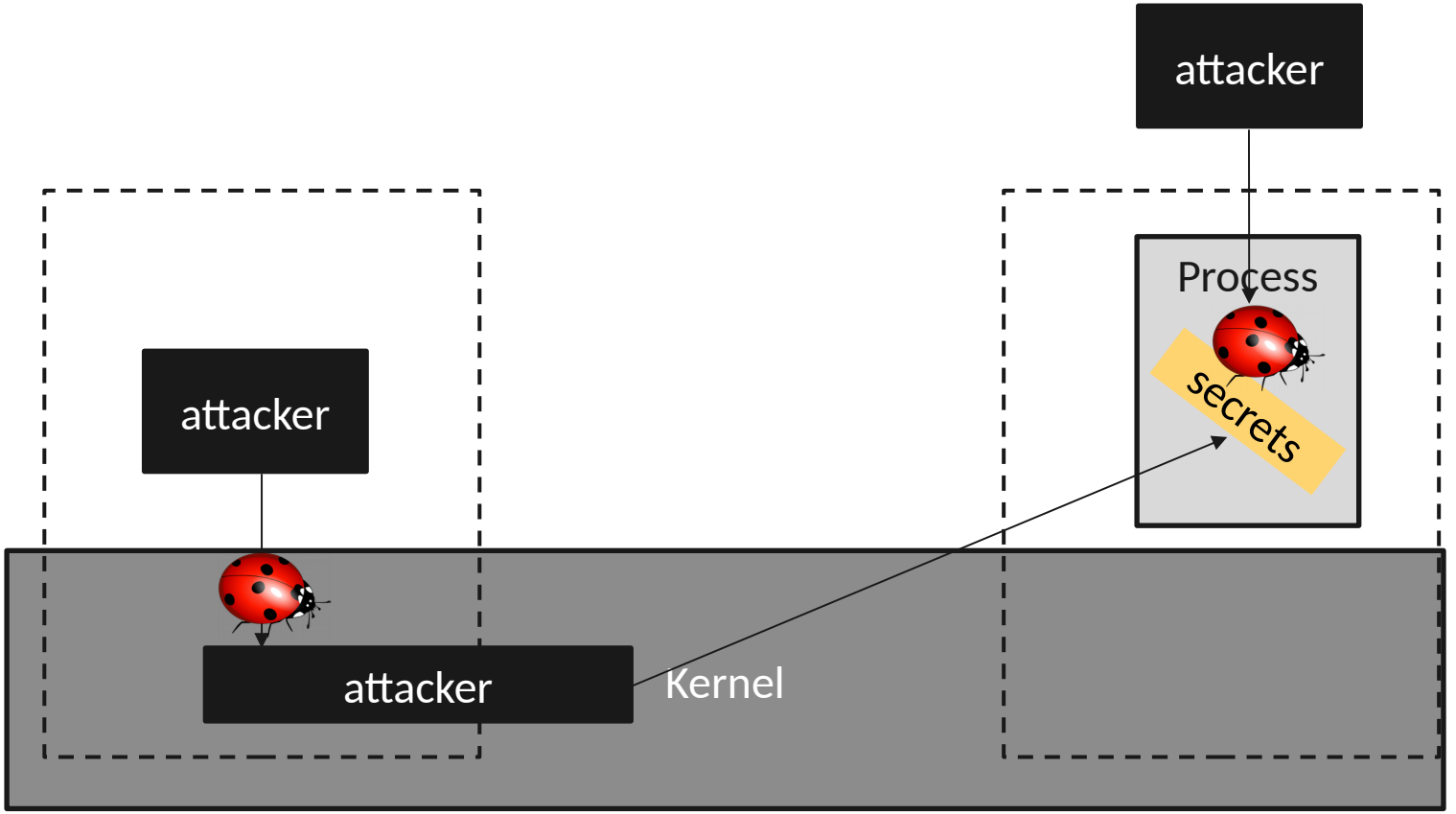
- How do we design systems to achieve better isolation in the cloud?

- What isolation metric to use?
  - Can our systems be bug free?
  - Is less code better?
    - What about sanitizing code? More code is better in that case

Nabla containers: https://nabla-containers.github.io/

BACKUP

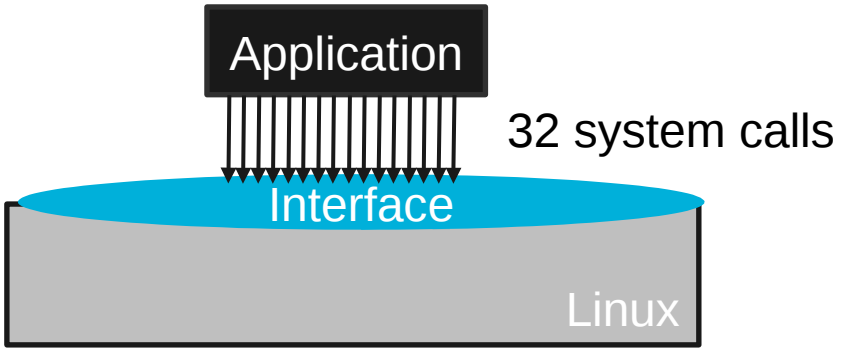
Horizontal vs. Vertical attacks



- How can we ensure the protection of secrets in a multi-tenant cloud

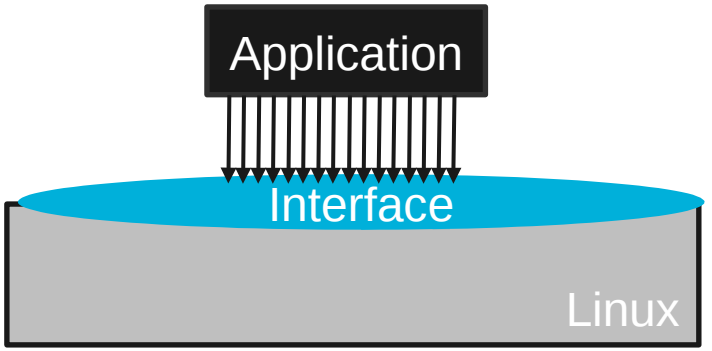
Metric for Isolation

- Thin interface is proxy for “less complexity to exploit”

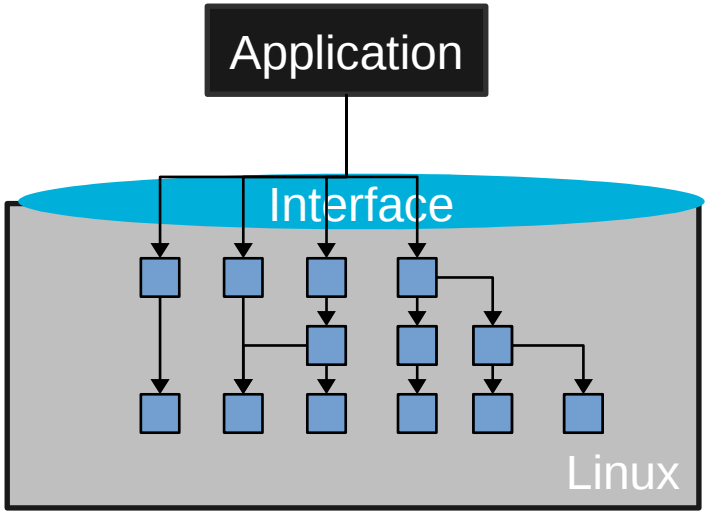


Metric for Isolation

- Thin interface is proxy for “less complexity to exploit”

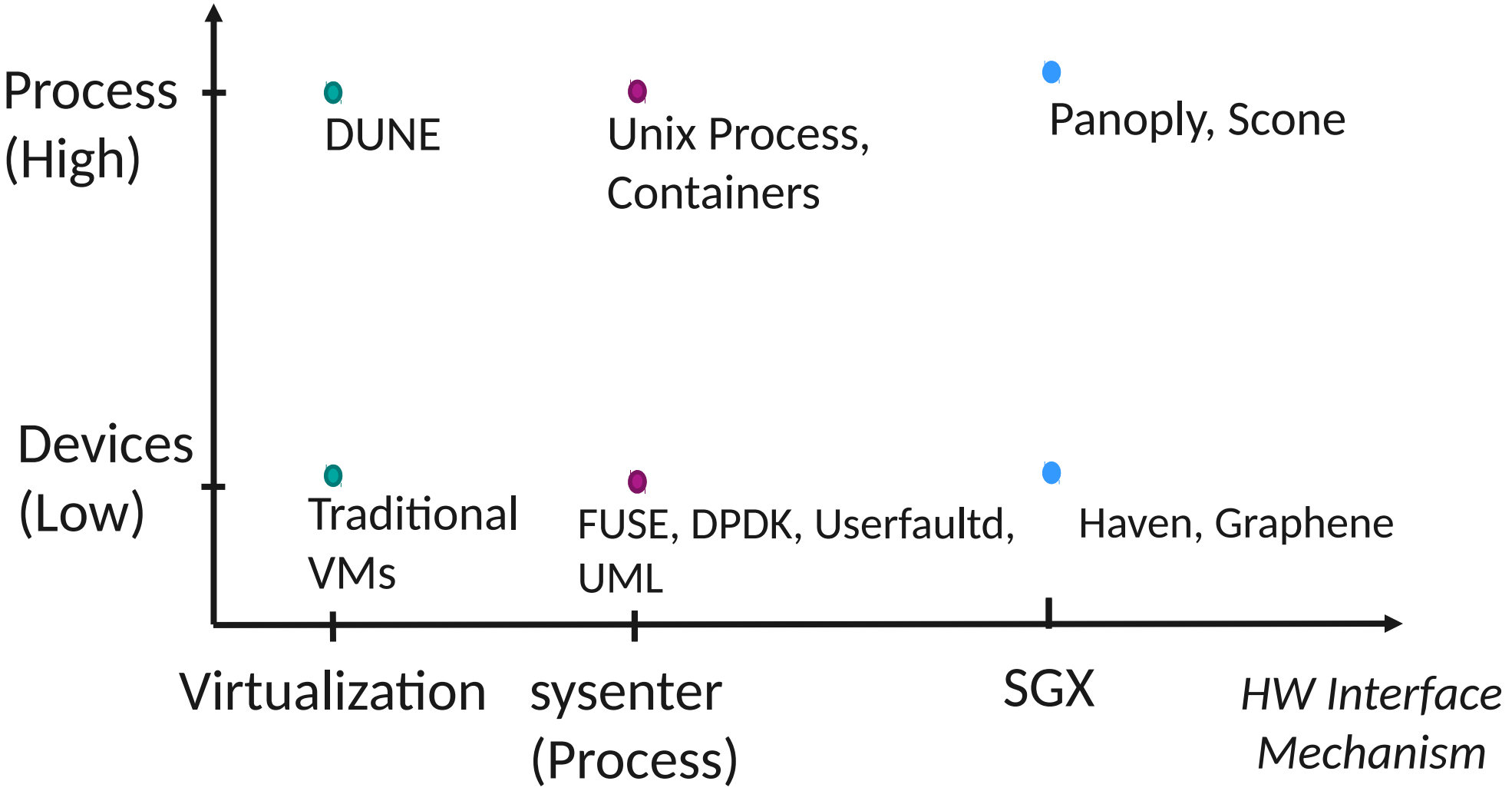


- Want to measure how much of the kernel is exposed
  - Kernel function tracing (ftrace)

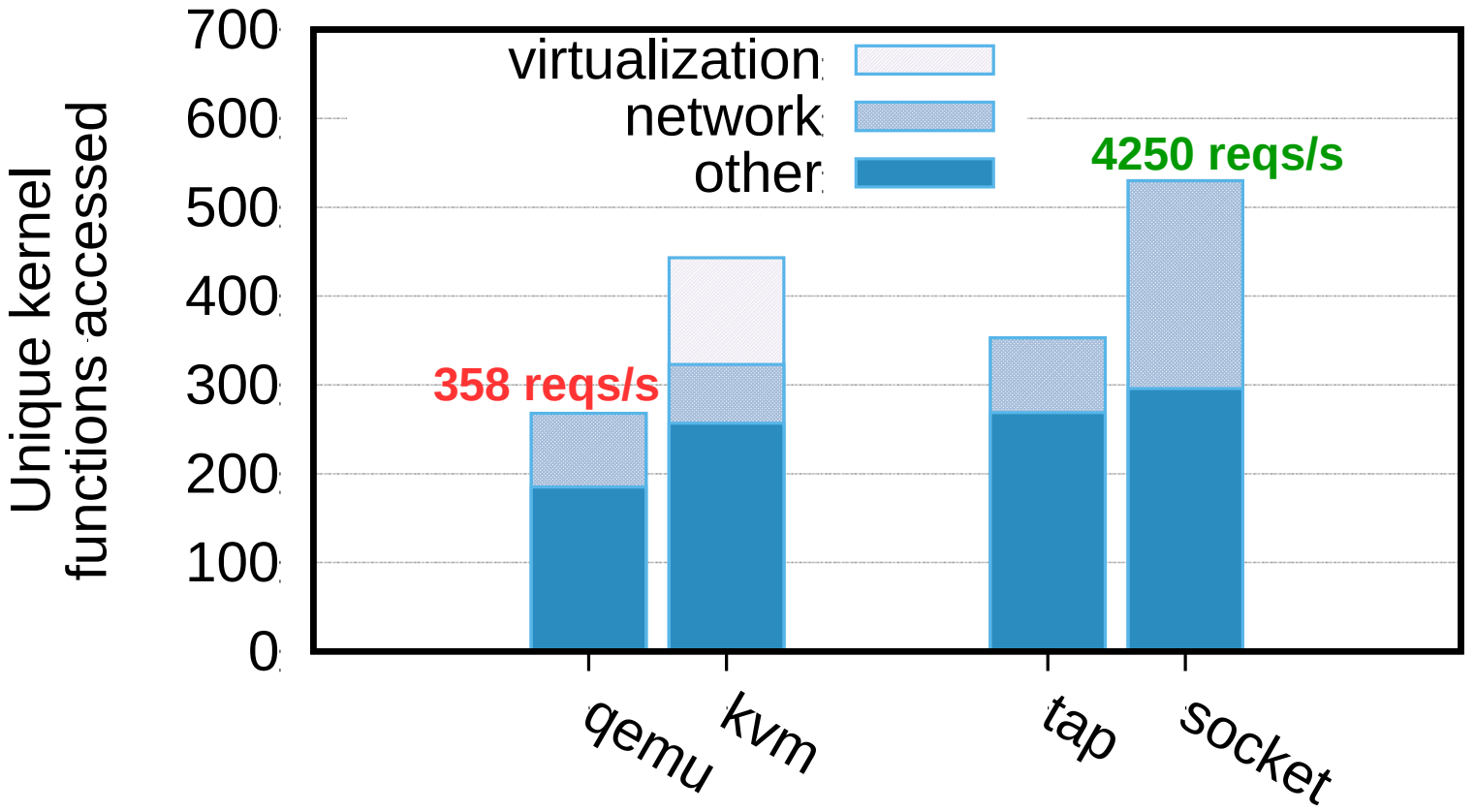


13 kernel functions

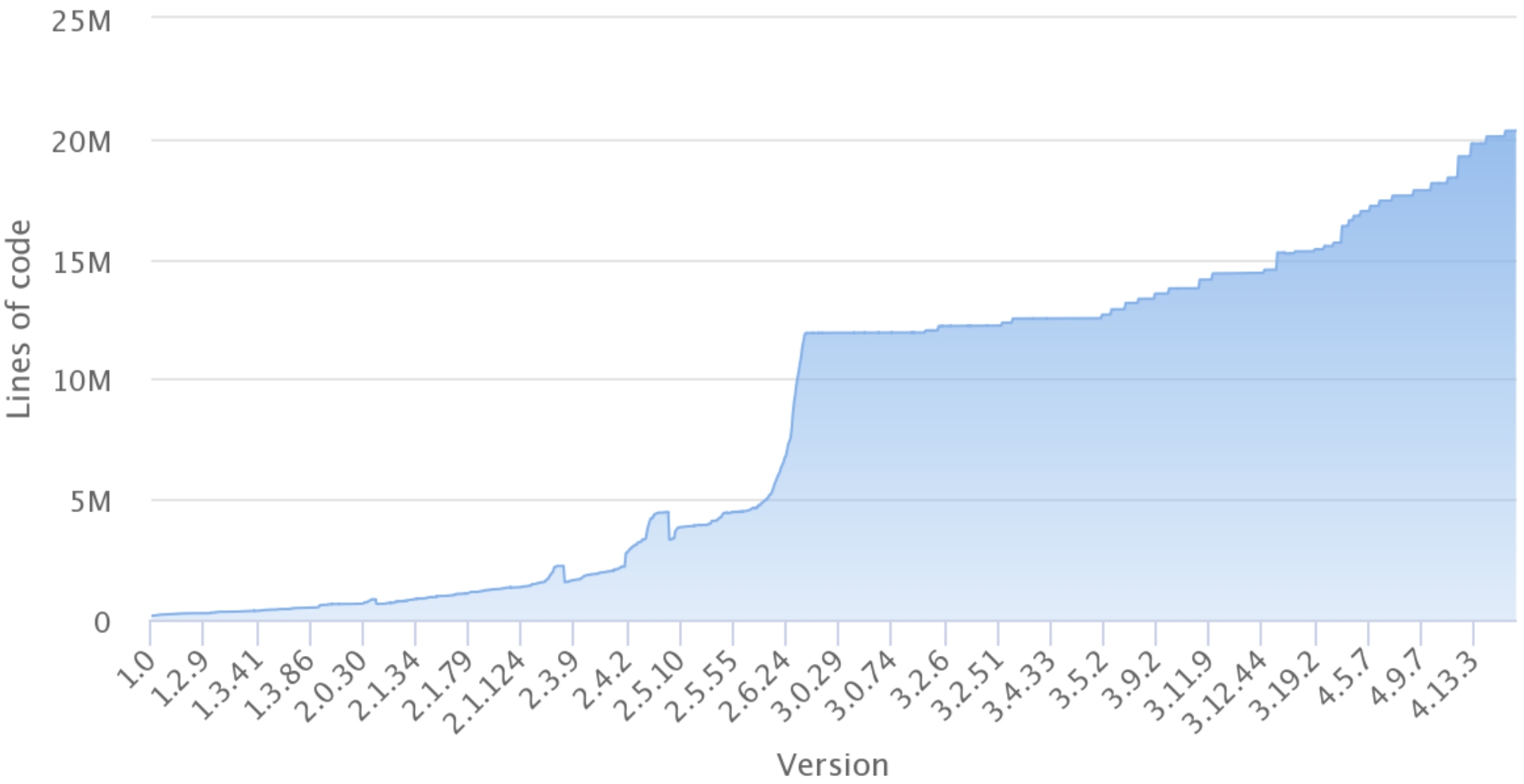
Interface level ≠ mechanism



Dangers of virtualization mechanisms

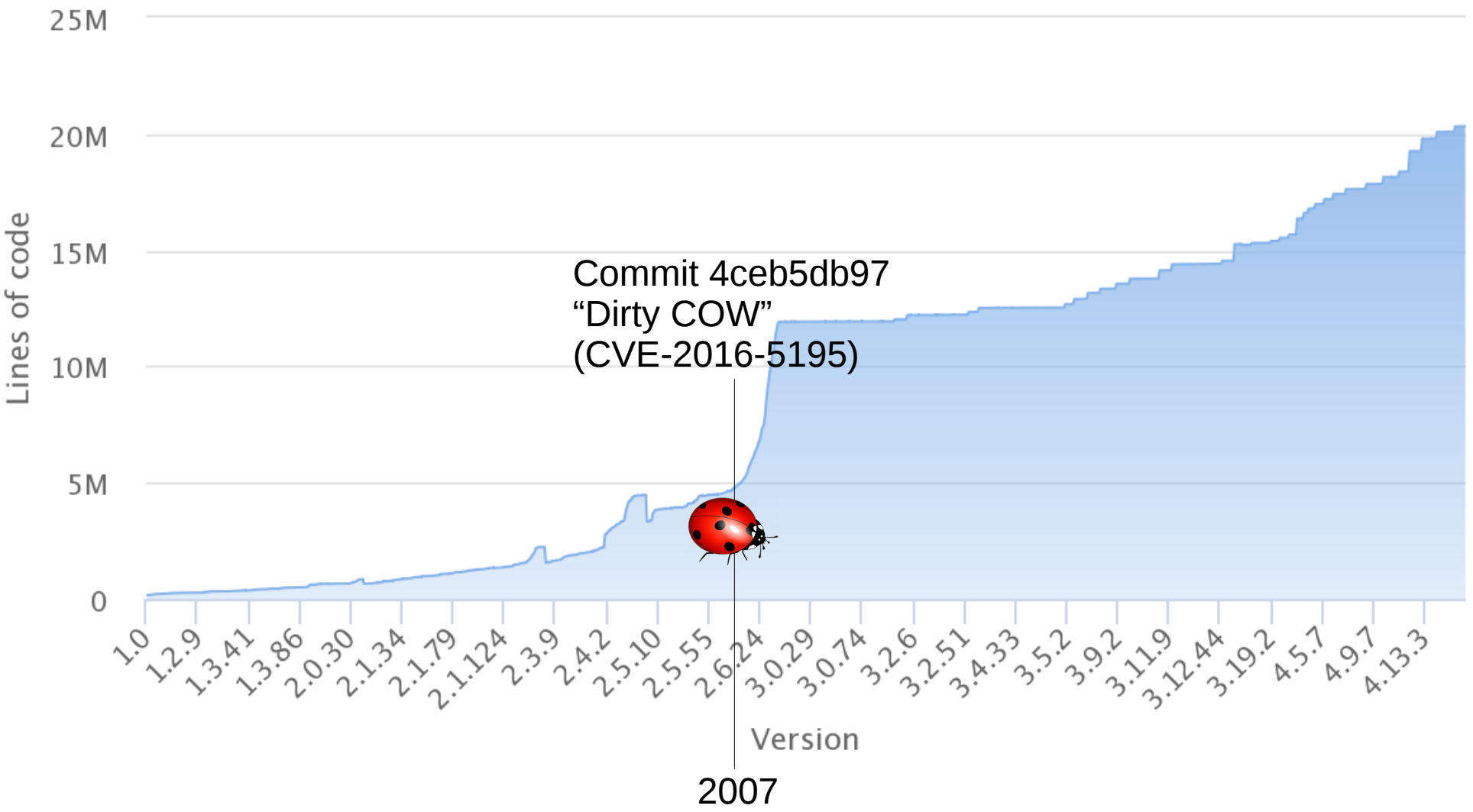


Linux Kernel's LOC increase



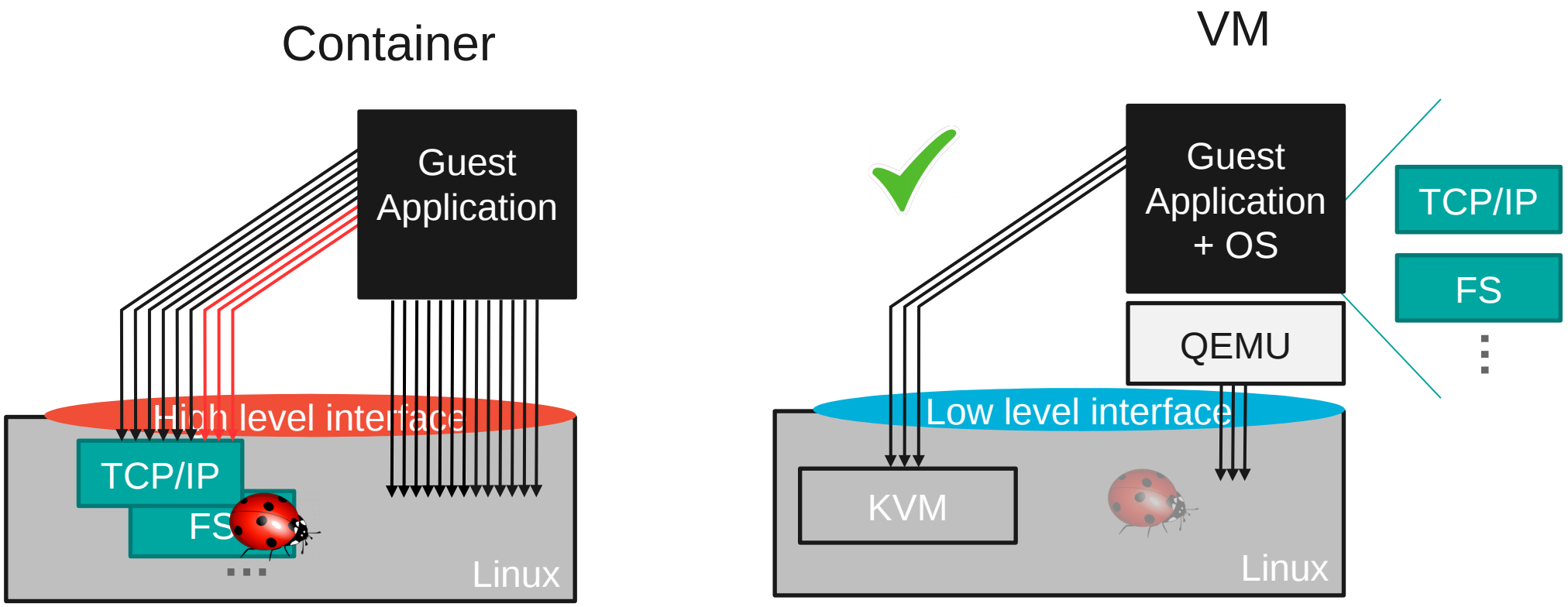
https://www.linuxcounter.net/statistics/kernel

Linux Kernel's LOC increase

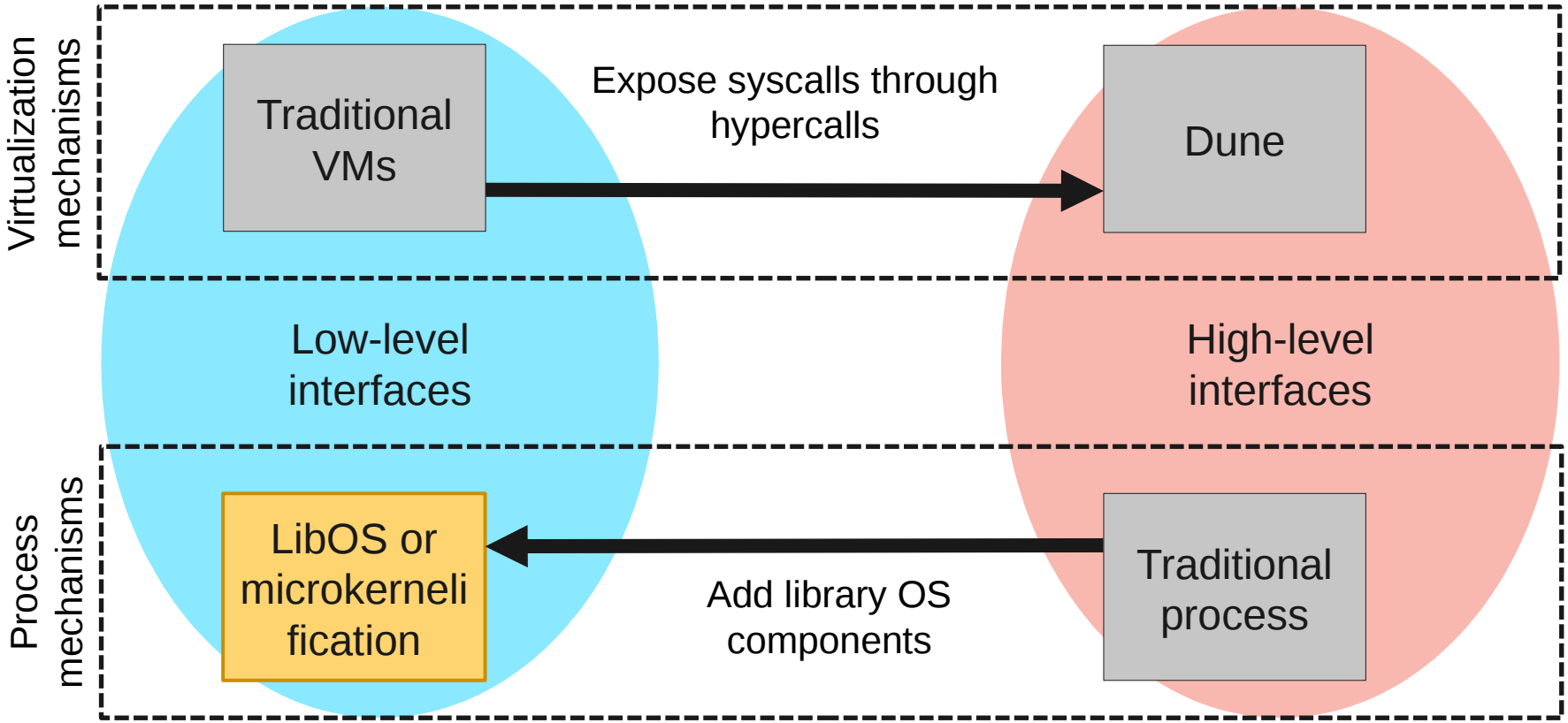


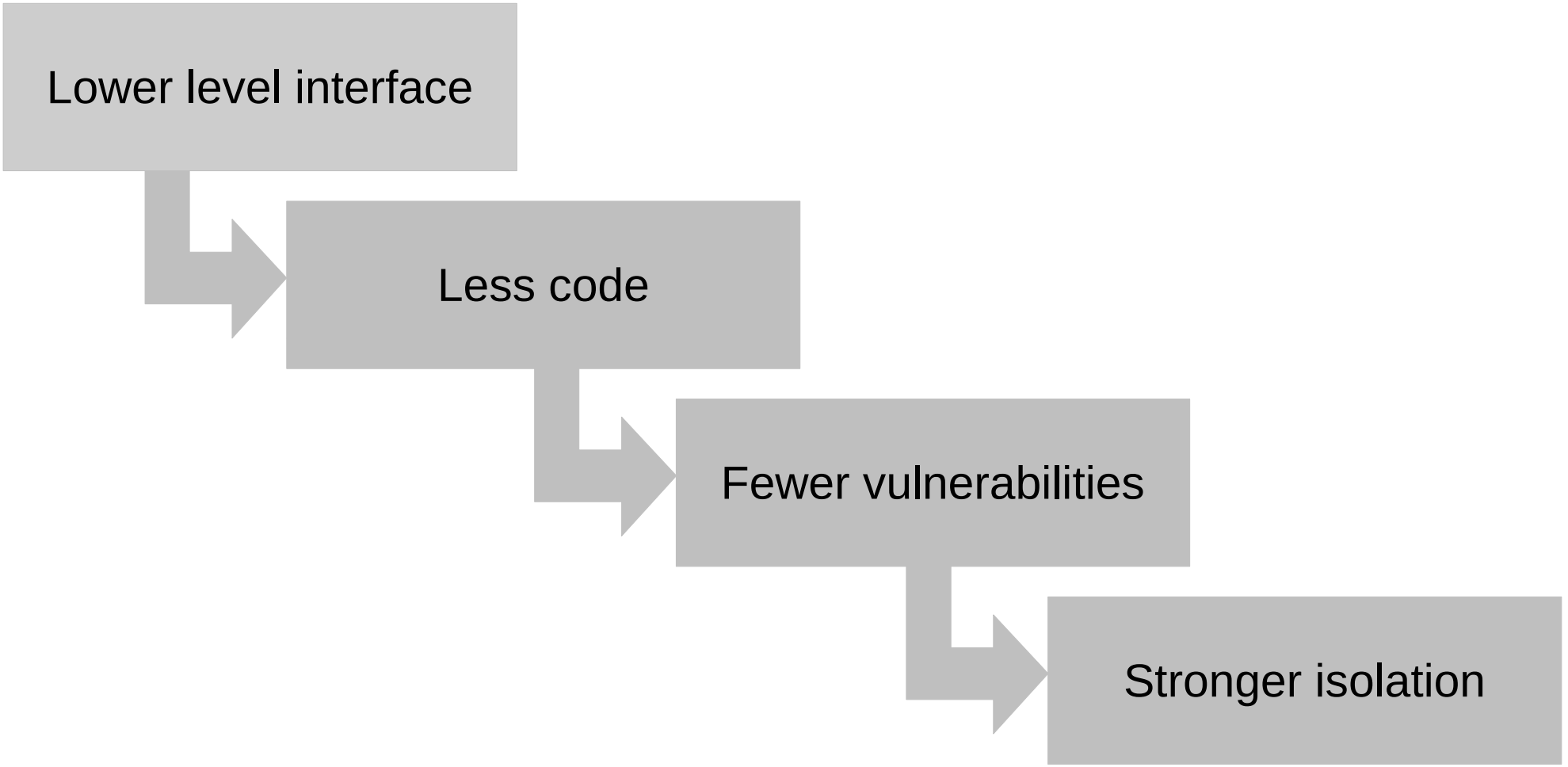
https://www.linuxcounter.net/statistics/kernel

Containers vs VMs for isolation?

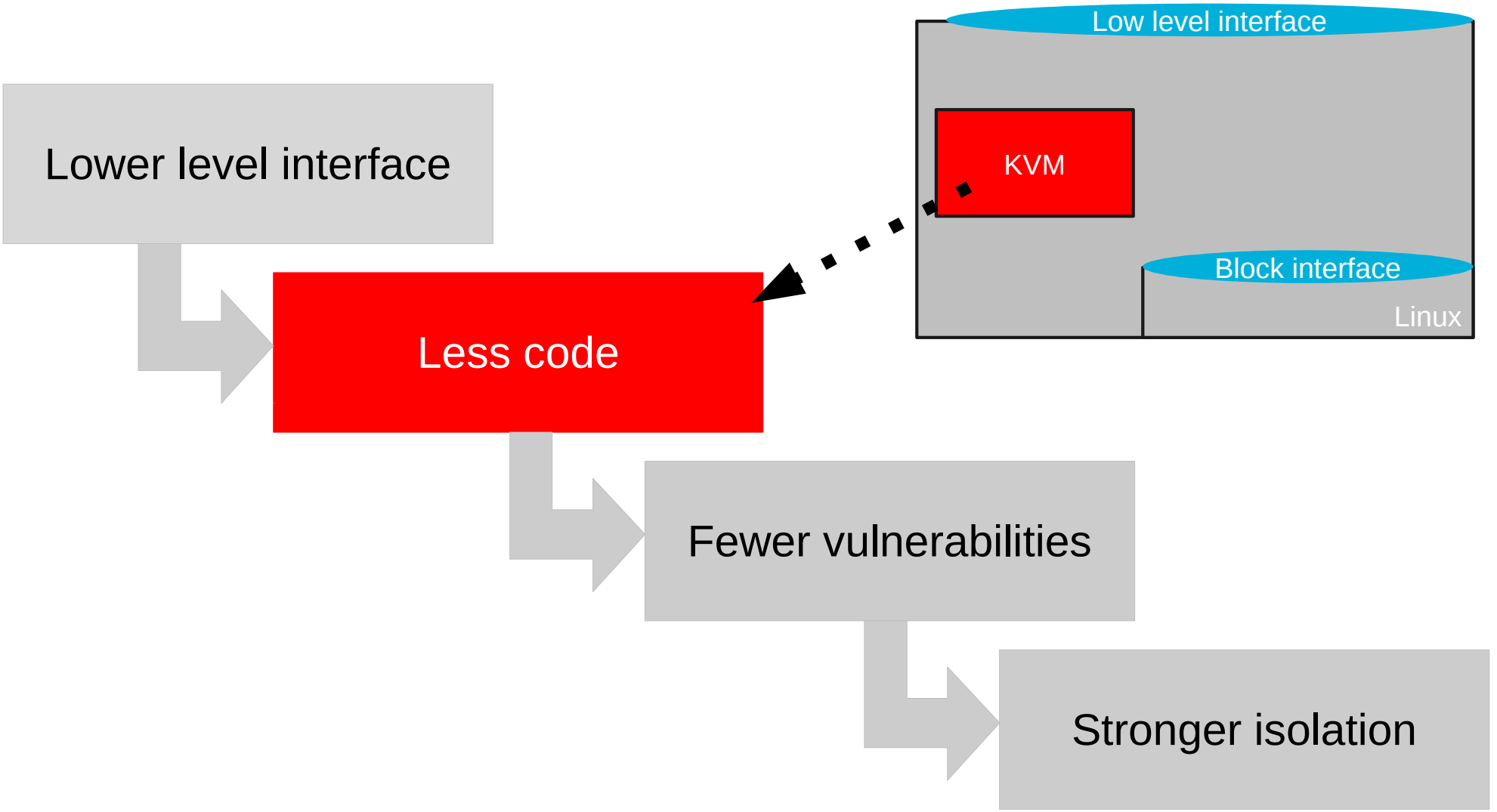


Interface level ≠ mechanism



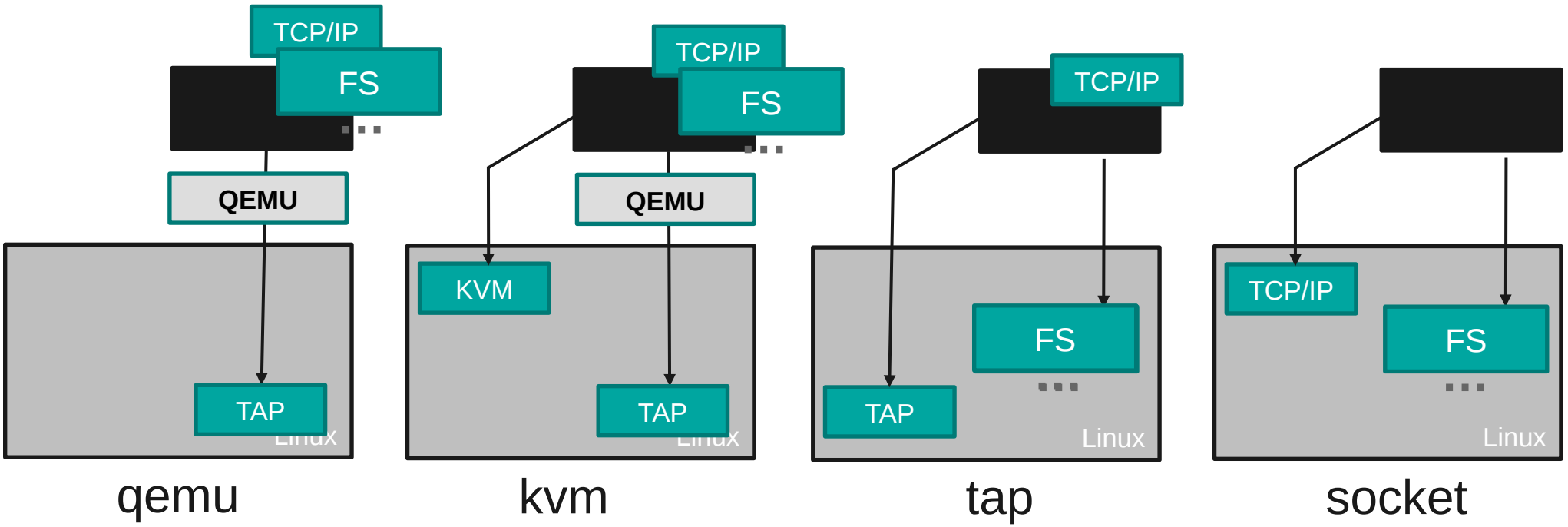


- The level of interface has nothing to do with virtualization

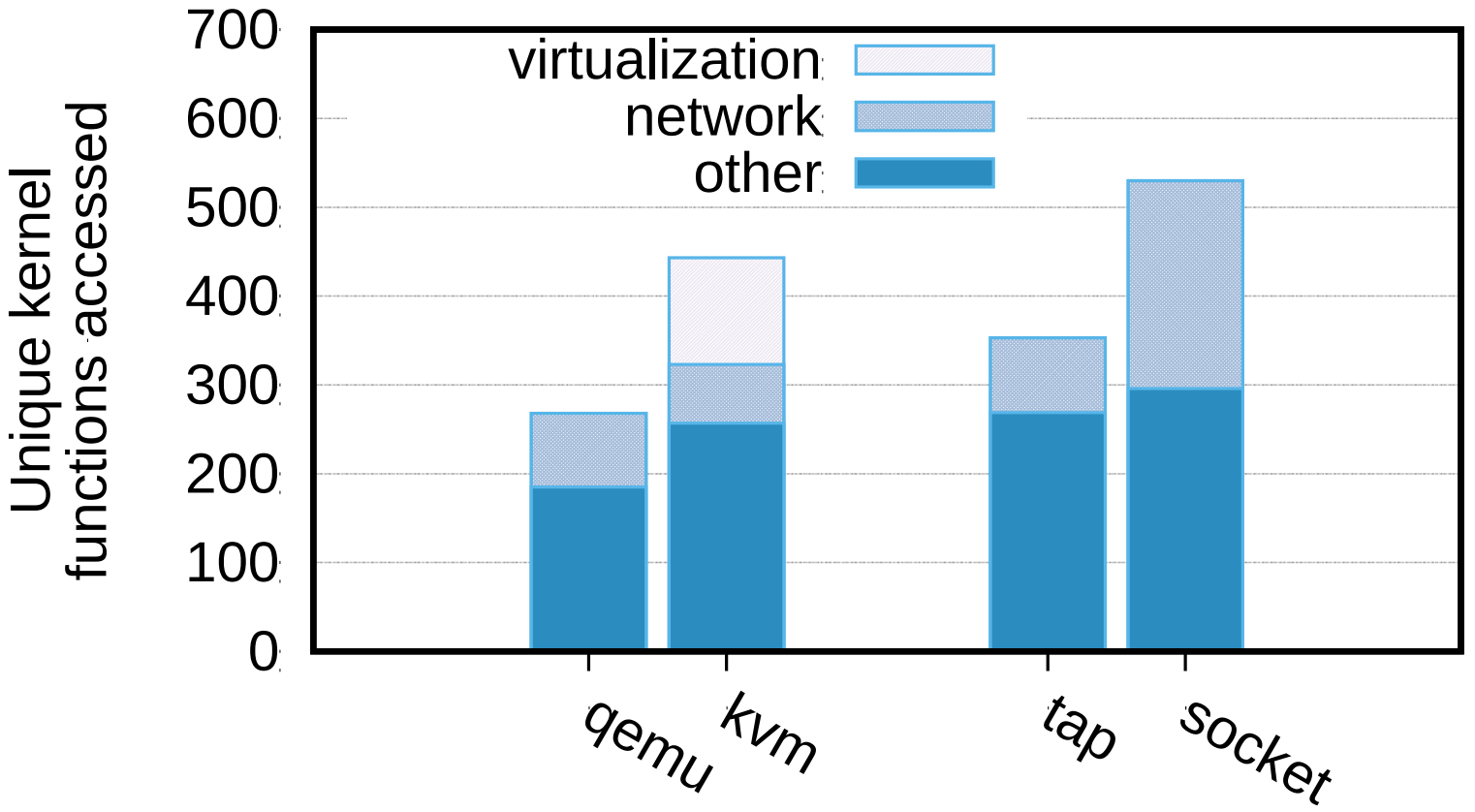


- Virtualization adds code and complexity

Comparing all options



Dangers of virtualization mechanisms



- qemu is the best you can do by taking stuff out of the kernel