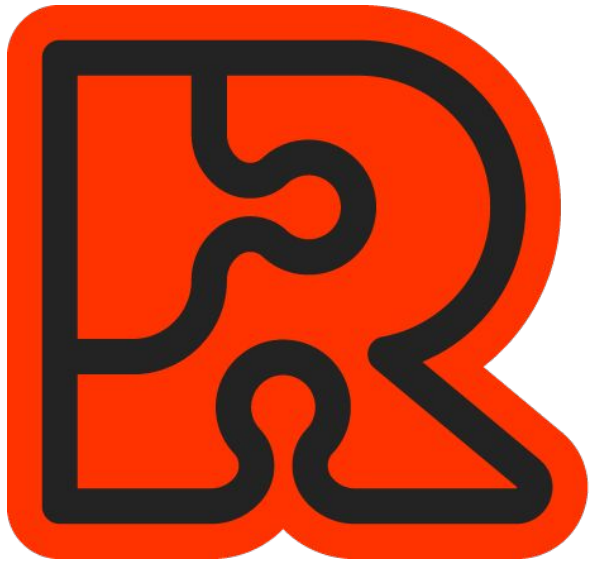


RUST-VMM

evolution on ecosystem and monorepo

FOSDEM 2026

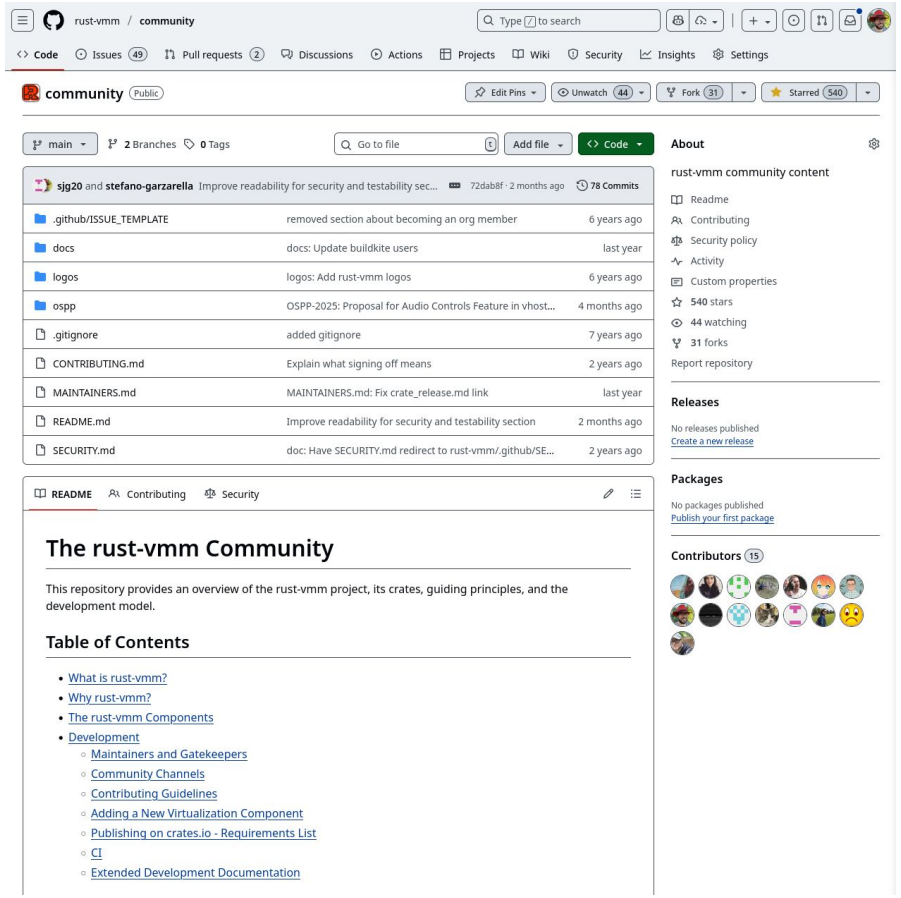
Stefano Garzarella
<sgarzare@redhat.com>

Ruoqing He
<heruoqing@iscas.ac.cn>

Patrick Roy
<patrick.roy@linux.dev>

rust-vmm

- https://github.com/rust-vmm/community
  - **open-source** project that empowers the community to **build custom** Virtual Machine Monitors (**VMMs**) and **hypervisors**
  - **set of virtualization components** that any project can use to quickly develop virtualization solutions
- Why rust-vmm?
  - Reduce code duplication
  - Faster development
  - Security & Testability
  - Clean interface
- Community Channels
  - Mailing list: http://lists.opendev.org/cgi-bin/mailman/listinfo/rust-vmm
  - Slack workspace: join here!



Repositories & crates

- **Core VM**
  - KVM wrappers
    - kvm-ioctls, kvm-bindings
  - Microsoft Hyper-V wrappers
    - mshv-ioctls, mshv-bindings
  - Xen hypercall and interfaces in Rust
    - xen, xen-ioctls, xen-bindings, etc.
  - Linux kernel loader
    - linux-loader
- **Resource management**
  - VM resource allocator (MMIO & PIO addresses, IDs, etc.)
    - vm-allocator
  - VM guest memory
    - vm-memory
- **Device Emulation & I/O**
  - VIRTIO abstractions for virtqueue and devices
    - virtio-queue, virtio-bindings, etc.
  - vhost, vhost-user, and vDPA abstractions
    - vhost, vhost-user-backend
  - vhost-user devices
    - vhost-device-can, vhost-device-console, etc.
  - Safe wrappers for VFIO "Virtual Function I/O"
    - vfio-user, vfio-ioctls, vfio-bindings
  - Legacy devices emulation
    - vm-superio, vm-superio-ser
- **rust-vmm ecosystem**
  - community
  - CI & development
    - rust-vmm-ci, rust-vmm-container
- ... and others

Adoption

Several projects are using rust-vmm crates :

- Firecracker
  - *Firecracker enables you to deploy workloads in lightweight virtual machines, called microVMs, which provide enhanced security and workload isolation over traditional VMs, while enabling the speed and resource efficiency of containers*
- Cloud Hypervisor
  - *Cloud Hypervisor is an open source Virtual Machine Monitor (VMM) that runs on top of the KVM hypervisor and the Microsoft Hypervisor (MSHV).*
  - *Cloud Hypervisor is implemented in Rust and is based on the Rust VMM crates.*
- libkrun
  - *libkrun is a dynamic library that allows programs to easily acquire the ability to run processes in a partially isolated environment using KVM Virtualization on Linux and HVF on macOS/ARM64.*
- Kata Containers' Dragonball Sandbox
  - *Dragonball Sandbox is a light-weight virtual machine manager (VMM) based on Linux Kernel-based Virtual Machine (KVM), which is optimized for container workloads.*

Adoption

- vhost-user devices
  - Can be used with any VMMs implementing vhost-user frontend like QEMU, Cloud Hypervisor, etc.
  - rust-vmm/vhost-device
  - virtiofsd
- hreitz/imago
  - *Provides access to VM image formats (e.g. raw, qcow2)*
- archlinux/vmexec
  - *vmexec is a zero-setup CLI tool that runs single commands in a throwaway virtual machines.*
- dragonflyoss/nydus
  - *Dragonfly image service, providing fast, secure and easy access to container images.*
  - *Nydus takes in either FUSE or virtiofs protocol to service POD created by conventional runc containers or vm-based Kata Containers.*

Related projects

- crosvm

*crosvm is **open to using rust-vmm modules**.*

Soon after crosvm's code was public, Amazon used it as the basis for their own VMM named Firecracker. After Firecracker came other rust-based VMM implementations, all using parts of crosvm. In order to drive collaboration and code sharing, an independent organization was created, named rust-vmm.

- OpenVMM

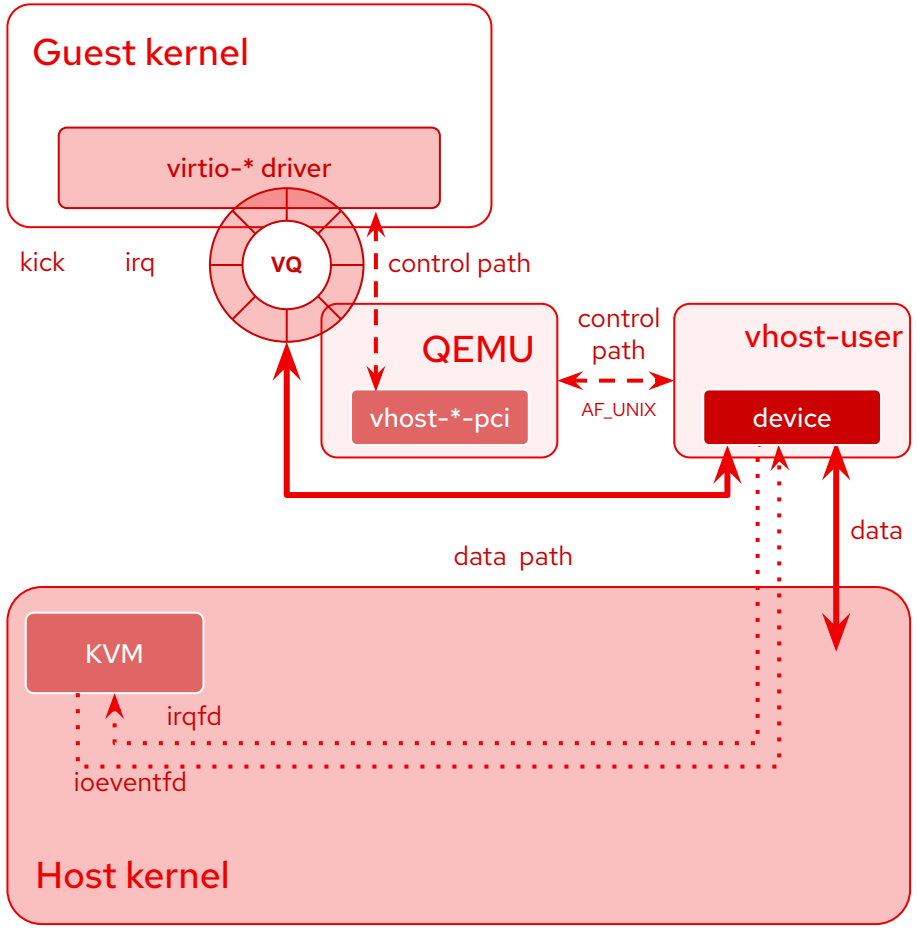
*We are **indebted to the Rust VMM community** for their trailblazing work. Now that the OpenVMM project is open source, we **hope to find ways to collaborate on shared code** while maintaining the benefits of the OpenVMM architecture.*

Supported Architecture & Operating Sytems

- Architectures
  - x86_64
  - aarch64
  - riscv64
- Operating Systems
  - Linux
  - Windows
    - partial support
  - POSIX (macOS/*BSD)
    - on-going effort mainly related to vhost-user devices
      - see FOSDEM 2025 - Can QEMU and vhost-user devices be used on macOS and *BSD?

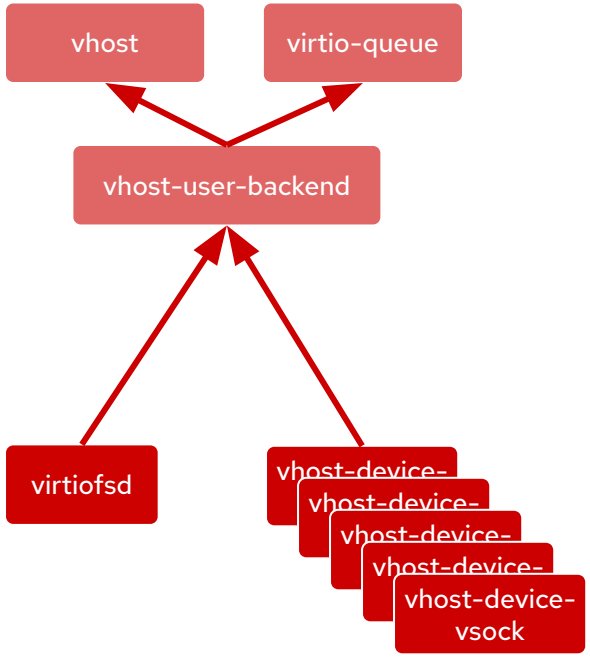
vhost-user protocol

- https://qemu-project.gitlab.io/qemu/interop/vhost-user.html
  - *control plane to establish virtqueue sharing with a user space process on the same host*
  - **frontend**
    - application that shares its virtqueues (i.e. VMM like QEMU)
  - **backend**
    - consumer of the virtqueues (i.e. virtio device emulation)
- Key components
  - UNIX domain socket (**AF_UNIX**)
  - **shared memory** represented by a file descriptor
  - notifications
    - **eventfd** or **pipe**/**pipe2**
- Interested on more details?
  - Attend FOSDEM 2026 - Where should my VIRTIO device live?
    - at 12:30 in this room



rust-vmm virtio: vhost-user devices

- vhost
  - A pure rust library for vDPA, vhost and vhost-user
- vhost-user-backend
  - A framework to implement vhost-user device services
    - daemon control object to start and stop the device service
    - trait to handle vhost-user control messages
    - vring trait to access virtqueues
- Several vhost-user devices implemented
  - https://github.com/rust-vmm/vhost-device
    - can, console, gpio, gpu, i2c, input, rng, scmi, scsi, sound, spi, video (staging), vsock
  - https://gitlab.com/virtio-fs/virtiofsd
    - fs



rust-vmm virtio: latest changes

- POSIX support [uran0sH]
  - rust-vmm/vhost#308 - Replace Eventfd with EventNotifier/EventConsumer
- Formal verification of VIRTIO devices (KANI) [priyasiddharth]
  - rust-vmm/vm-virtio#338
  - rust-vmm/vm-virtio#346
  - rust-vmm/vm-virtio#371
- Others
  - rust-vmm/vhost#266 - Add support for `VHOST_USER_RESET_DEVICE` [XanClic]
  - rust-vmm/vm-virtio#278 - Add `Reader/Writer` classes to iterate over virtqueue descriptors [MatiasVara]
  - rust-vmm/vm-virtio#310 - Add Packed Descriptor [uran0sH]
- Live migration support for vhost-user devices
  - rust-vmm/vhost#203 - Add back-end's internal state migration support [germag]
  - rust-vmm/vhost#206 - Add replaceable-mmapped bitmap support [germag]
  - rust-vmm/vhost#218 - Adding `POSTCOPY` support [ShadowCurse]
- Shared object and shared memory support for vhost-user devices
  - rust-vmm/vhost#251 - Add support for SHMEM_MAP/UNMAP backend requests [aesteve-rh]
  - rust-vmm/vhost#241 - Add shared objects support [aesteve-rh]
  - rust-vmm/vhost#268 - Add support for `VHOST_USER_GET_SHARED_OBJECT` [dorindabassey]

rust-vmm: GSoC under QEMU community

- GSoC 2025
  - vhost-user devices in Rust on macOS and *BSD
    - Contributor: Wenyu Huang
      - GSoC final report
    - Mentors: Stefano Garzarella, German Maglione, Oliver Steffen
  - Adding Kani proofs for Virtqueues in Rust-vmm
    - Contributor: Siddharth Priya
      - GSoC final report
    - Mentor: Matias Ezequiel Vara Larsen

Google Summer of Code

- **GSoC 2026**
  - https://wiki.qemu.org/Google_Summer_of_Code_2026
  - **Implement virtio-rtc device as a vhost-user device**
    - https://wiki.qemu.org/Google_Summer_of_Code_2026#virtio-rtc_vhost-user_device
  - Mentors:
    - Matias Ezequiel Vara Larsen <mvaralar@redhat.com>
    - Francesco Valla <francesco@valla.it>
    - Stefano Garzarella <garzare@redhat.com>
  - **If you are interested, reach us!**

rust-vmm vm-memory: on-going works

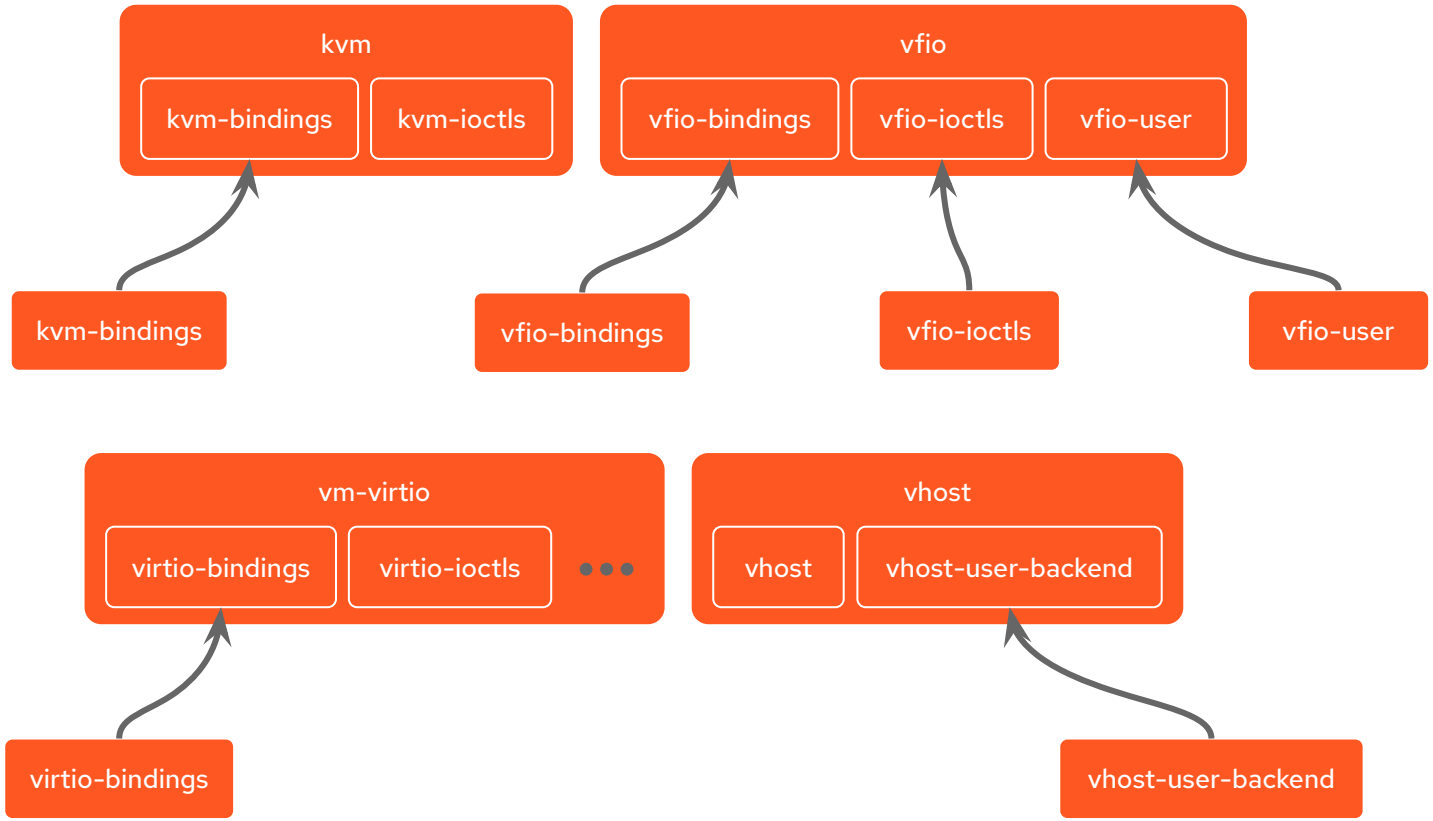
- [Done] Support for guest_memfd / memory regions with "metadata"
  - #312: Introduce generic `GuestRegionCollection`
- [Done] IOMMU Support
  - #339, #327
  - IOMMU in rust-vmm, and new FUSE+VDUSE use cases :: KVM Forum 2025
- [WIP] Make crate features additive
  - #317: Make xen feature not mutually exclusive with support for non-Xen memory regions
- [WIP?] Usage in QEMU
  - https://lore.kernel.org/qemu-devel/aJynpEWNfbrTeirQ@intel.com/T/
- [Idea] "Slim down" `VolatileSlice` to be just pointer + length?
  - Ease conversion between VolatileSlice and iovec in virtio code
- [Idea] Use zerocopy instead of ByteValued
  - #246: Leverage rust golden standard for safe transmute instead of homegrown solution

Operational Challenges: Churn of Releases

- Deep dependency chains, version bumps need to be propagated
  - E.g. vmm-sys-util ← vm-memory ← vm-virtio ← vhost ← (binaries) firecracker, vhost-device, virtiofsd
- Potential/Partial Solutions
  - Automate crates.io releases with GitHub actions [example]
  - Do major dependency upgrade in minor release using version ranges [example]
    - `vmm-sys-util = ">=0.14, <=0.15"`
    - Tricky, need to be careful when to bump lower end of range
  - Mono-Repo(s) [see next slide]
    - Also answers questions such as “how to keep configuration consistent across our ~20 repositories”

rust-vmm mono repo: Works done

- Mono-repos:
  - kvm: Merge kvm-bindings into kvm-ioctls
  - vfio: Merge vfio-user into vfio
  - vm-virtio: Merge virtio-bindings into vm-virtio workspace
  - vhost: Convert vhost repo as a workspace

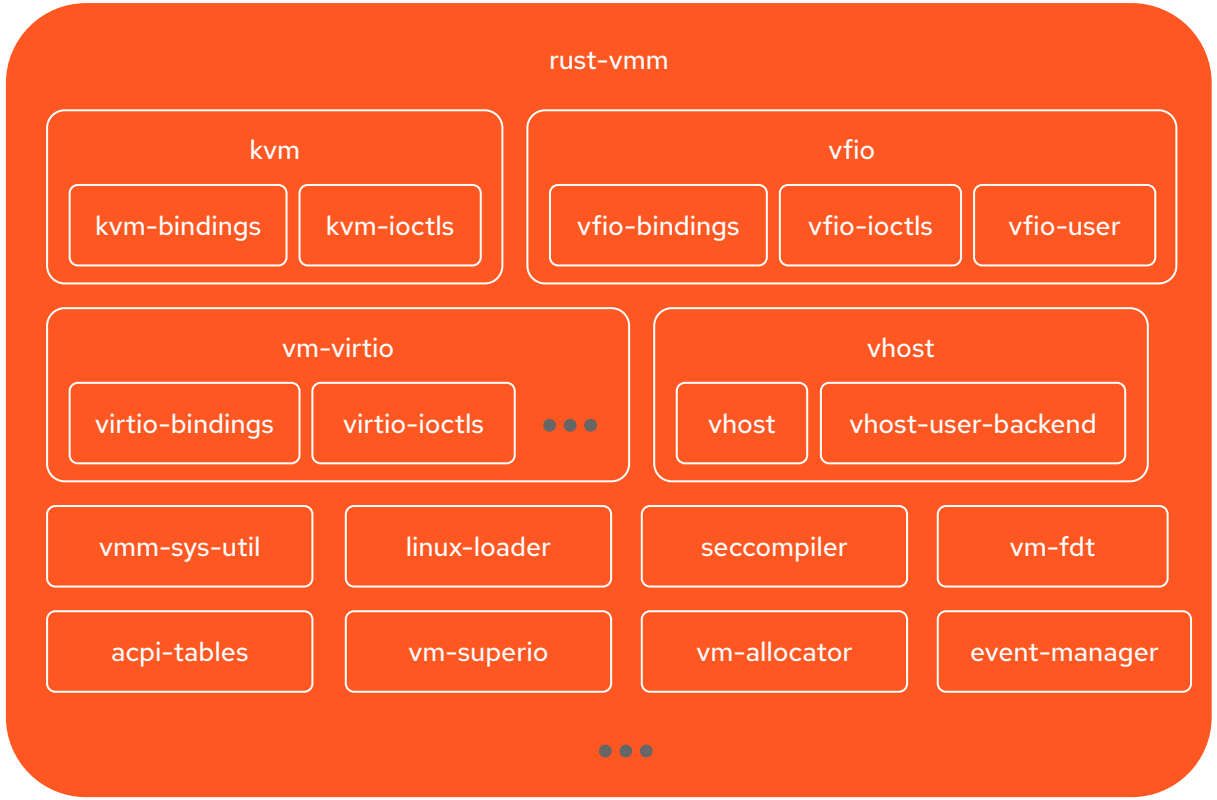


rust-vmm mono repo: End Goal (WiP)

- All crates resides in a single rust-vmm monorepo.
- All crates are member of the `rust-vmm` workspace.
- All crates use path dependency without version to stay up-to-date.

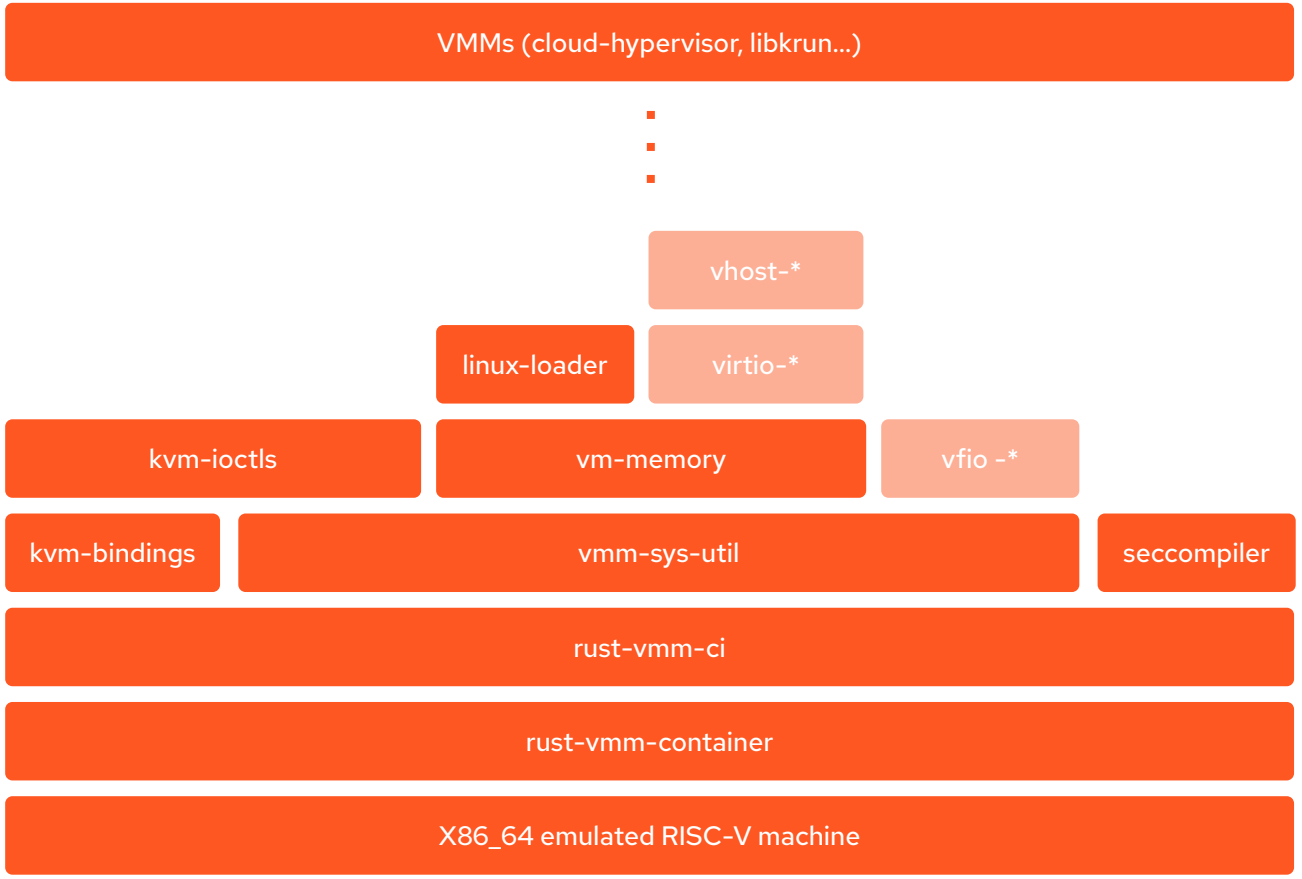
Current status:

- CI initialized and ready
- vmm-sys-utils merging on the way



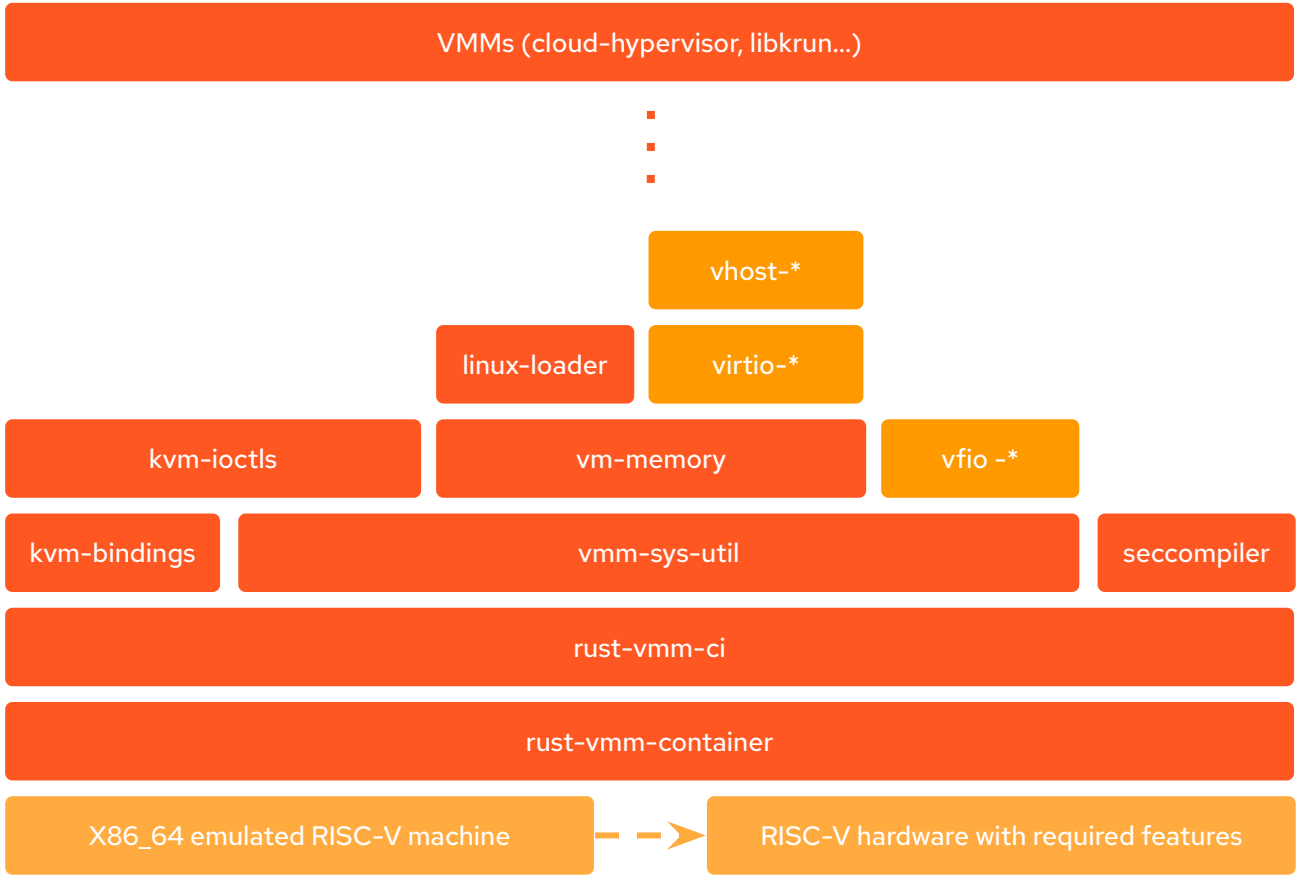
rust-vmm new arch: RISC-V

- Support rust-vmm infrastructure (rust-vmm-container & rust-vmm-ci) on RISC-V:
  - Introduce RISC-V CI container
- Support rust-vmm crates on RISC-V:
  - Introduce RISC-V bindings
  - Introduce RISC-V ioctls
  - Introduce RISC-V architecture support for linux-loader
  - ...



rust-vmm RISC-V: Future works

- CI Infrastructure
  - Switch from QEMU emulated environment to RISC-V hardware with H extension, AIA and IOMMU feature.
- Core crates
  - Extend RISC-V support to `vhost-*`, `vfio-*` and other crates, to address the gaps between X86 and ARM.



Thank you!

https://github.com/rust-vmm/community

Stefano Garzarella
<sgarzare@redhat.com>

Ruoqing He
<heruoqing@iscas.ac.cn>

Patrick Roy
<patrick.roy@linux.dev>