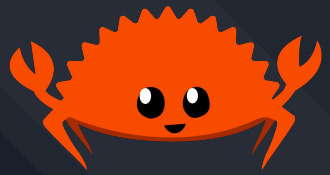
Rustifying the VM Introspection Ecosystem



FOSDEM 2020

Dorian Eikenberg Mathieu Tarral

Agenda

- What is VM Introspection ?
- VMI ecosystem today
- Rustifying the VM Introspection ecosystem
- Future work

Virtualization  Rust

- 2015:
  - Rust 1.0
- 2016:
  - rustyvisor
- 2017:
  - crosvm
  - Firecracker
- 2019:
  - rust-vmm
  - orange_slice
  - cloud-hypervisor

`Wenzel/awesome-virtualization`

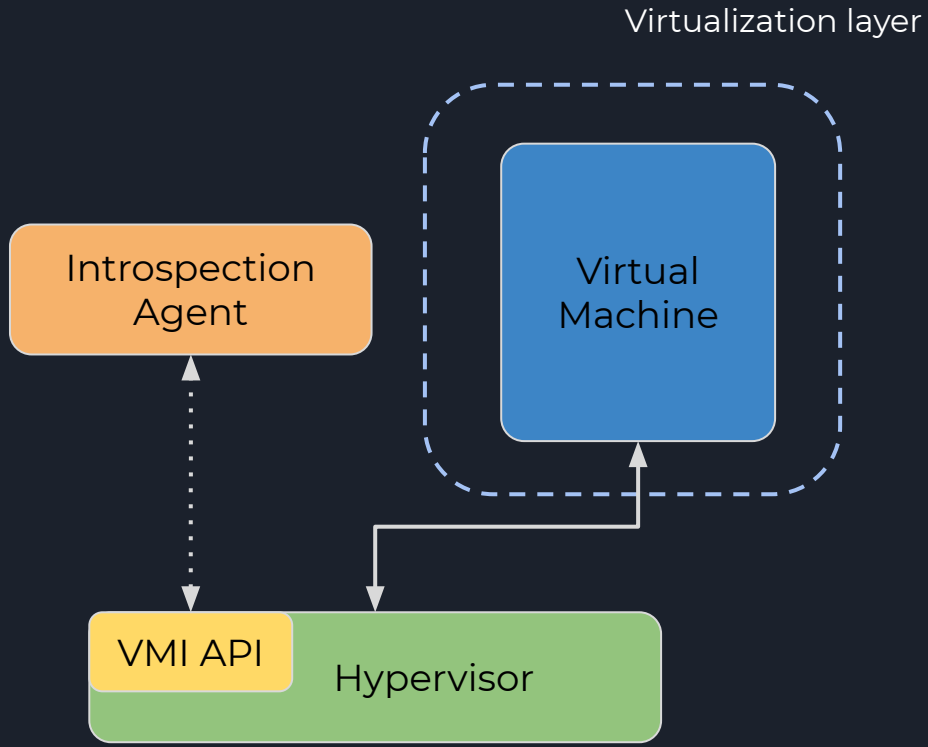
VM Introspection

VM Introspection

“Deriving the execution context of a virtual machine, from the hypervisor interface, by querying its hardware state, for security purposes”

VM Introspection : Concepts

- Intercept hardware events
  - memory access (r/w/x)
  - interrupts
    - set breakpoints ! (`int 3`)
  - MSR registers
  - control registers
  - etc...
- Modify hardware state
  - VCPUs registers
  - physical memory



VM Introspection : Core Strenghts

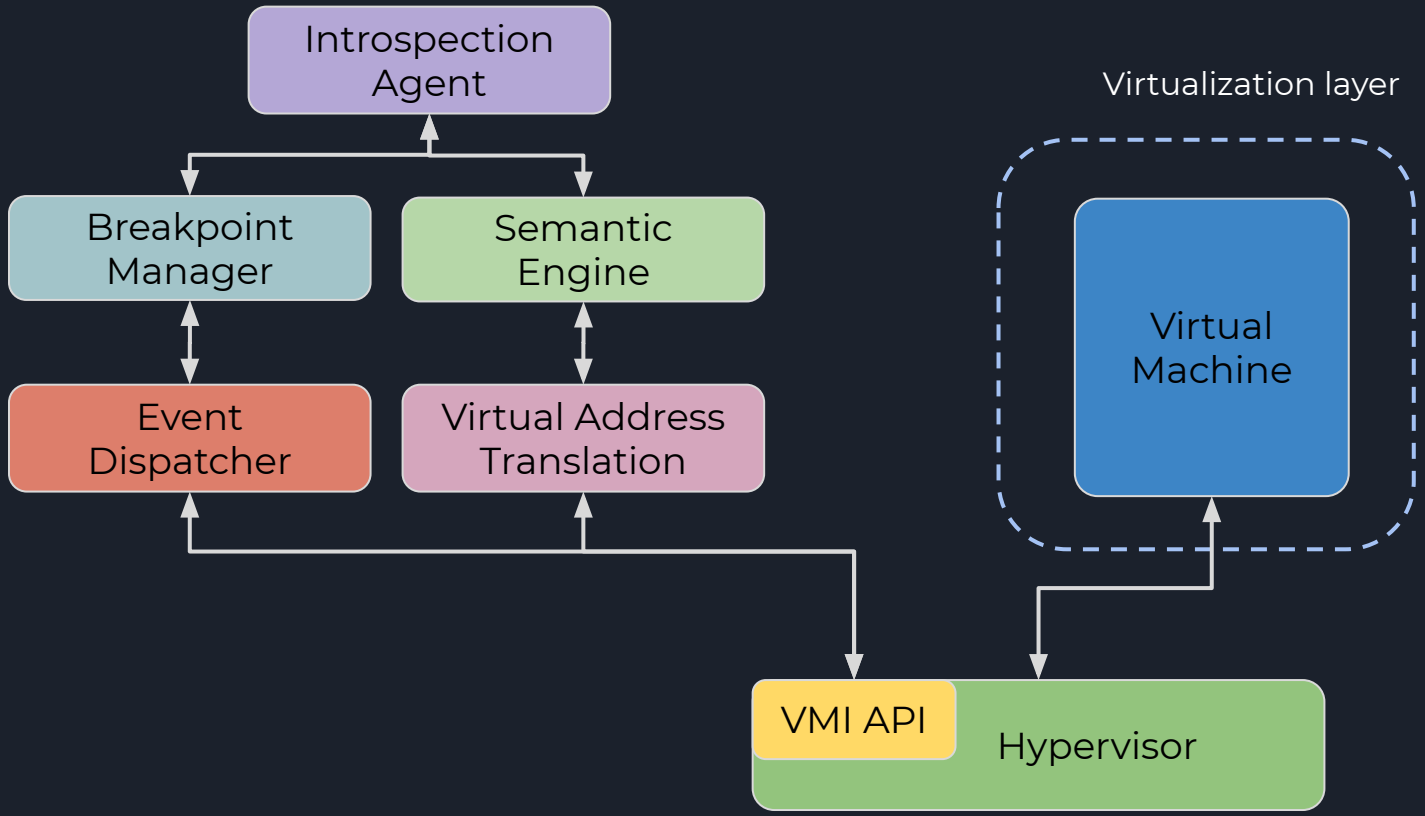
What VMI provides:

- VM hardware access
  - full system view at hypervisor-level privilege
- Interposition
  - control what hardware events to catch
  - manipulate what the OS should see of itself

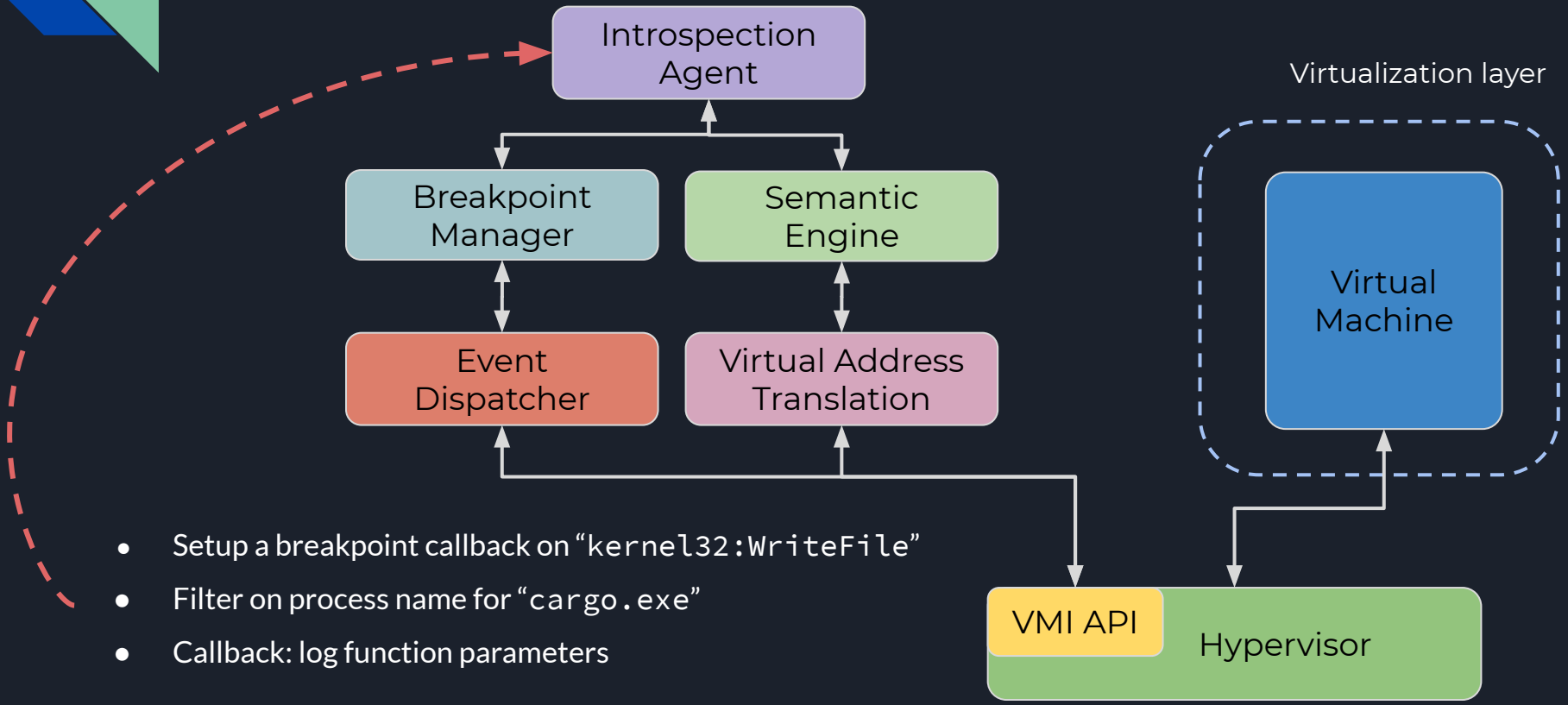
VM Introspection : Scenarios

- When detectability is an issue
  - stealth malware analysis
- Need a full-system approach
  - complex debugging scenarios (nested hypervisor)
  - advanced in-kernel fuzzing
- Can’t rely on guest OS
  - to give you a view of itself
  - assuming compromised kernel
  - Unikernel (!)

VM Introspection : Complexity

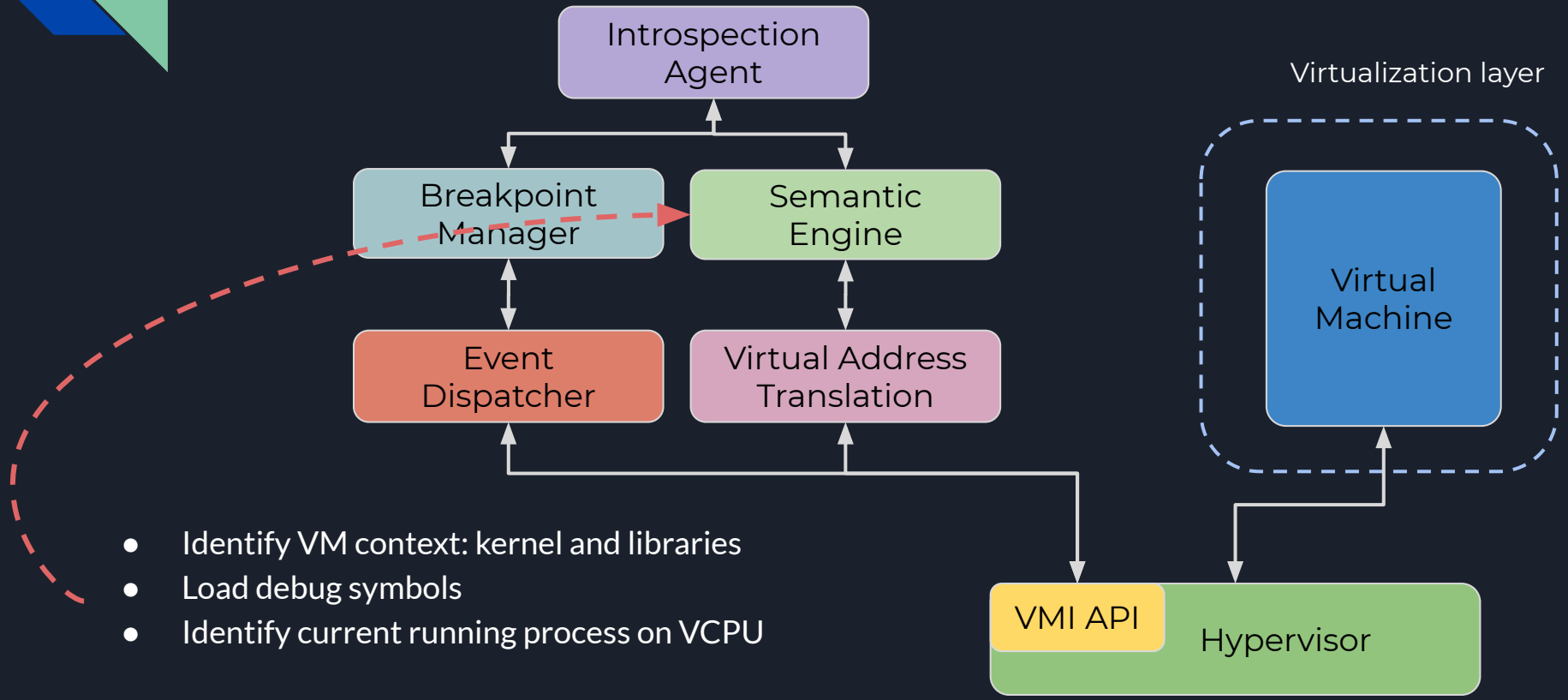


VM Introspection : Complexity



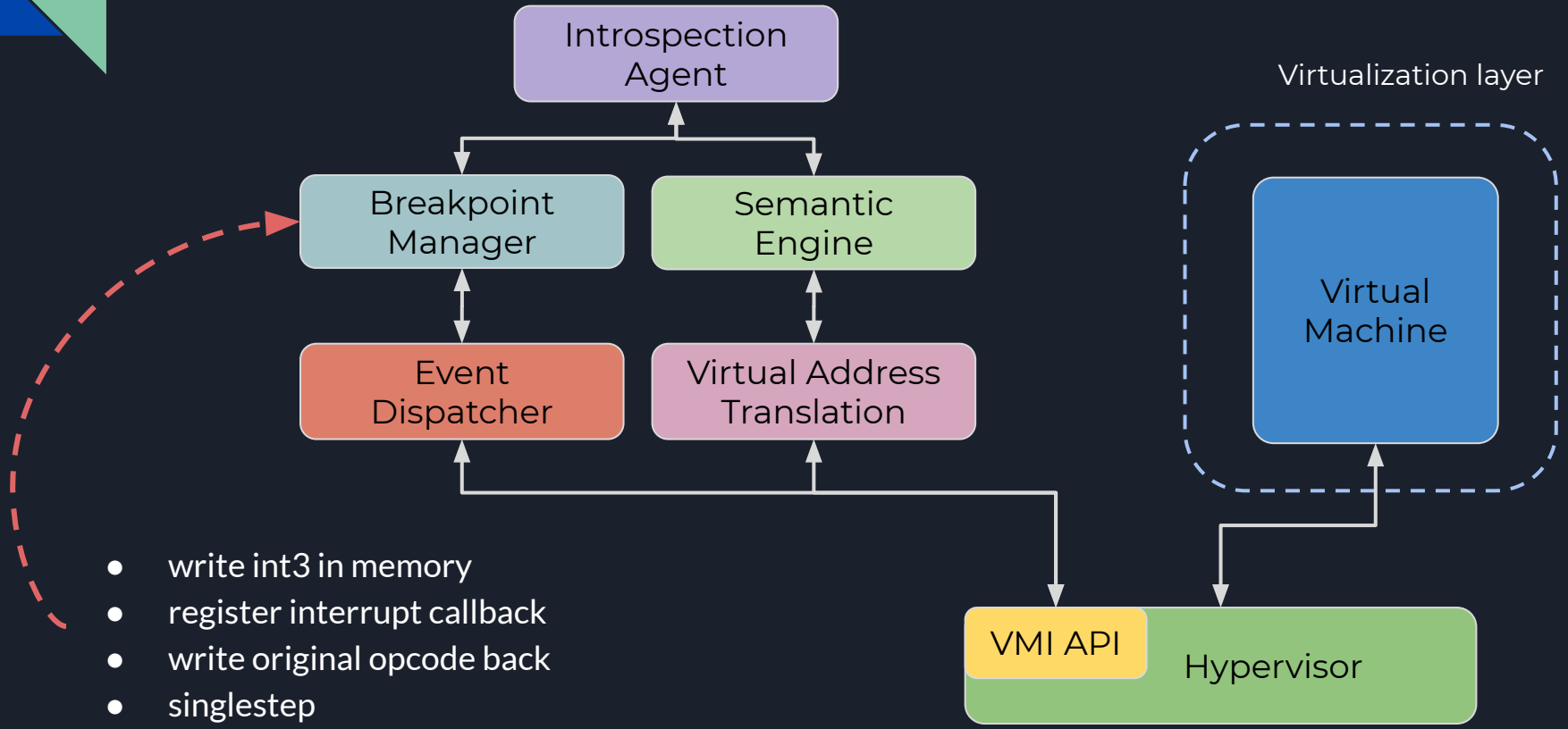
- Setup a breakpoint callback on “`kernel32:WriteFile`”
- Filter on process name for “`cargo.exe`”
- Callback: log function parameters

VM Introspection : Complexity



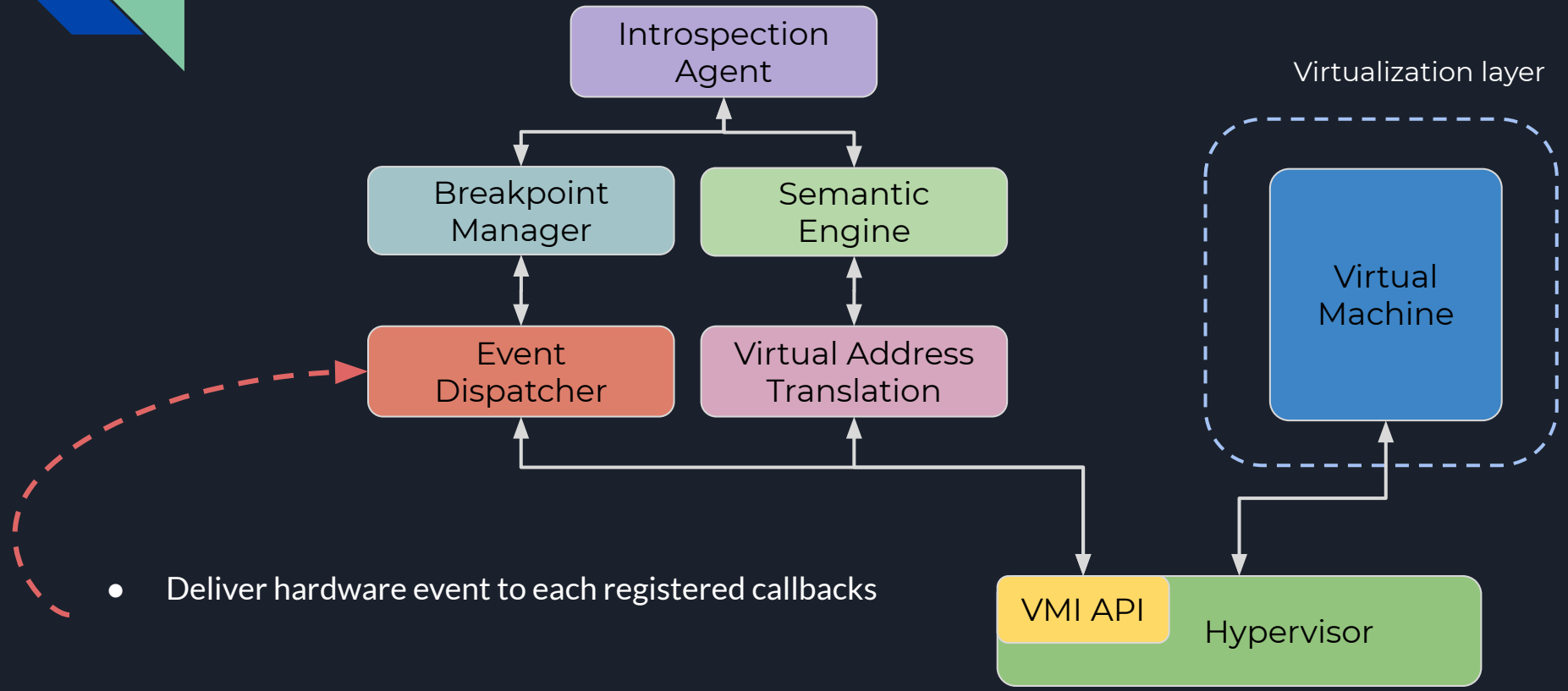
- Identify VM context: kernel and libraries
- Load debug symbols
- Identify current running process on VCPU

VM Introspection : Complexity



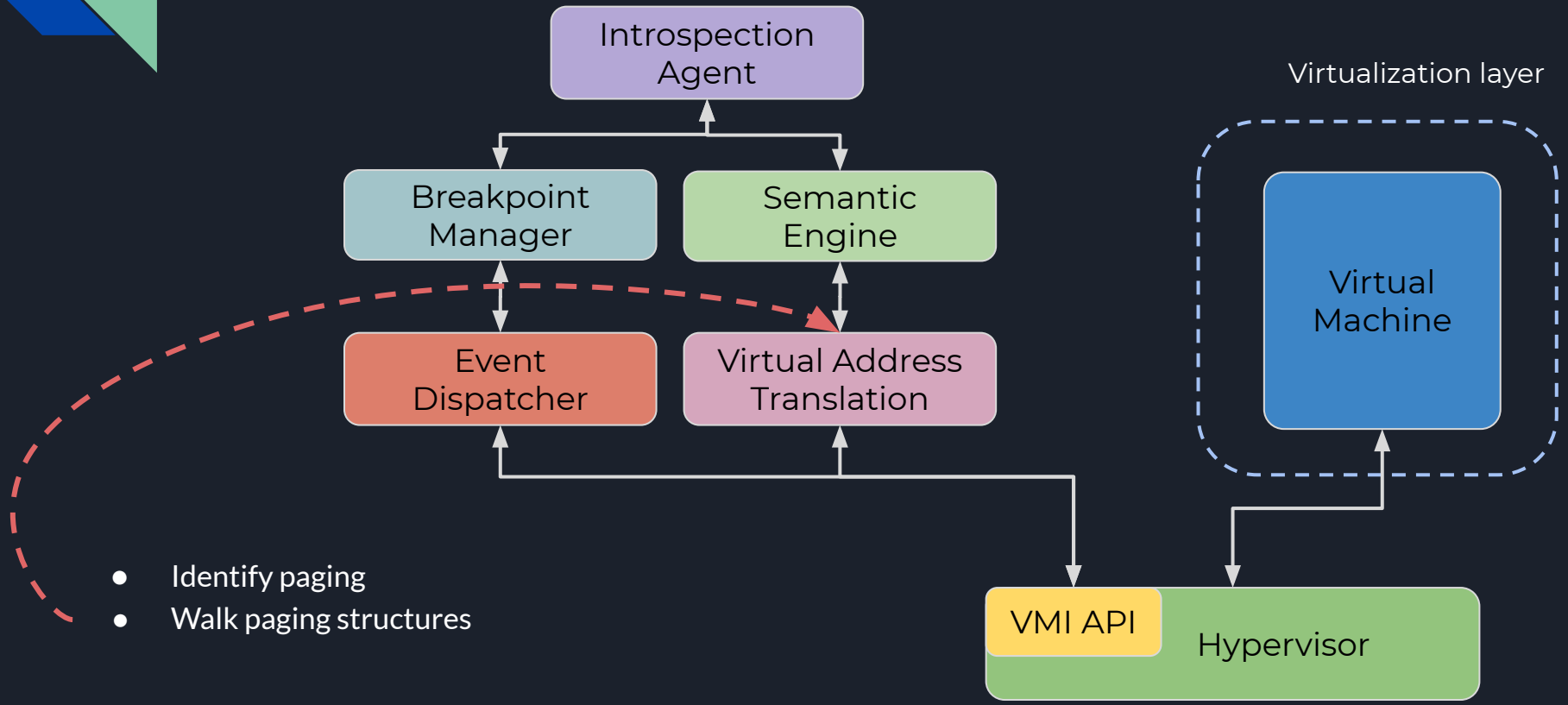
- write int3 in memory
- register interrupt callback
- write original opcode back
- singlestep

VM Introspection : Complexity



- Deliver hardware event to each registered callbacks

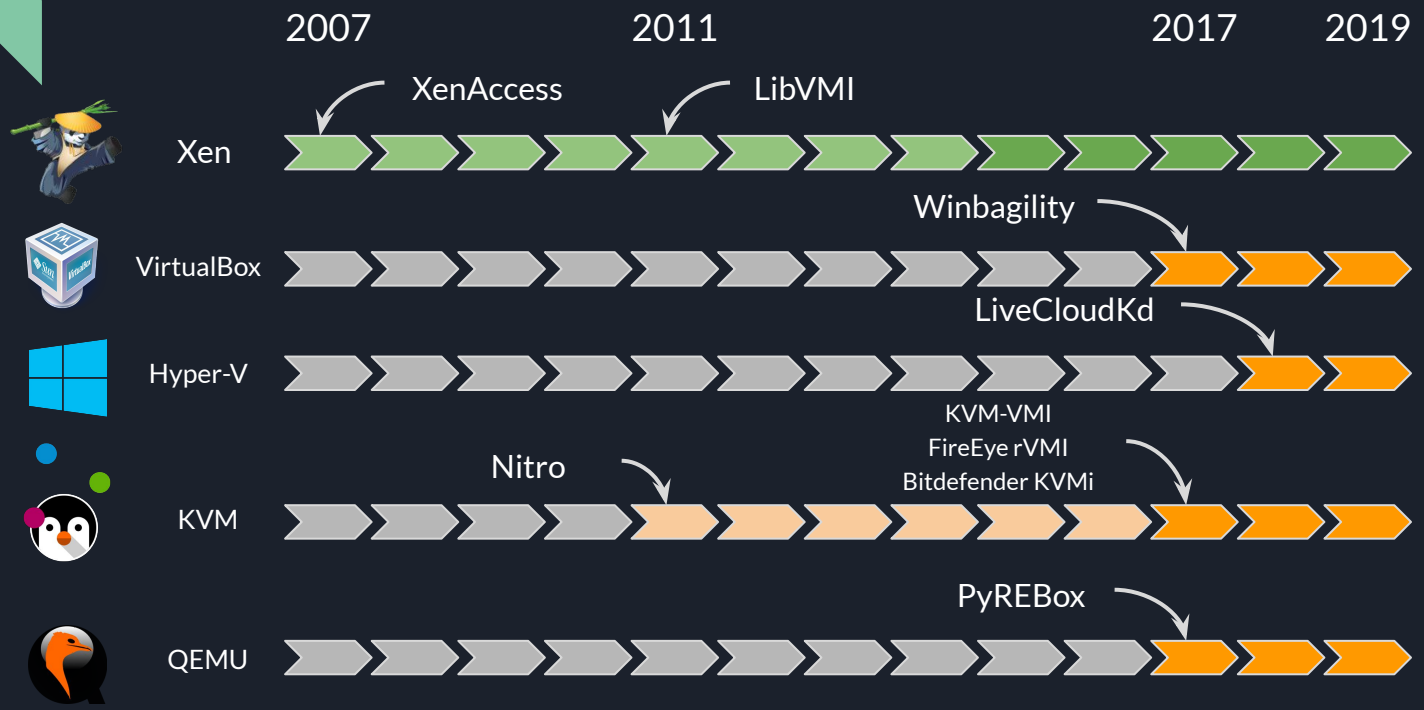
VM Introspection : Complexity



- Identify paging
- Walk paging structures

VMI ecosystem in 2020

VMI API: Hypervisor Support

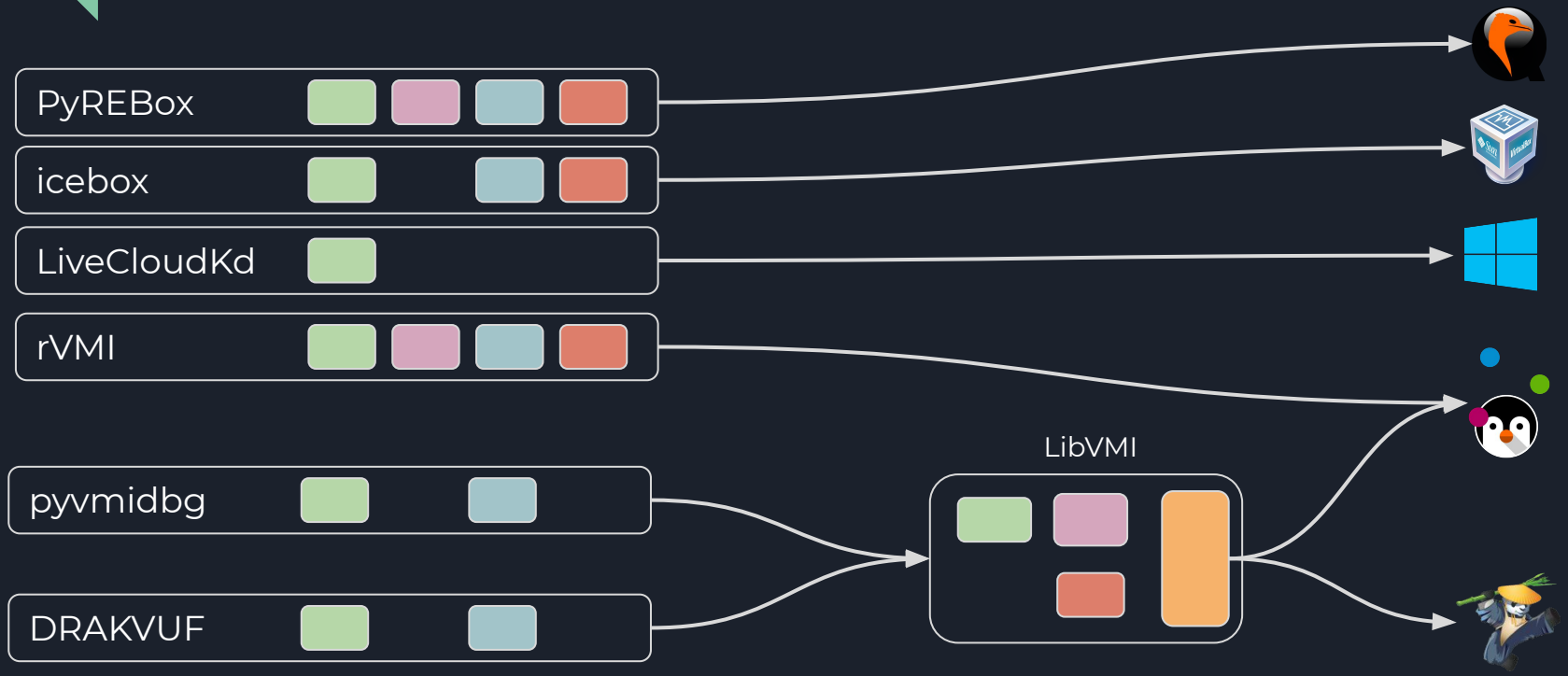


Community Effort

Upstream integration

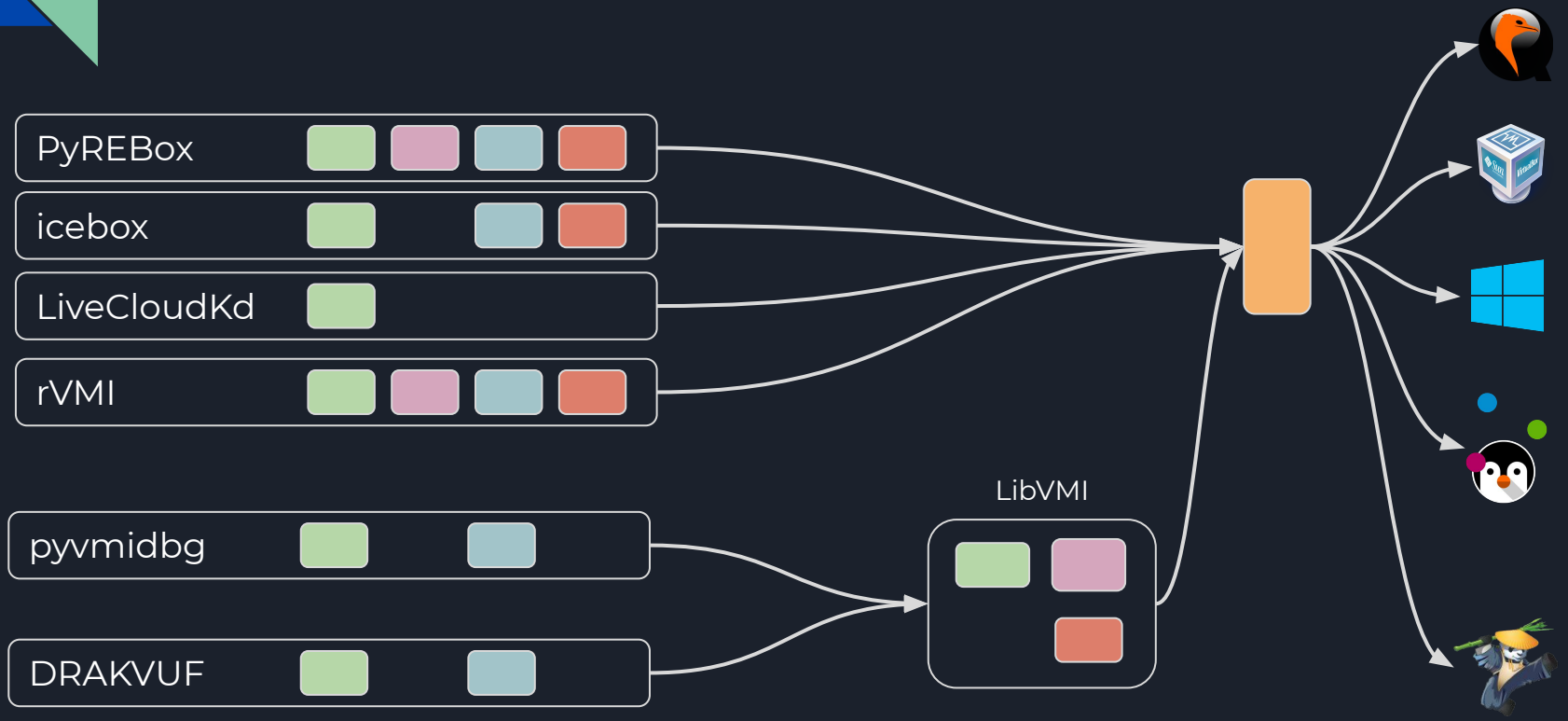
Alternate EPT/RVI available

VMI Projects : Silos

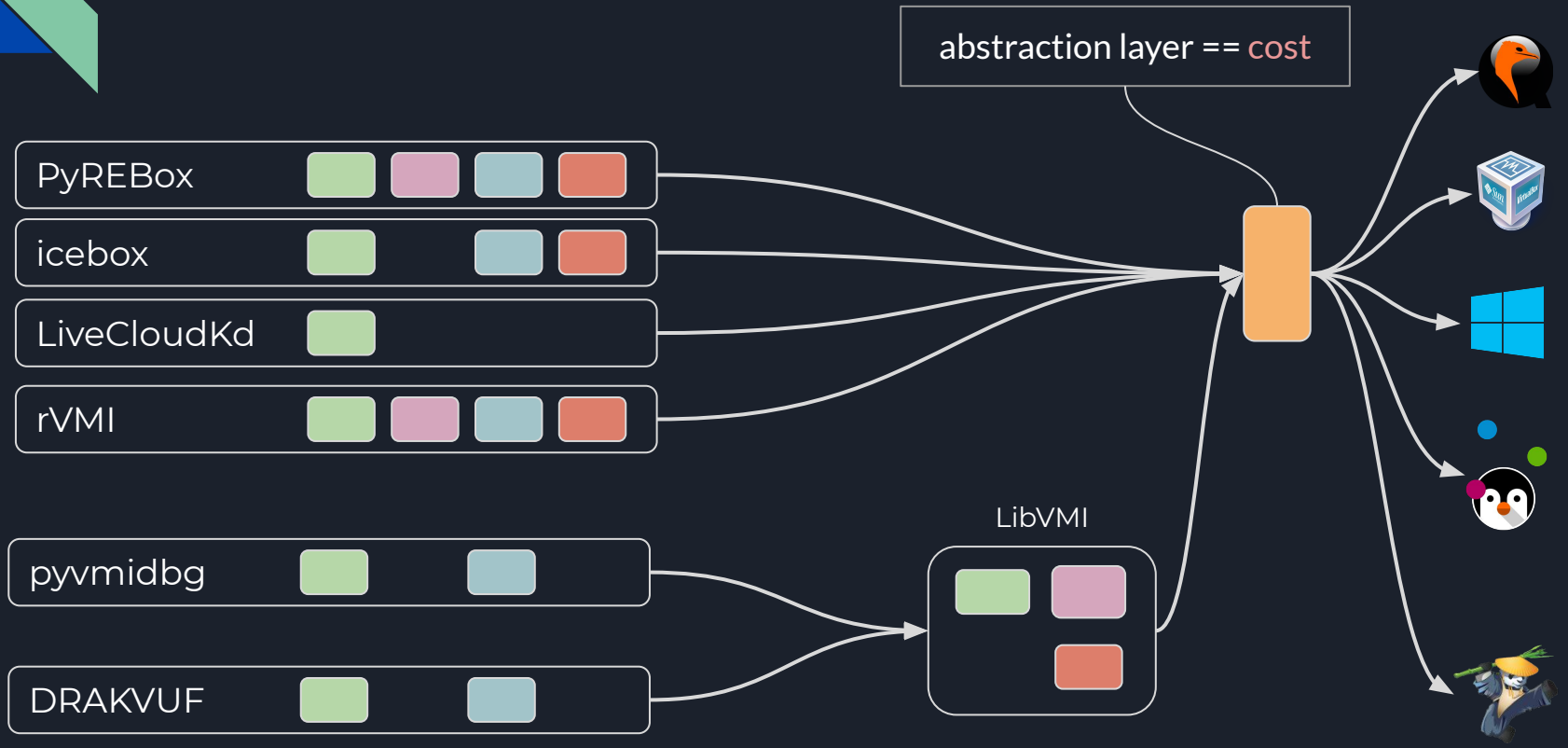


The Idea :
Unifying the ecosystem

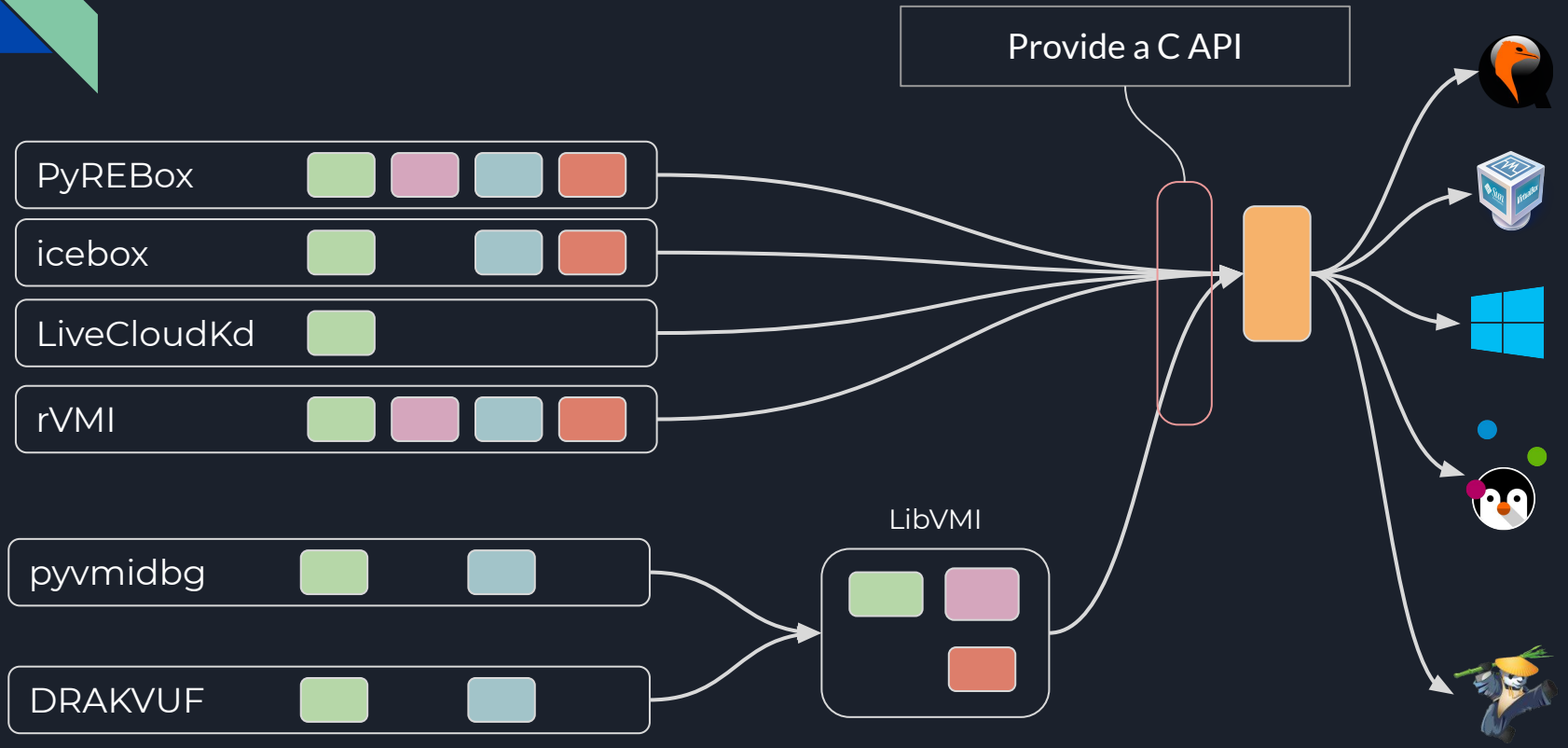
Unifying the ecosystem



Unification : Constraints - Speed

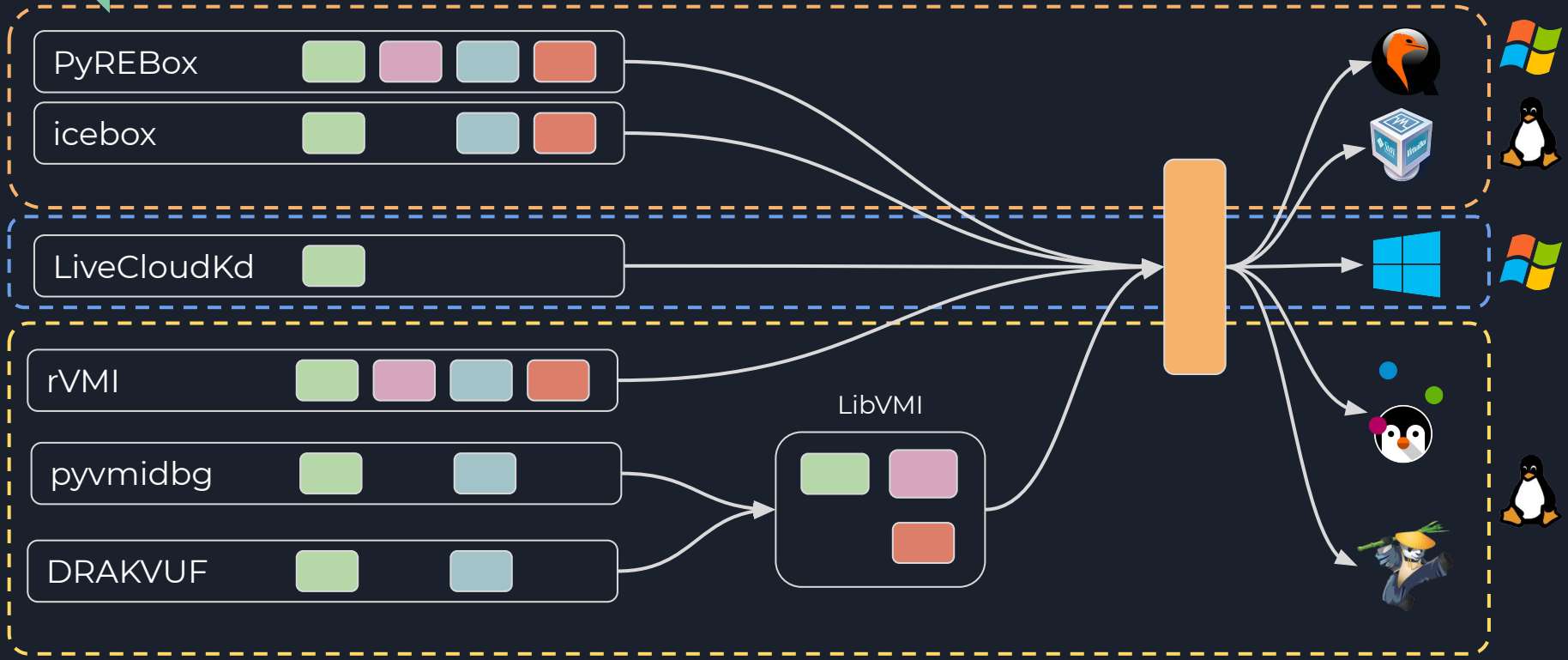


Unification : Constraints - Compatibility

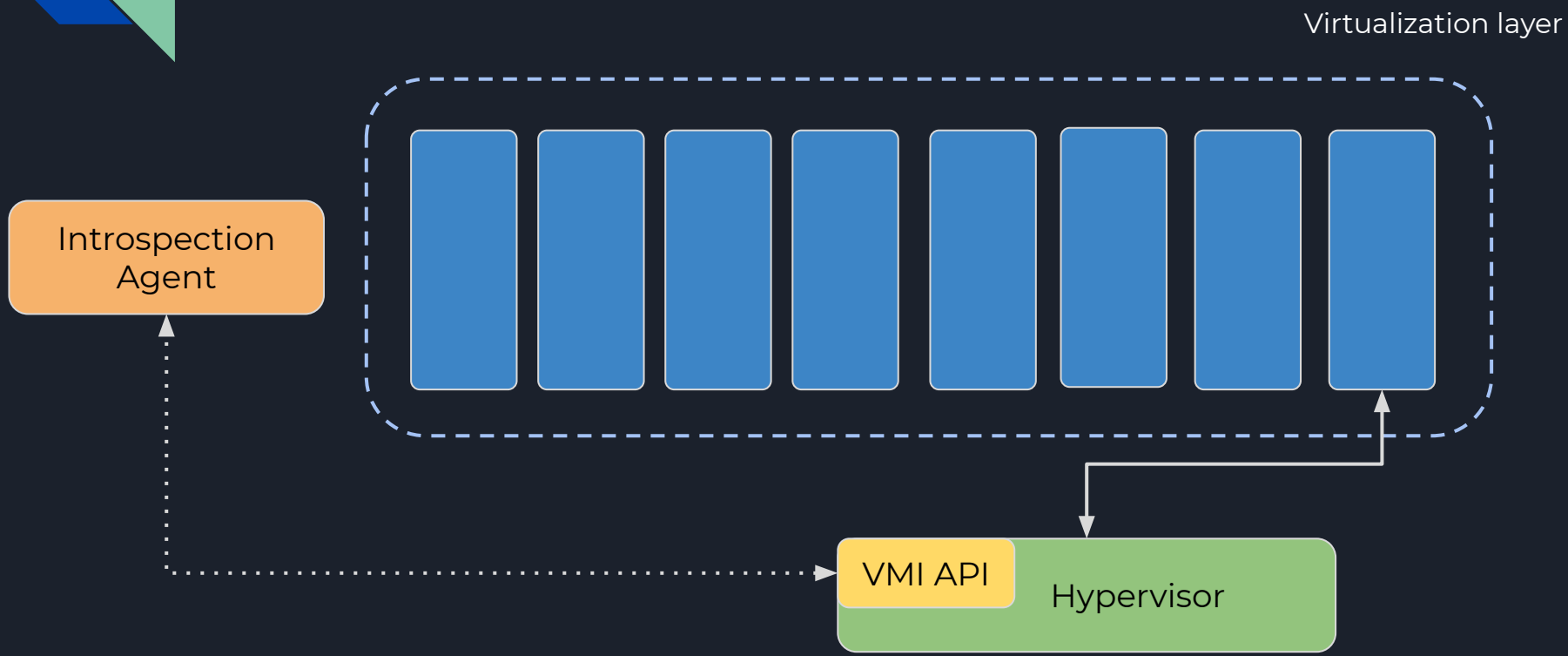


Unification : Constraints - Cross-Platform

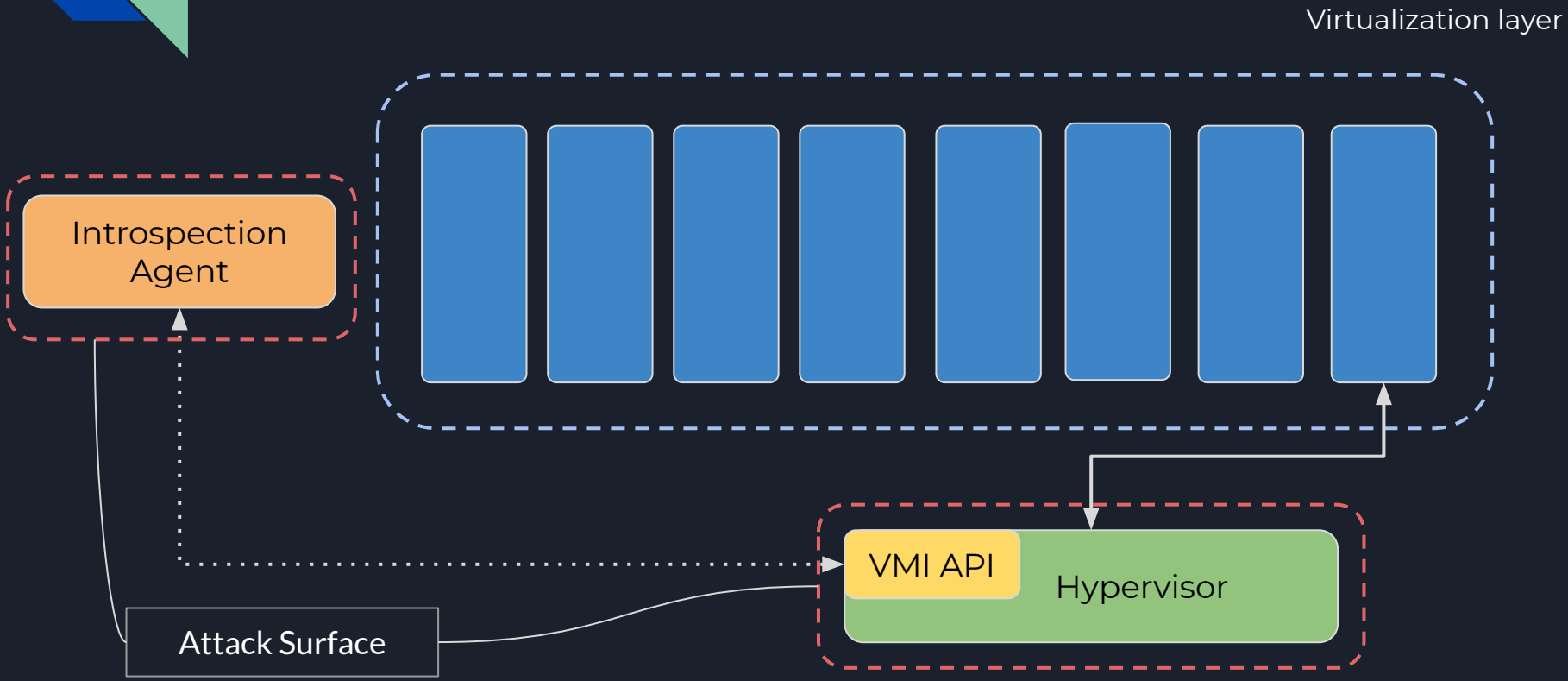
Be easy to maintain on Windows/Linux



Desired Quality - Memory Safety



Desired Quality - Memory Safety



Unifying the ecosystem

- ✅ Speed
- ✅ C compatibility
- ✅ Cross-platform
- ✅ Memory safety

libmicrovmi : Playing lego with VMI

https://github.com/Wenzel/libmicrovmi

VMI Apps

Dynamic Analysis
- pyvmidbg
- icebox
- rVMI
- LiveCloudKd
- DECAF
- PANDA
- PyREBox
- Drakvuf

Live-Memory Analysis
- Volatility
- Rekall

OS Hardening

Monitoring

Fuzzing
- ApplePie

Semantic Engine

Address Translation

Breakpoint Manager

Event Dispatcher

Unified low-level VMI API

Emulators

Hypervisors

Custom Hypervisor

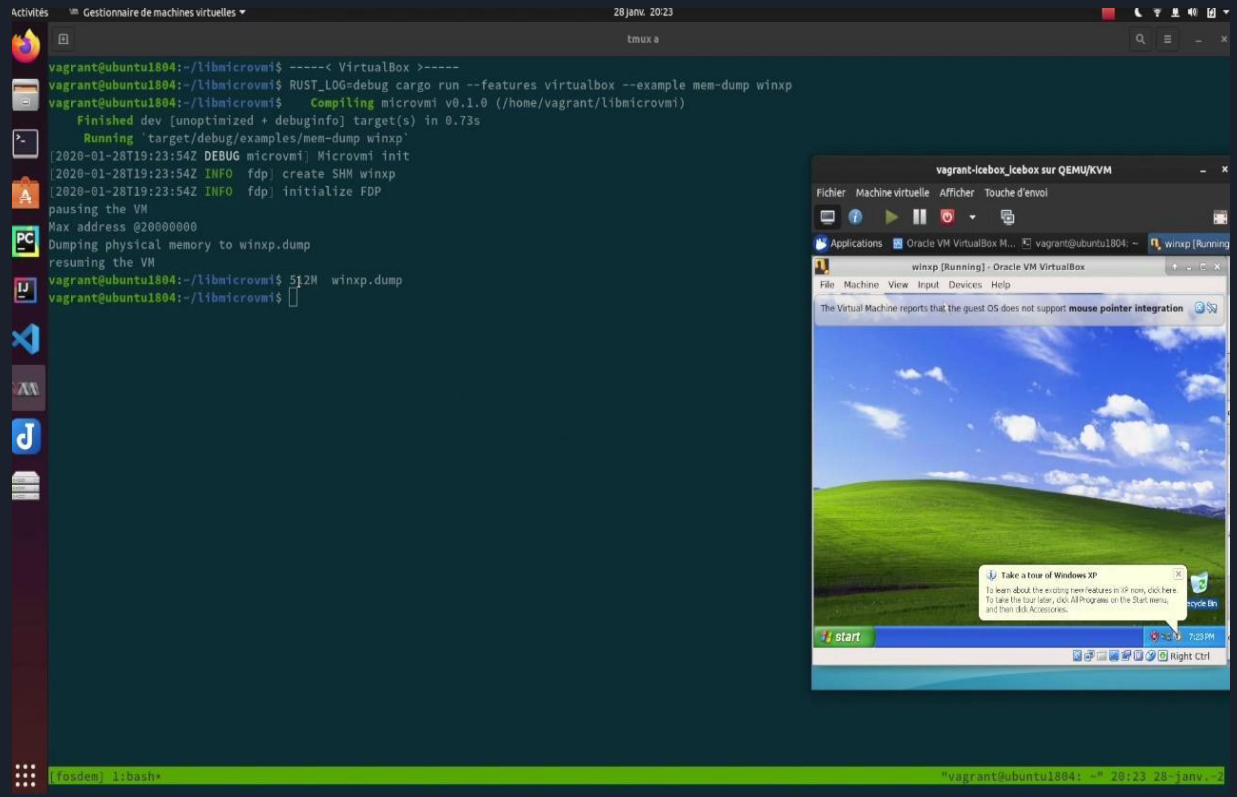
libmicrovmi

libmicrovmi : Status

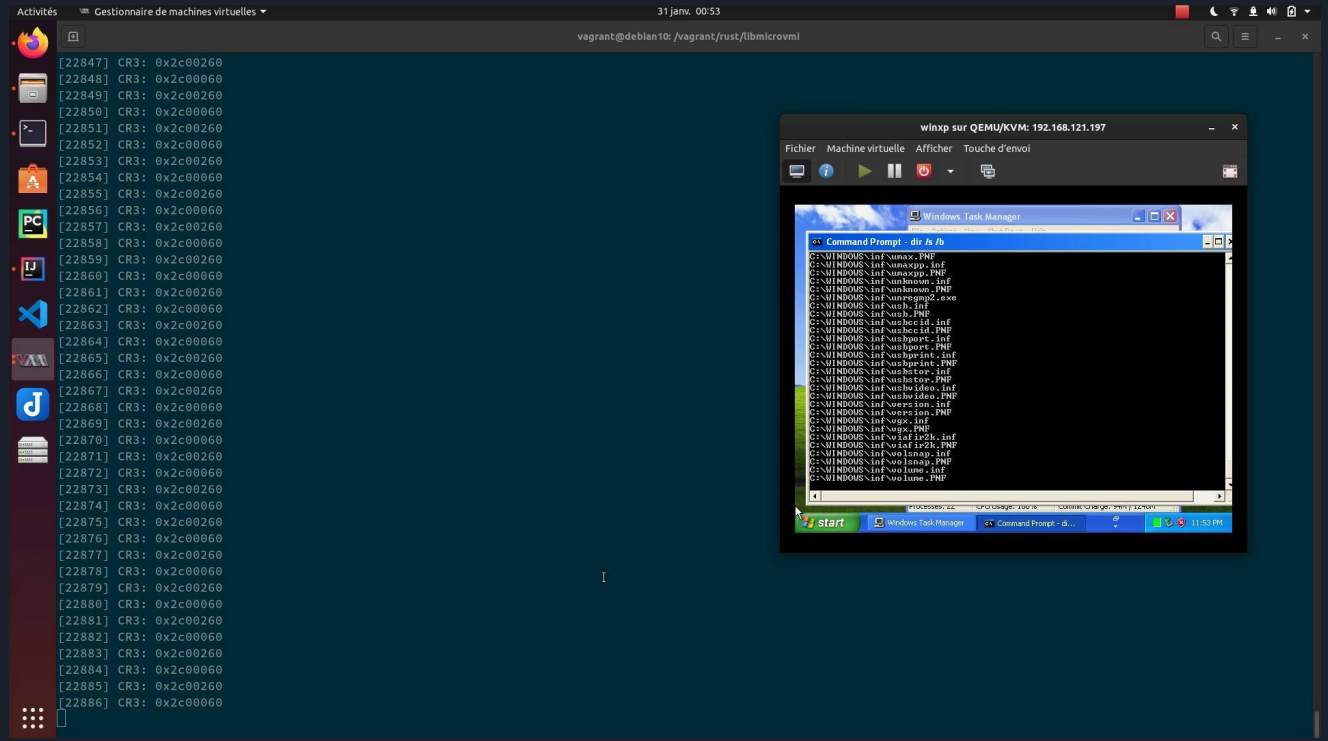
- ✅ read physical memory
- 🟧 r/w VCPU registers
- 🟧 Subscribe on hardware events
  - 🟧 registers
    - 🟧 mov CR0/CR3/CR4
    - mov DRx
    - r/w MSR
  - interrupts
  - singlestep
  - descriptors
  - hypercalls
  - memory
    - r/w/x on frame
    - switch on alternate EPT views
- Utilities
  - foreign memory mapping
  - pagefault injection

- ✅ C API
- 🟧 LibVMI integration
- 🟧 Xen
  - xenctrl / -sys
  - xenstore / -sys
  - xenforeignmemory / -sys
- 🟧 KVM
  - kvmi / -sys
- 🟧 VirtualBox
  - fdp / -sys
- ⬜ Hyper-V
  - vid-sys
- QEMU

Demo: mem-dump on Xen / KVM / VirtualBox



Demo : Intercepting context switch on KVM (CR3 events)



- Demo is running in nested virtualization

Future - VM Introspection

- An OS-independent hooking framework
  - Hypervisor-based intrusion detection
  - Full-system view for debuggers
  - A new layer of hardening and defense in depth
  - Snapshot-based fuzzing capabilities
- Make VM Introspection a new commodity

One Last Thing : GSoC

- We will propose libmicrovmi for the GSoC
- Part of the Honeynet organization
- Ideas
  - Improve an existing driver
    - Xen / KVM / VirtualBox
  - Add support for emulators
    - QEMU / Bochs / Unicorn
  - Propose stealth breakpoints implementation based on EPT
  - Add libloading support to `rust-lang/bindgen #1541`

Rustifying the VM Introspection ecosystem

FOSDEM 2020

https://github.com/Wenzel/libmicrovmi

@rageagainsthepc @mtarral

Dorian Eikenberg Mathieu Tarral