RVISOR

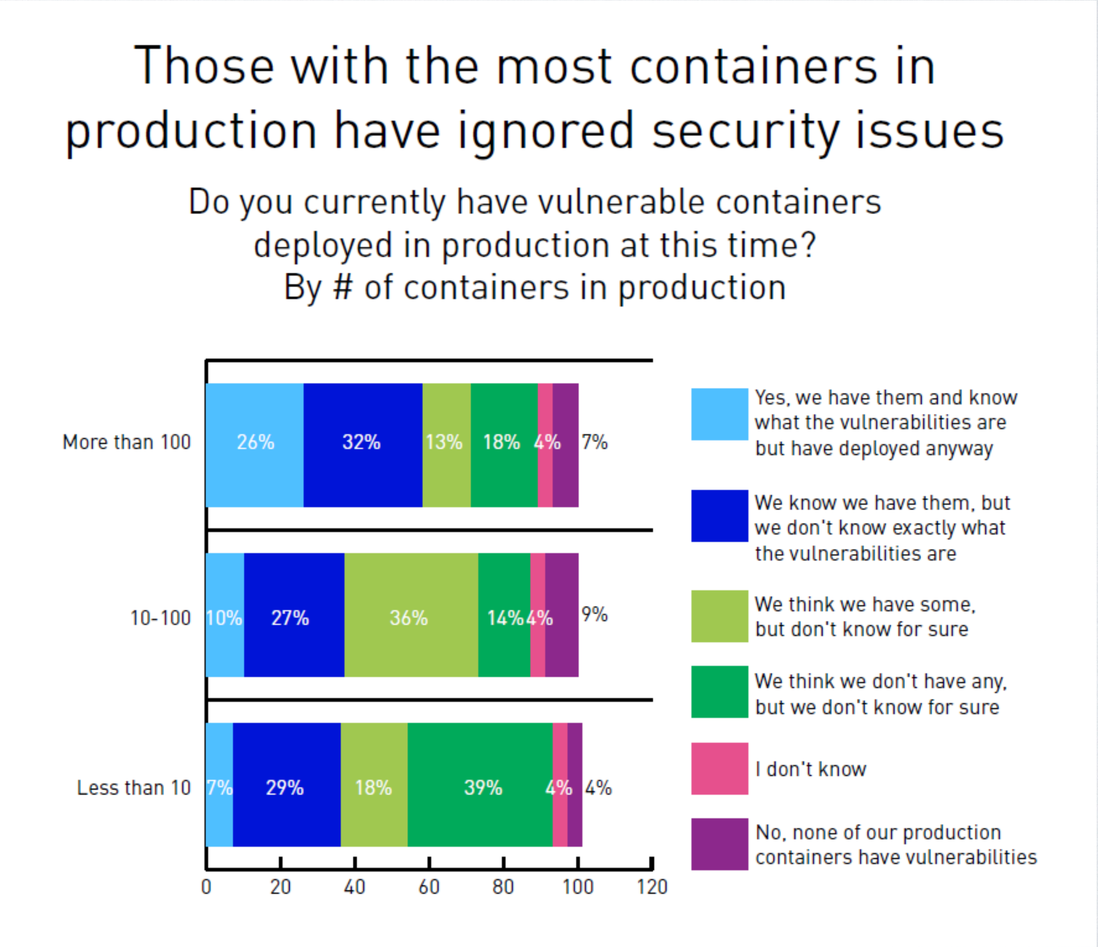
Yet Another Efficient Approach for Container Security

Midterm Presentation - Chital Group

Part I - Container Security

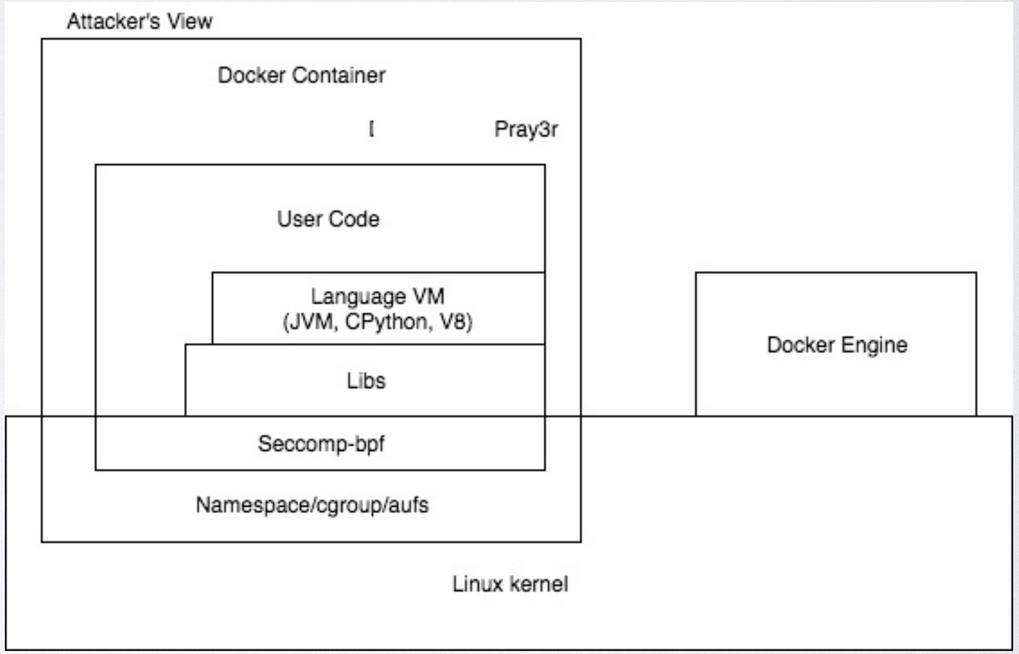
The Limitation of Traditional Container - Security

- Along with the development of container tech, security is becoming an important matter.
- According to a report from Tripwire, about half of the companies surveyed, especially those that run a deployment of more than 100 containers, believe that their containers have security vulnerabilities.
- Tripwire's survey also point out: 42 percent of respondents said that their company has delayed or limited container adoption because of security concerns.



Why Do Containers Lack Security ?

- Namespace/cgroup: Provided by Linux Kernel
- Linux isolation problems in it cannot be eradicated and may worsen due to some newly added features.
- Attacker may exploit Linux kernel vulnerabilities to break out of the container.

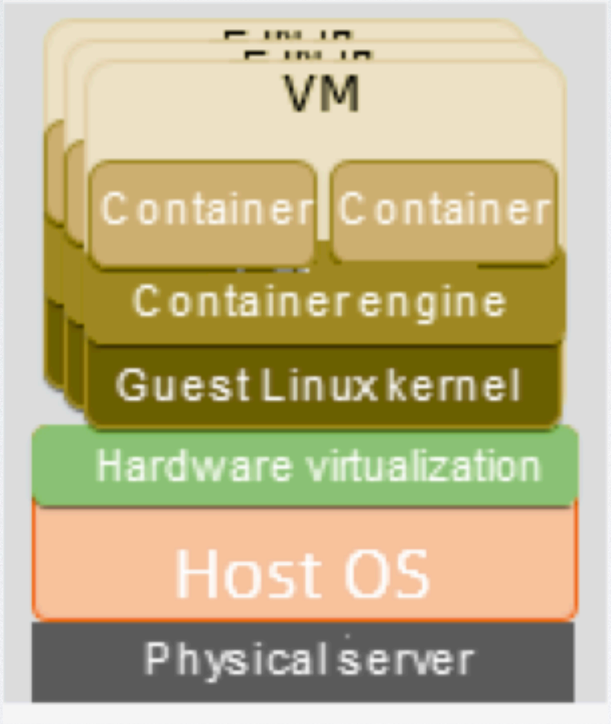


Is Solving all Security Bugs of Linux Kernel Possible ?

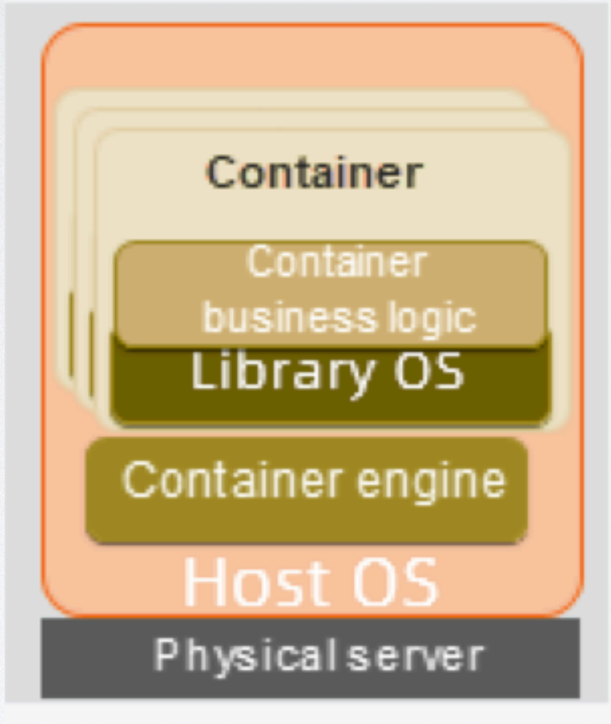
- “It’s always really hard to try to get rid of unnecessary fat, because as every developer knows, things tend to grow …”
- “only real solution is to admit that bugs happen, and try to mitigate them by having multiple layers of security, so that if you have a hole in one component, the next component will catch it.”

— By Linus Torvalds (LinuxCon)

Add Multiple Layers - Solutions Before 2018

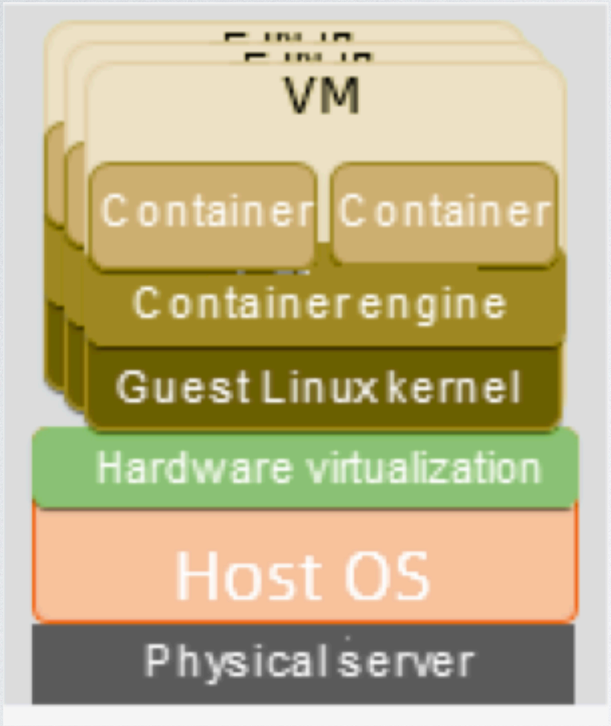


Traditional VM

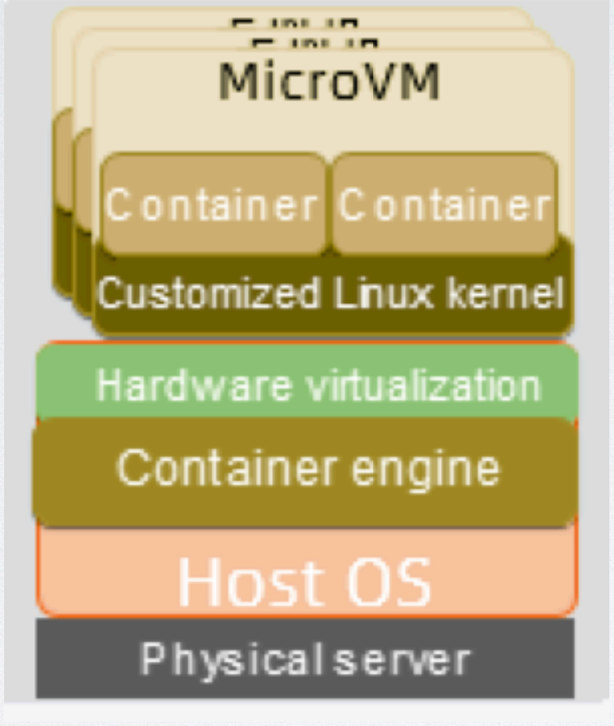


Unikernel

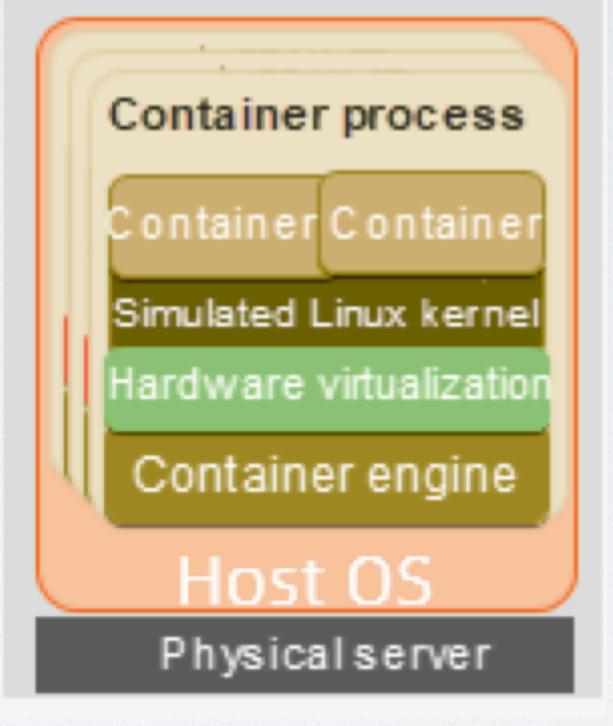
Add Multiple Layers - Solutions After 2018



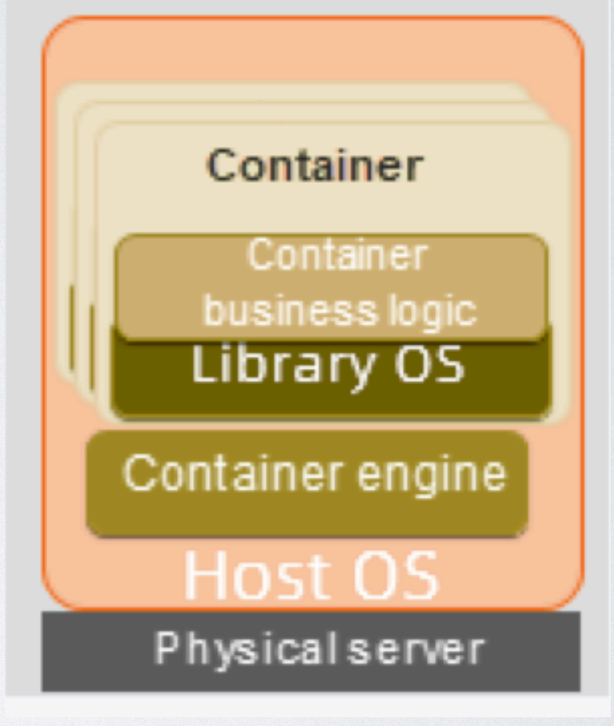
Traditional VM



MicroVM



Process Virtualization



Unikernel

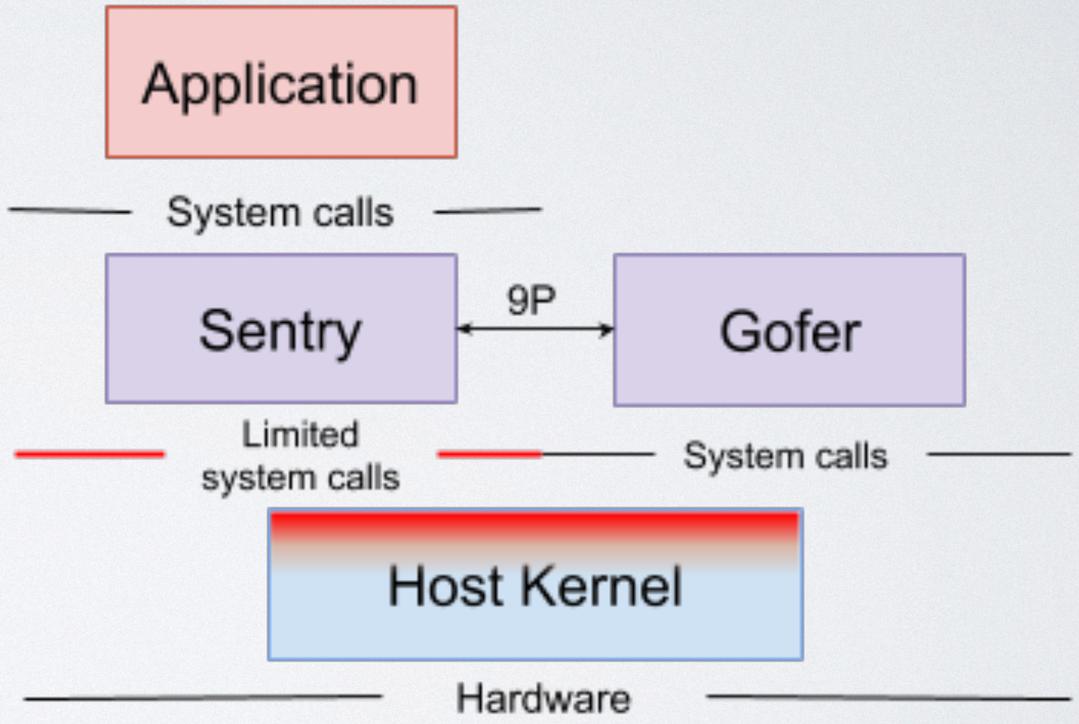
Implements of These Solution

Traditional VM	MicroVM	Process Virtualization	Unikernel
• KVM • Xen	• Kata Container • Firecracker • WSL2	• gVisor • WSL	• Nabla Container • …

Part II - gVisor Container Sandbox

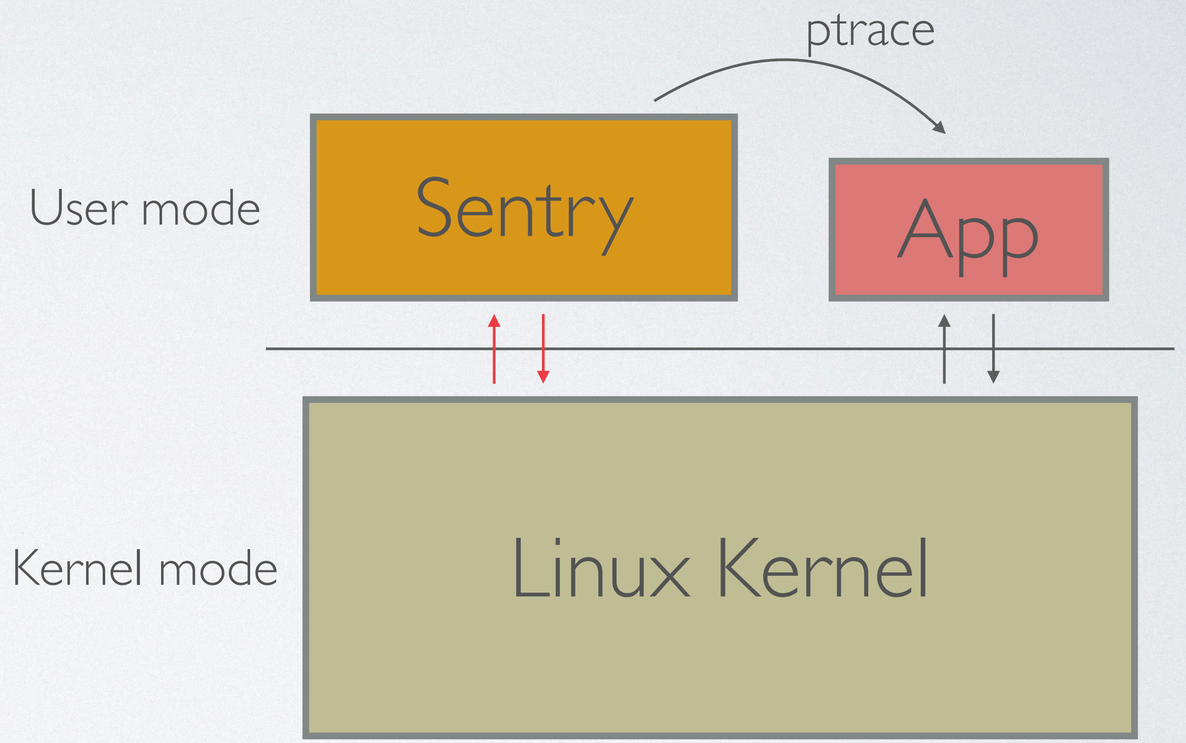
gVisor - Architecture

- The Sentry, A user-space kernel that runs the container and intercepts and responds to system calls made by the application.
- A Gofer which provides file system access to the container.
- gVisor written in Go, which is a safer language with GC.



gVisor - Limitations

- one more context switch between User mode and Kernel mode, and two context switch between two different process.



gVisor -Benchmark

- These are benchmark conducted by Xu Wang & Fupan Li (Kata Container)
- Test Nginx : ab -n 50000 -c 100 http://10.100.143.131:8080/

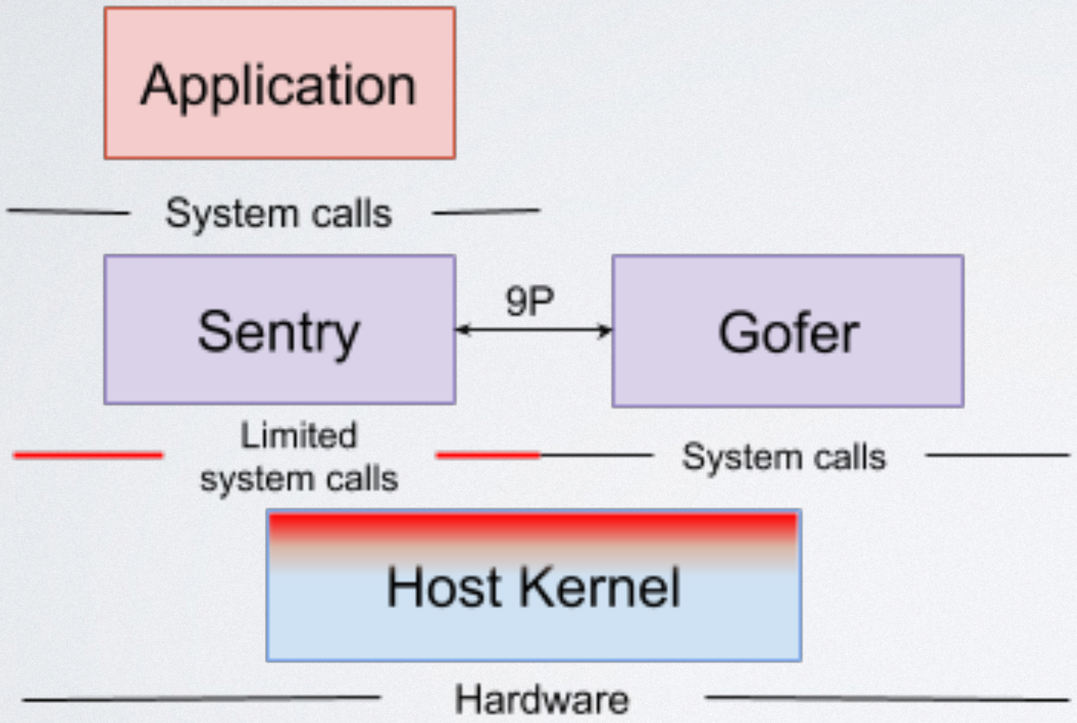
	runC	Kata*	gVisor		
Concurrency Level	100	100	100		
Time taken for tests	3.455	3.439	161.338	seconds	
Complete requests	50000	50000	50000		
Failed requests	0	0	0		
Total transferred	42250000	42700000	42250000	bytes	
HTML transferred	30600000	30600000	30600000	bytes	
Requests per second	14473.73	14541.18	309.91	[#/sec]	(mean)
Time per request	6.909	6.877	322.677	[ms]	(mean)
Time per request	0.069	0.069	3.227	[ms]	(mean, across all concurrent requests)
Transfer rate	11943.65	12127.12	255.73	[Kbytes/sec]	received

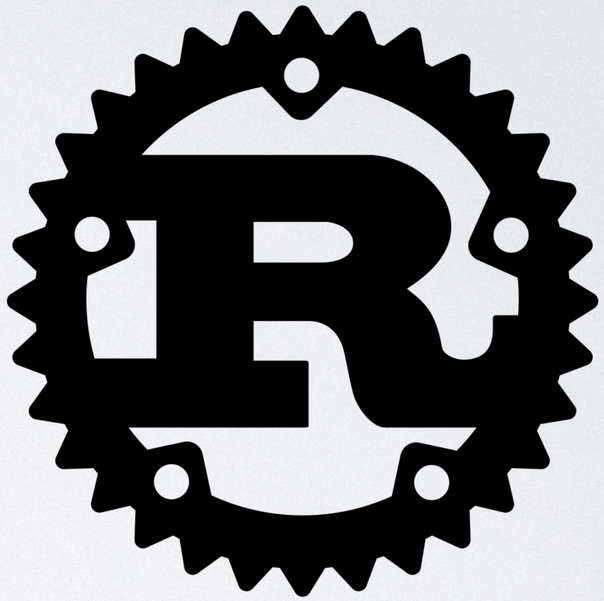
MicroVM VS gVisor

- MicroVM(Like Kata):
  - Advantages: The speed of containers, The security of VMs
  - Disadvantages: More Boot Time than runC
- gVisor
  - Advantages: Fast boot performance and low memory footprint
  - Disadvantages: Time Performance for IO, compatibility for special use

Part III - our Attempt — rVisor

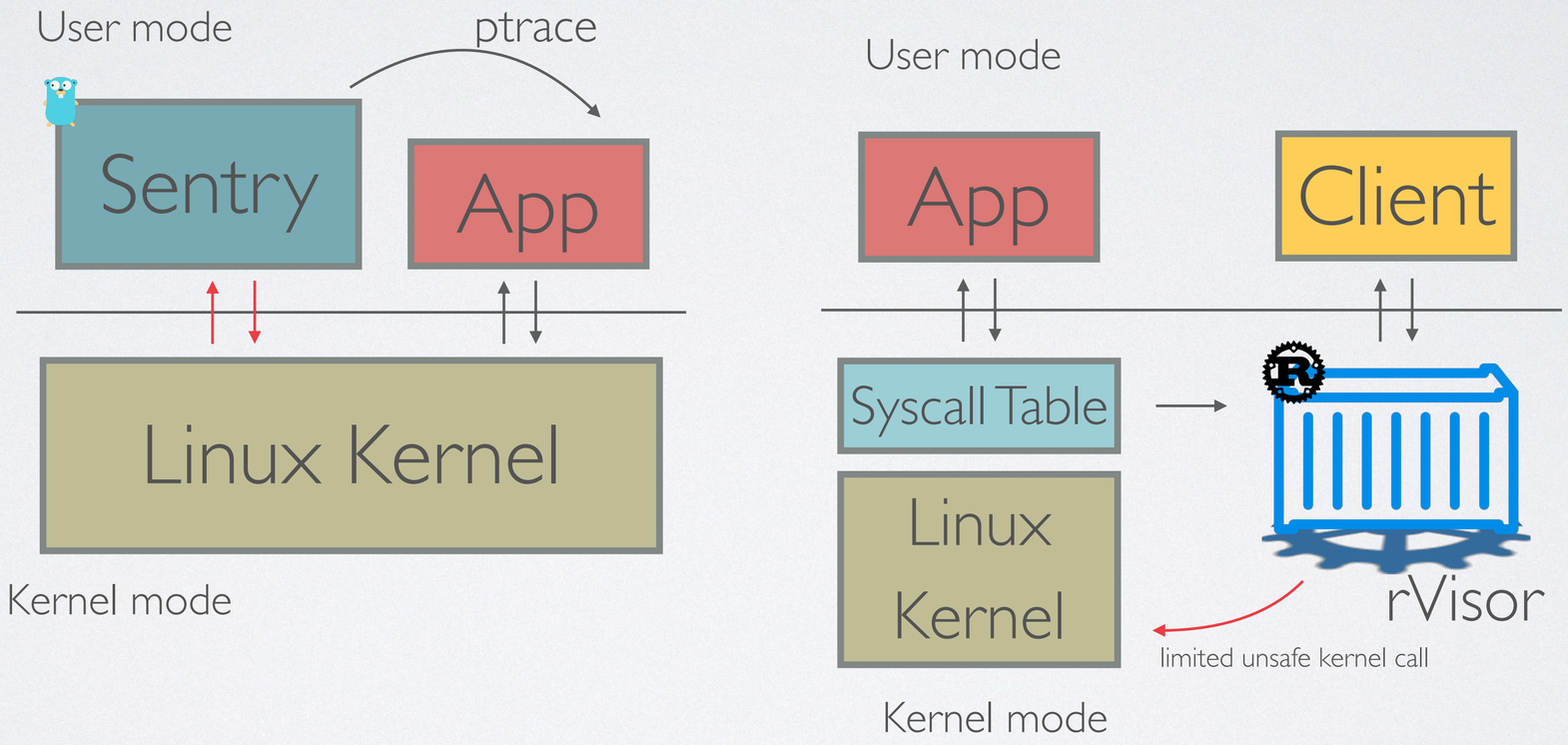
gVisor Security — Rust Safety





keyword unsafe

An Attempt To Solve gVisor's Performance Problem



So We Have …

rVisor

=

gVisor

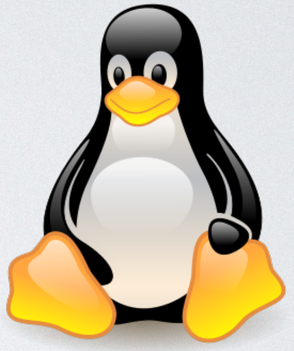
Process Virtualization

+

Rust

Limit the Unsafe

+

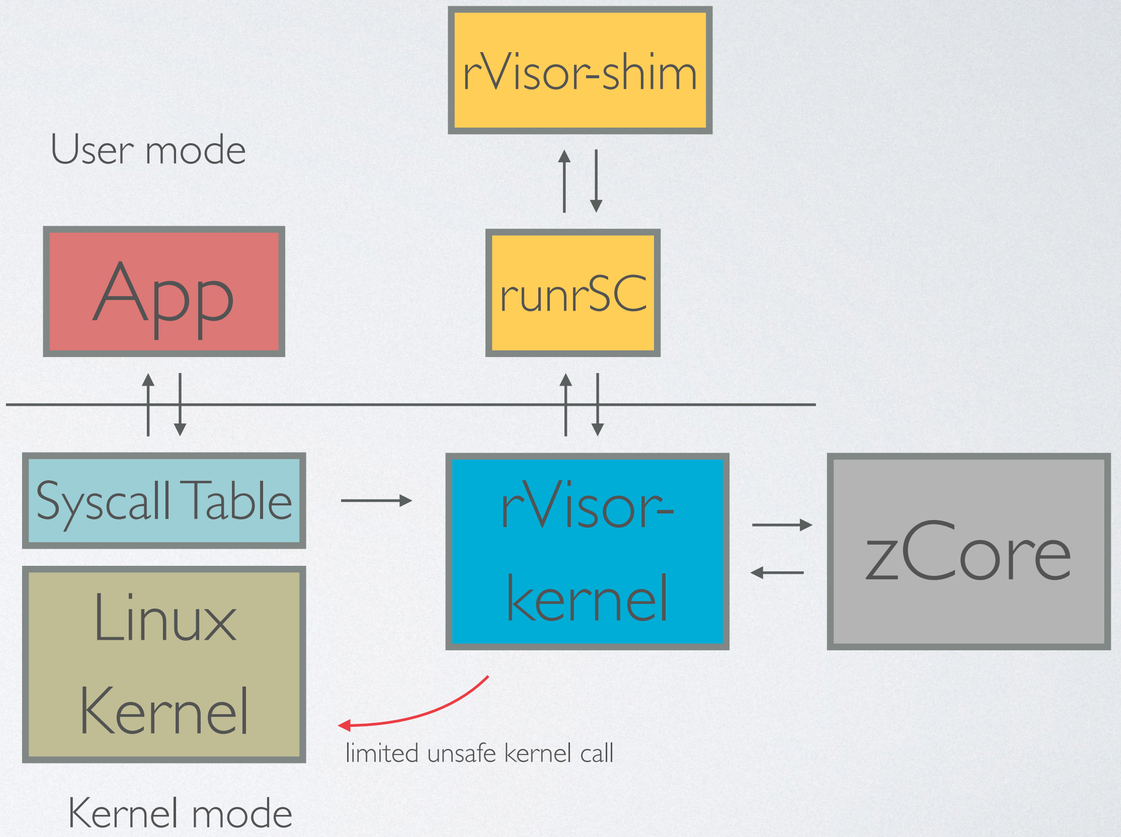


Linux Kernel

Inside the Kernel

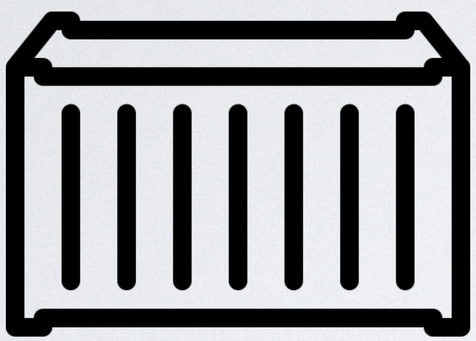
rVisor - Architecture

- rVisor-ptrace sentry : a debugger for rVisor-kernel sentry.
- rVisor-kernel sentry : main component of rVisor
- runrSC : like runC/runSC a controller process of the container
- rVisor-shim : An adapter to containerd.



Part IV - our Workflow

Our Workflow



rVisor-ptrace

50%

transplant

rVisor-kernel (experimental)

0%

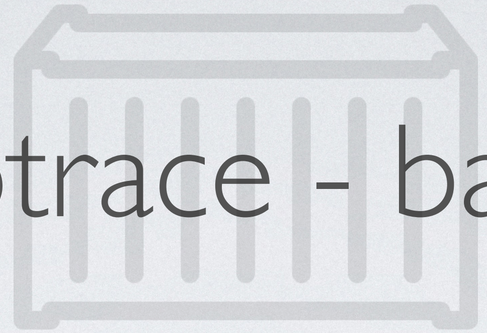
improve

(if possible)

rVisor-kernel

0%

rVisor-ptrace - basic idea



- Ptrace (two choice)
  - ptrace(PTRACE_SYSCALL, …) (ptrace::syscall in Rust)
  - ptrace(PTRACE_SYSEMU …) (ptrace::sysemu in Rust)
- As rVisor-ptrace just for debuging, we use ptrace(PTRACE_SYSCALL, …)
- Then we can build event loop by waitpid - ptrace(PTRACE_SYSCALL)

rVisor-ptrace - in general

- According to our target - implement chroot first.
- To implement chroot, we need to translate guest path to the hosts os or other file for special use. (completed)
- Then for almost every syscalls, change their path to host os, or provide special file they want.

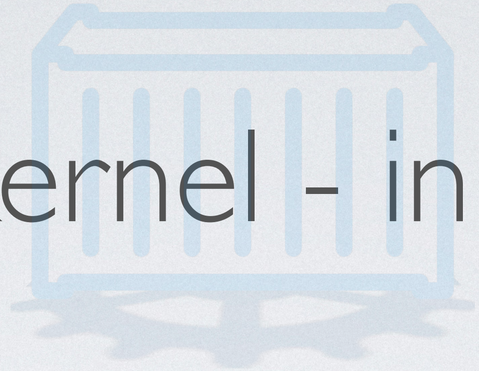
rVisor-ptrace - what we expect

- At Least
  - Provide file for simple application like ls, stat, ps. (dylib, /usr/lib/locale)
  - Run a shell safely that writes by our own. (lab2)
  - Use file/network IO to test the performance of rVisor-ptrace
- if possible:
  - We will test rVisor-ptrace in Nginx
- complete week 10-11.

rVisor-kernel - basic

- Run rust in Linux Kernel Module (2 ways)
  - Use Library to write safe rust
  - Directly interact with C using FFI
- We will directly interact with C using FFI.
- Then we use rust core library to build kernel module.
- syscall_table is provided by linux kernel, we can modify it directly.

rVisor-kernel - in general



- We should limit the use of unsafe.
- We will provide some API for rVisor-runrSC to use.
- We will complete runrSC and rVisor-shim, at same time.

rVisor-kernel - Team Collaboration

- rVisor-kernel sentry: 丁垣天 郑在一
- rVisor-runrSC: 何灏迪
- rVisor-shim: 叶之帆
- documentation: rust-doc

rVisor-kernel - what we expect

- At Least
  - It have same functionality as rVisor-ptrace.
  - We could do some benchmark on it. (same as rVisor-ptrace)
  - Complete runrSC as our client
- If Possible
  - Complete rVisor-shim, and use rVisor on containerD.
- complete week 13-14

rVisor-kernel - next step

- gVisor is much larger. gVisor implemented a minimal Linux kernel to ensure its compatibility and security.
- More supports for threads, memory management , etc.
- More benchmark, like Nginx, Redis, etc.

ABOUT CHITAL

- 丁垣天 dnailzb@outlook.com
- 叶之帆 rabbitforyou@foxmail.com
- 何灏迪 hardyhe2019@outlook.com
- 郑在一 donpanica@outlook.com


QUESTIONS ?