Userland TCP/IP stack for external container connectivity

Usermode networking in CodeReady Containers

Christophe Fergeau
Senior Software Engineer

Introduction

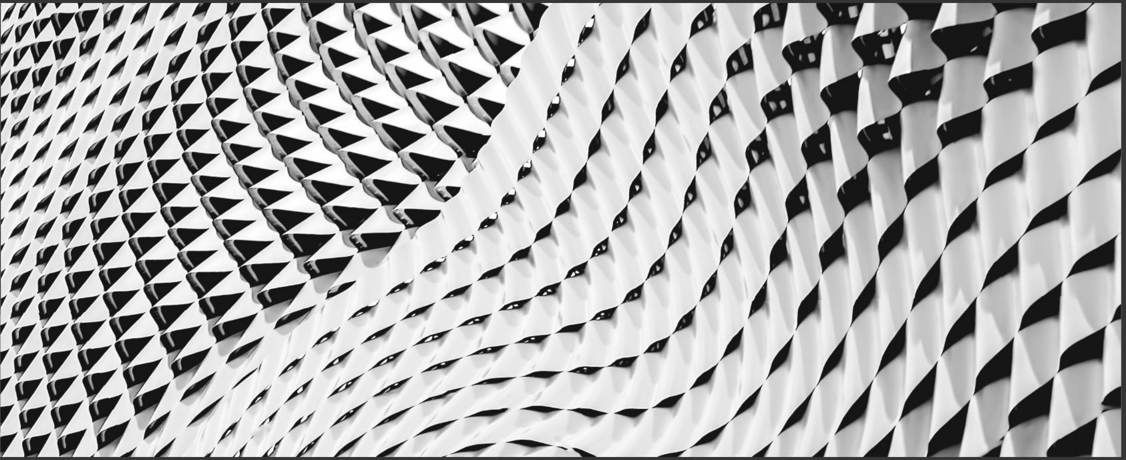
- Christophe Fergeau <cfergeau@redhat.com>
- Working at Red Hat
- Member of the CodeReady Containers team
- Previously worked in the virtualization team (SPICE)

What we'll discuss today

- CodeReady Containers
- User-mode networking

Red Hat

CodeReady Containers



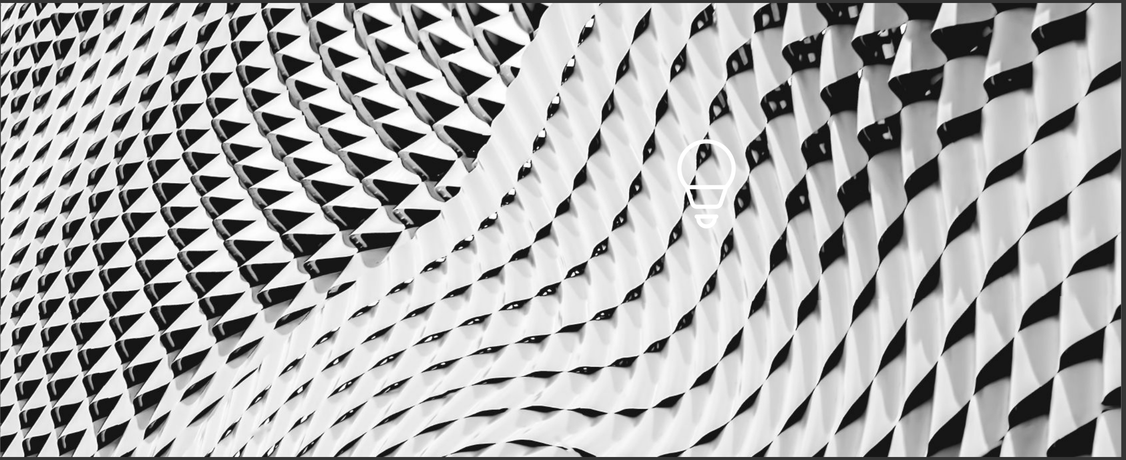
What is CodeReady Containers?

- Runs a Red Hat OpenShift 4 cluster on your laptop or desktop
    - « Red Hat® OpenShift® is an enterprise-ready Kubernetes container platform built for an open hybrid cloud strategy. »
- Meant for development and testing on a throw-away local cluster
- Works on Linux, macOS and Windows
- Work in progress to offer a lighter weight podman-only runtime

Under the hood

- Go binary + pre-generated virtual machine image
- Uses native hypervisors
  - QEMU+KVM on linux
  - HyperKit on macOS
  - Hyper-V on Windows
- User-mode stack for VM networking

User-mode networking



Why?

- Simplifies VM networking
- Consistent IP addressing
- Works around strict firewalls/VPNs

gvisor-tap-vsock

- https://github.com/containers/gvisor-tap-vsock
- Users:
  - crc
  - podman-machine
- Based on gVisor
  - « gVisor is an application kernel, written in Go, that implements a substantial portion of the Linux system call interface. »

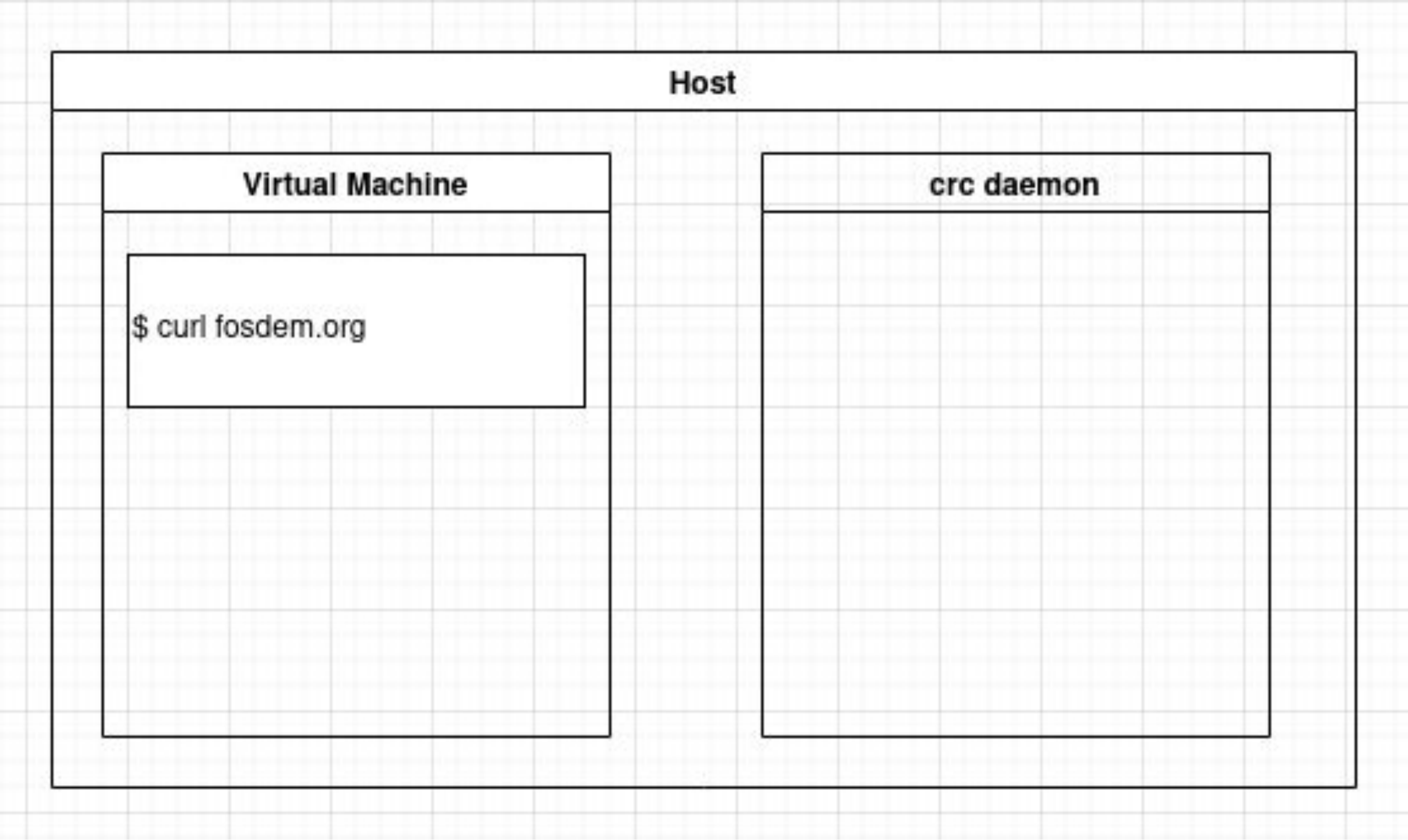
Under the hood

- 2 separate parts:
  - helper running in the VM
  - daemon running on the host
- Usermode networking implemented in the host daemon
- gvisor-tap-vsock implements a network switch (ethernet/layer 2) in software

Under the hood (2)

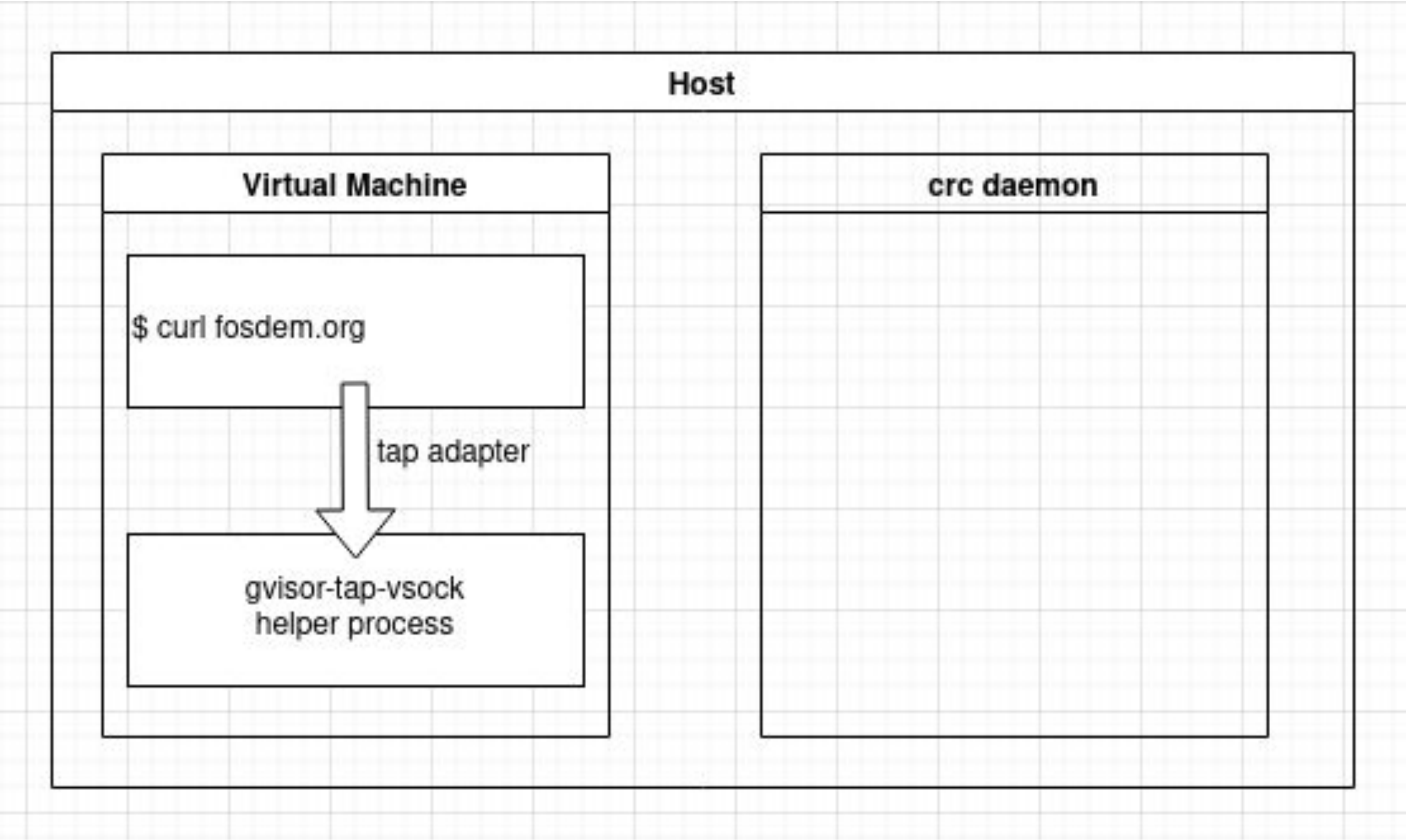
- the daemon running on the host connects to this virtual switch as 192.168.127.1
- gvisor-tap-vsock acts as a dhcp server for the VM, which gets a 192.168.127.x address and uses 192.168.127.1 as the gateway
- gvisor-tap-vsock/pkg/tap transmits packets within that internal network
- gVisor is used for encapsulating/decapsulating network packets, and to transmit packets outside of the 192.168.127.0/24 virtual network

How does it work?

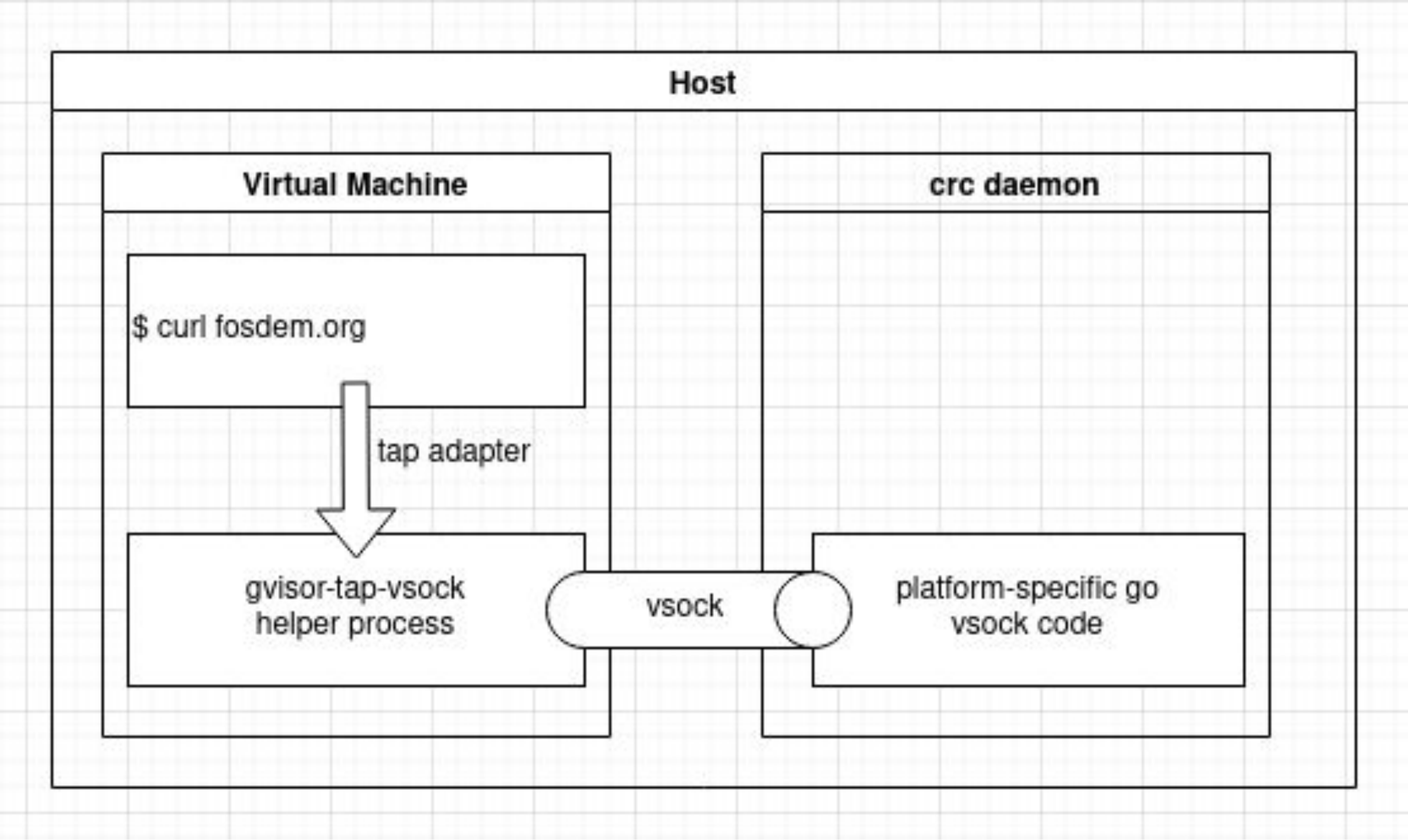


```
$ curl fosdem.org
```

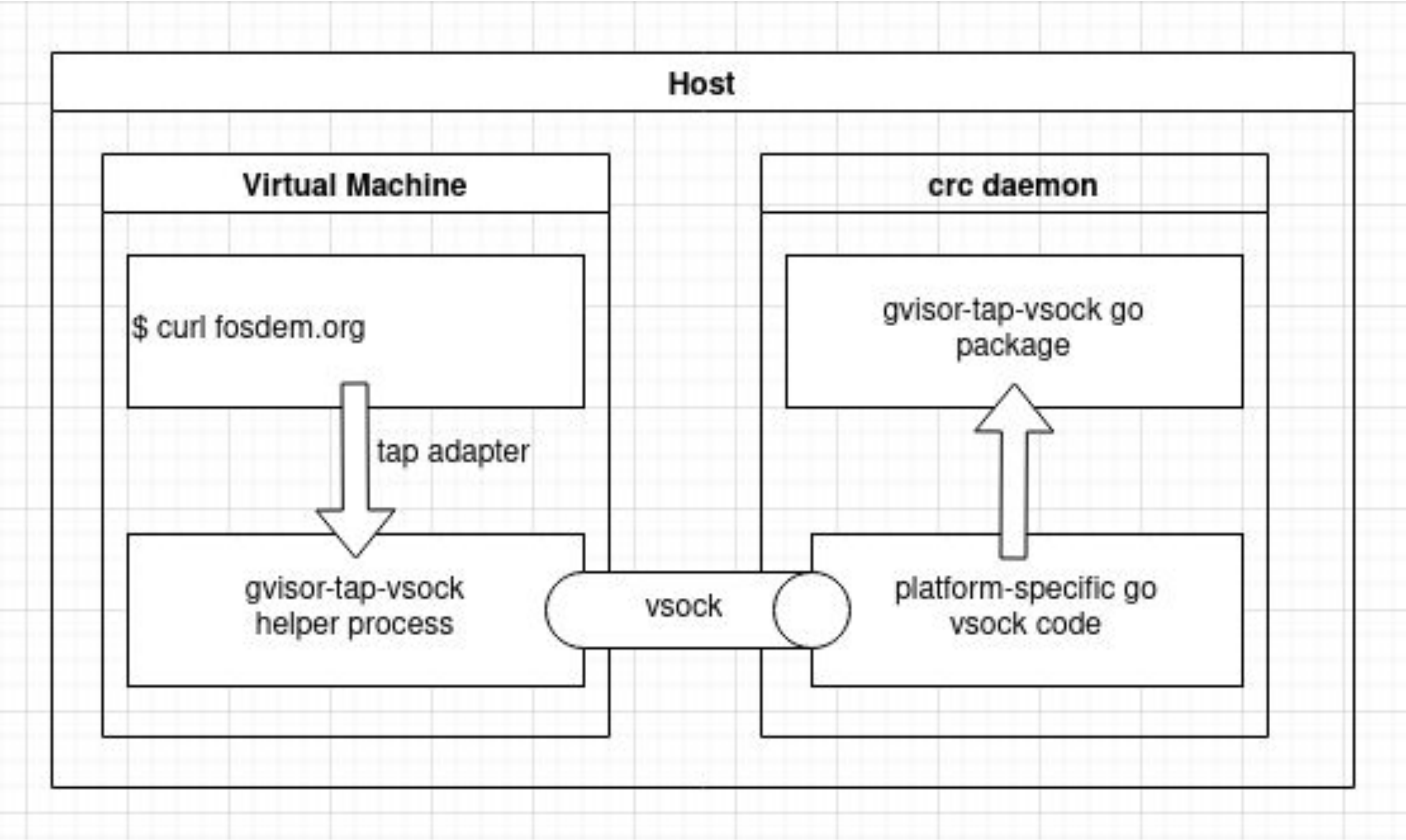
How does it work?



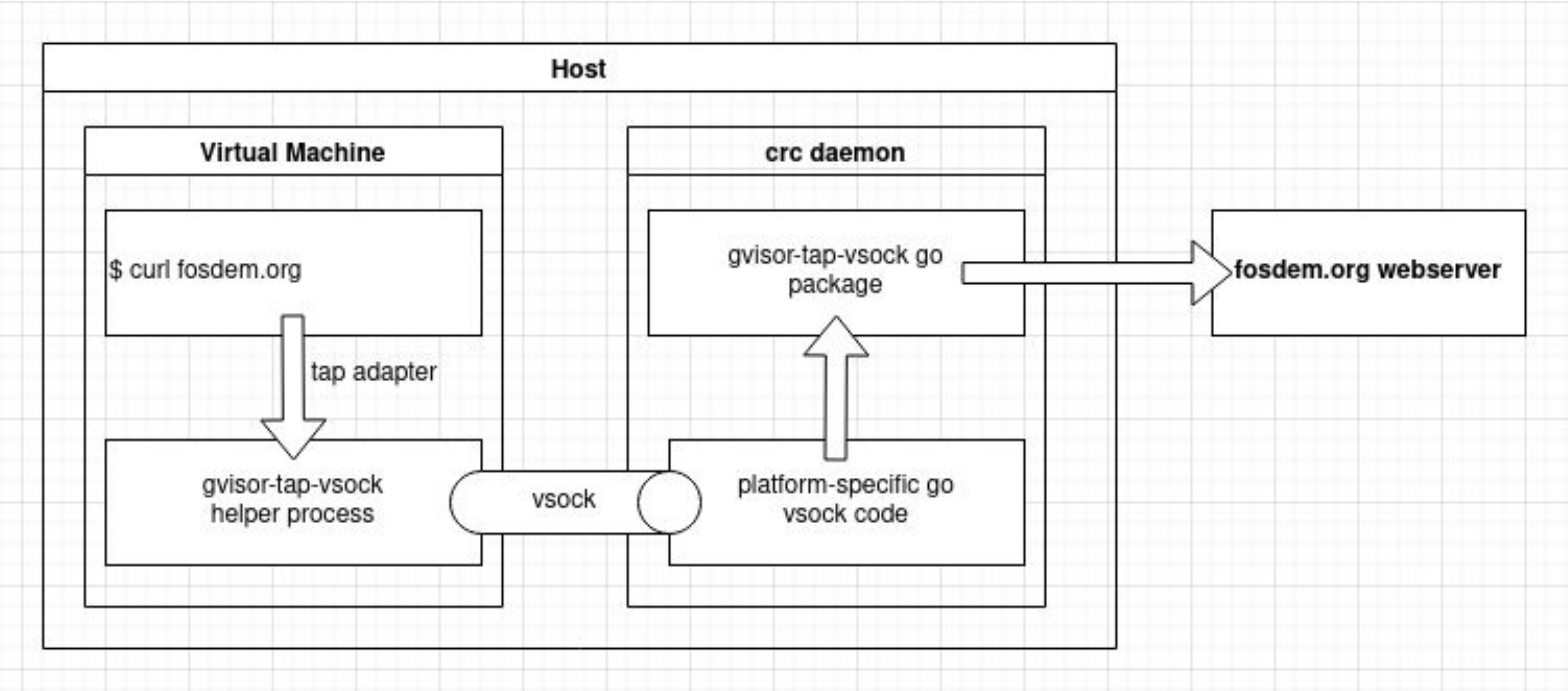
How does it work?


Host
Virtual Machine
$ curl fosdem.org
tap adapter
gvisor-tap-vsock helper process
vsock
crc daemon
platform-specific go vsock code

How does it work?


Host
Virtual Machine
$ curl fosdem.org
tap adapter
gvisor-tap-vsock helper process
vsock
crc daemon
gvisor-tap-vsock go package
platform-specific go vsock code

How does it work?


Host
Virtual Machine
$ curl fosdem.org
tap adapter
gvisor-tap-vsock helper process
vsock
crc daemon
gvisor-tap-vsock go package
platform-specific go vsock code
fosdem.org webserver

What about incoming connections?

- The virtual machine has no externally visible IP address
- Only reachable through its 192.168.127.x address through the daemon
- HTTP API to expose ports:

  -
    ```
    curl -X POST -d '{"local": "127.0.0.1:1234", "remote": "192.168.127.2:22"}' --unix-socket ~/.crc/crc-http.sock http:/unix/network/services/forwarder/expose
    ```
- Services running on the VM need ports to be opened on the host
  - Potential port conflicts (ssh port)

Useful links

- CodeReady Containers: https://github.com/code-ready/crc/
- gvisor-tap-vsock: https://github.com/containers/gvisor-tap-vsock
- Contact information: cfergeau@redhat.com

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat