Mar 2021

Qualcomm

Rust-VMM

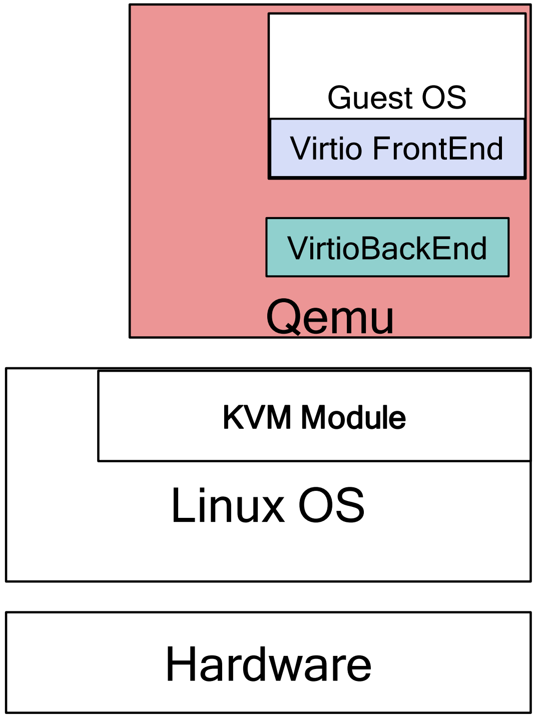
Srivatsa Vaddagiri

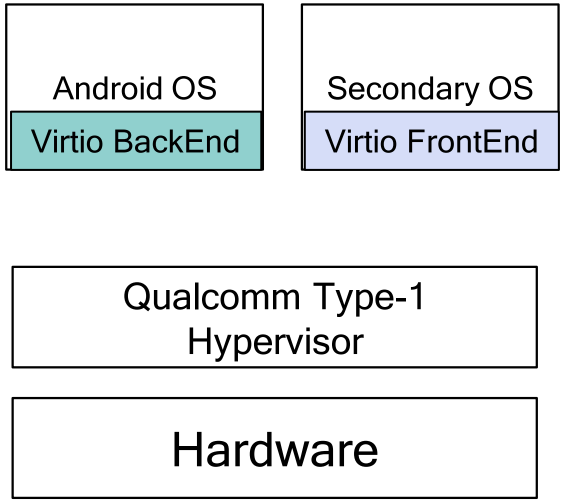
Summary

- **Observations**
  - STR-5 (Virtio RPMB) – using QEMU for backend development
  - STR-26 (Virtio I2C) – Using C language for backend development?

- **Proposal/Ask**
  - Adopt Rust-VMM as default platform for virtio backends
  - Improve Rust-VMM for adoption in ARM64 production environment

Virtio – KVM vs Qualcomm Hypervisor



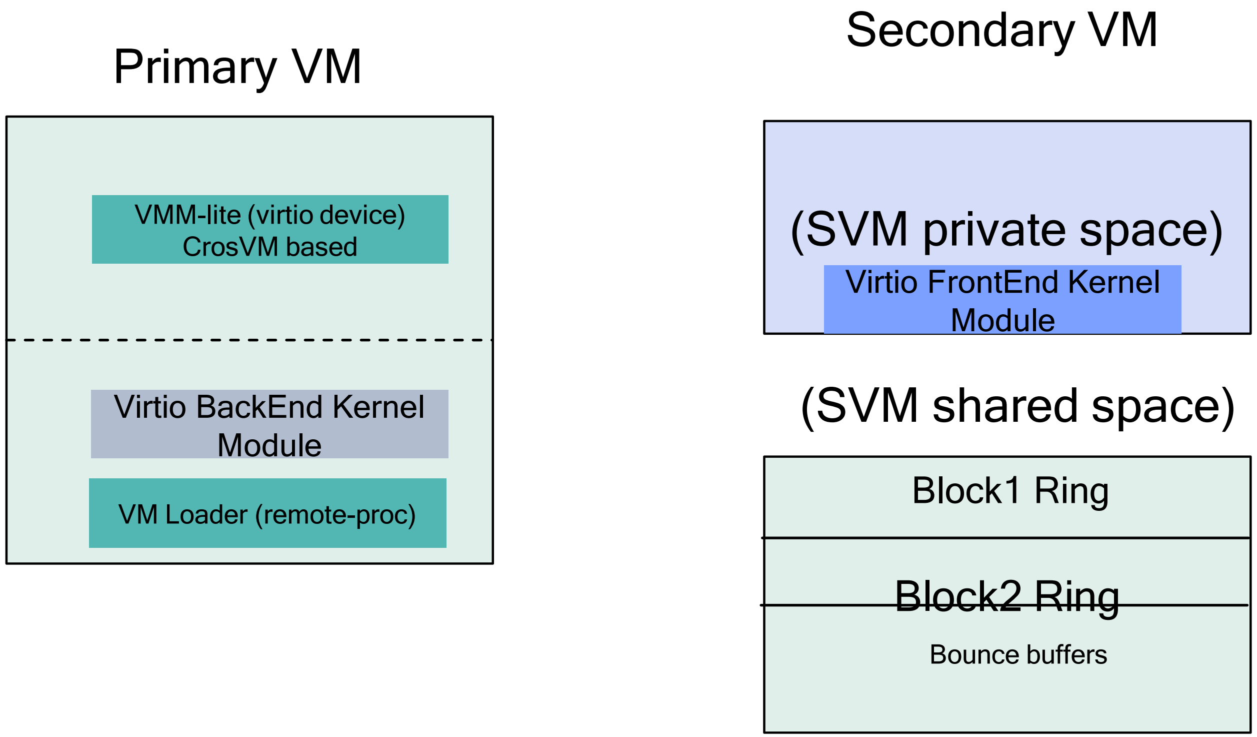


✓ Virtio Backend driver has full access to guest OS memory.

✗ Virtio Backend has NO access to guest OS memory

✗ **No ready-to-use backend drivers**

Virtio Usage



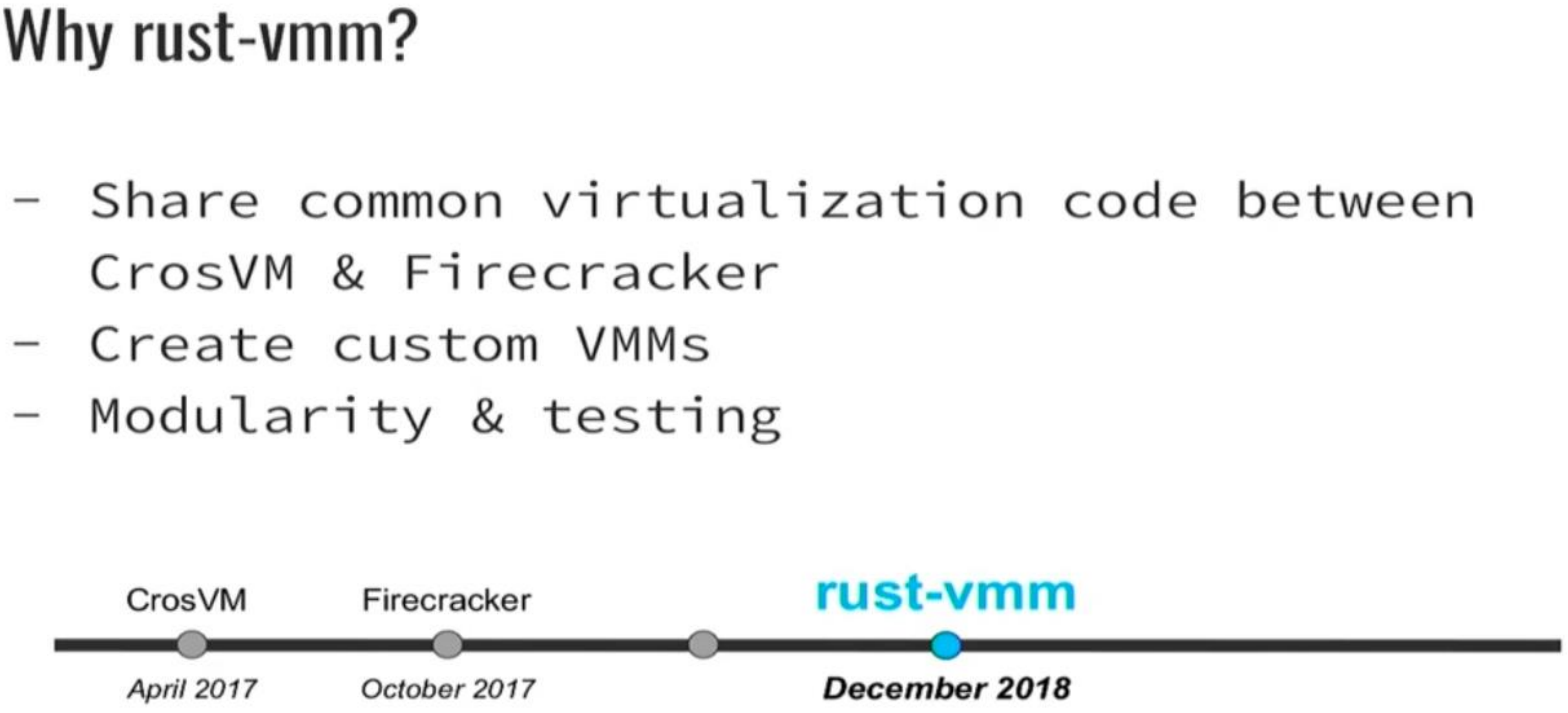
- Kernel-space VM loader (remote_proc/PIL)
- VMM required to host only virtio device backends

Backend Selection

- Choices Evaluated
  - LKVM – not production ready
  - Qemu – Complexity
  - ACRN
  - Rust-VMM
  - CrosVM
- Why we went with CrosVM?
  - Promise of RUST language to avoid memory-related bugs
  - Adoption of CrosVM in Android
- Future Plans
  - Evaluate Rust-VMM and adopt in scenarios where CrosVM may not be feasible

Rust Experience So far

- **Takeaways**
  - Modified CrosVM undergoing product adoption
  - Has been relatively “easy” to make required changes
- **Observations**
  - Initial learning curve - ~1 month
  - Extensive examples/documentation on Internet helped make required changes
  - Android specific build mechanism for Rust
  - Use of traits – came in handy to override the default implementation of some functions (roughly accomplished with function pointers in C)
  - “auto” generated code – when variables go out of scope (closing file descriptors for example)
  - Good reliability – no language related issues found (so far!)
  - IDE – vim integration did not work (for me)
- **Unexplored**
  - Debugging via GDB
  - Profiling

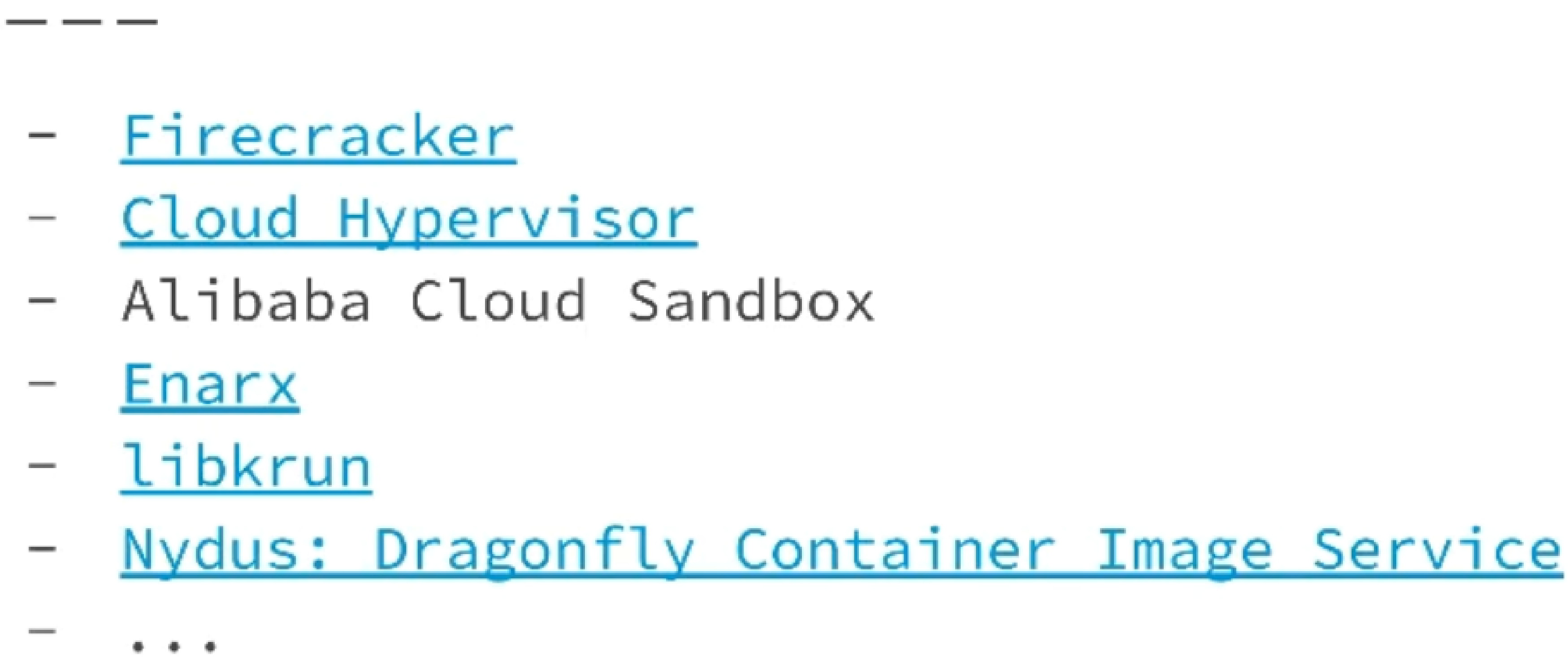


Why rust-vmm?

- Share common virtualization code between CrosVM & Firecracker
- Create custom VMMs
- Modularity & testing

Reference: rust-vmm shared virtualization crates

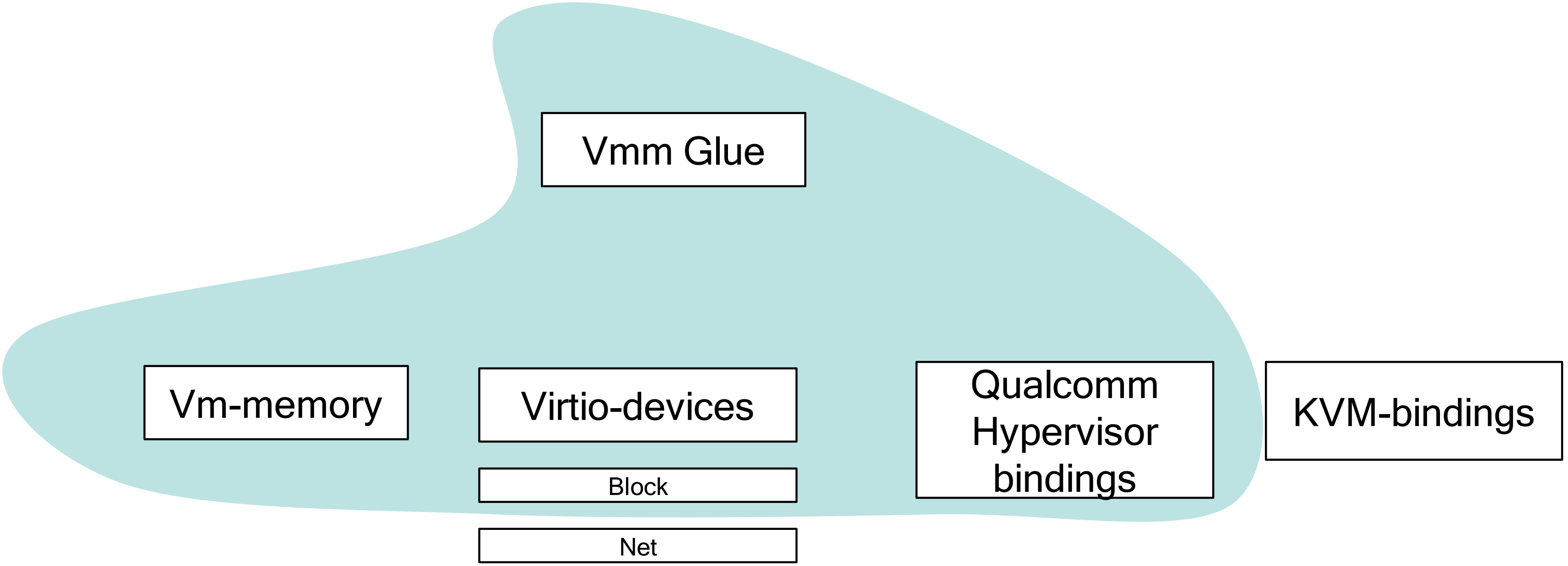
Rust-vmm in Production



- Firecracker
- Cloud Hypervisor
- Alibaba Cloud Sandbox
- Enarx
- libkrun
- Nydus: Dragonfly Container Image Service
- ...

Reference: [2020] Rust-vmm Status Report by Andreea Florescu

Rust-VMM

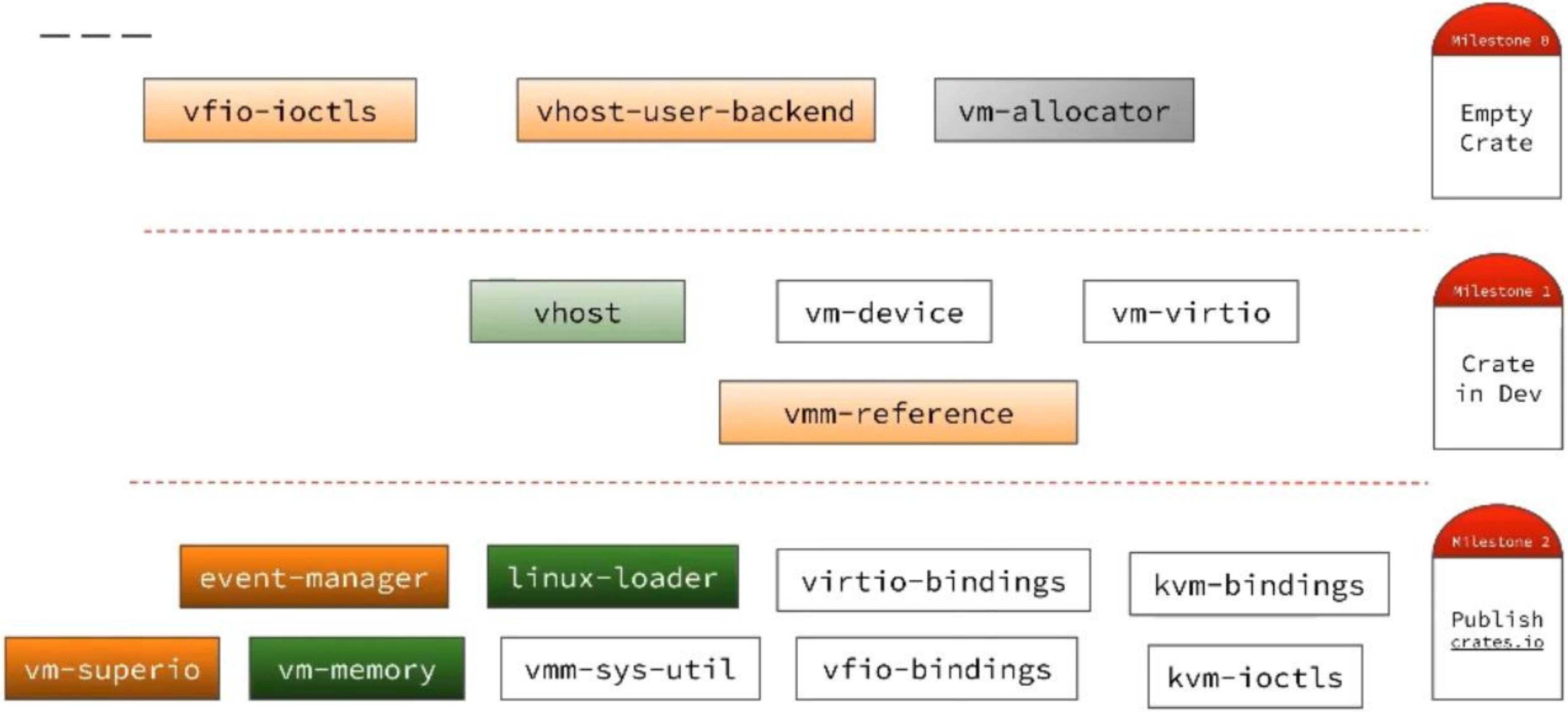

Vmm Glue
Vm-memory
Virtio-devices
Block
Net
Qualcomm Hypervisor bindings
KVM-bindings

Rust-vmm - todo

- Aarch64 support
- Promote required crates to “production-ready” level
- Qualcomm hypervisor bindings

BACKUP

Status 2020 - New Development



vfio-ioctls

vhost-user-backend

vm-allocator

Milestone 0

Empty Crate

vhost

vm-device

vm-virtio

vmm-reference

Milestone 1

Crate in Dev

event-manager

linux-loader

virtio-bindings

kvm-bindings

vm-superio

vm-memory

vmm-sys-util

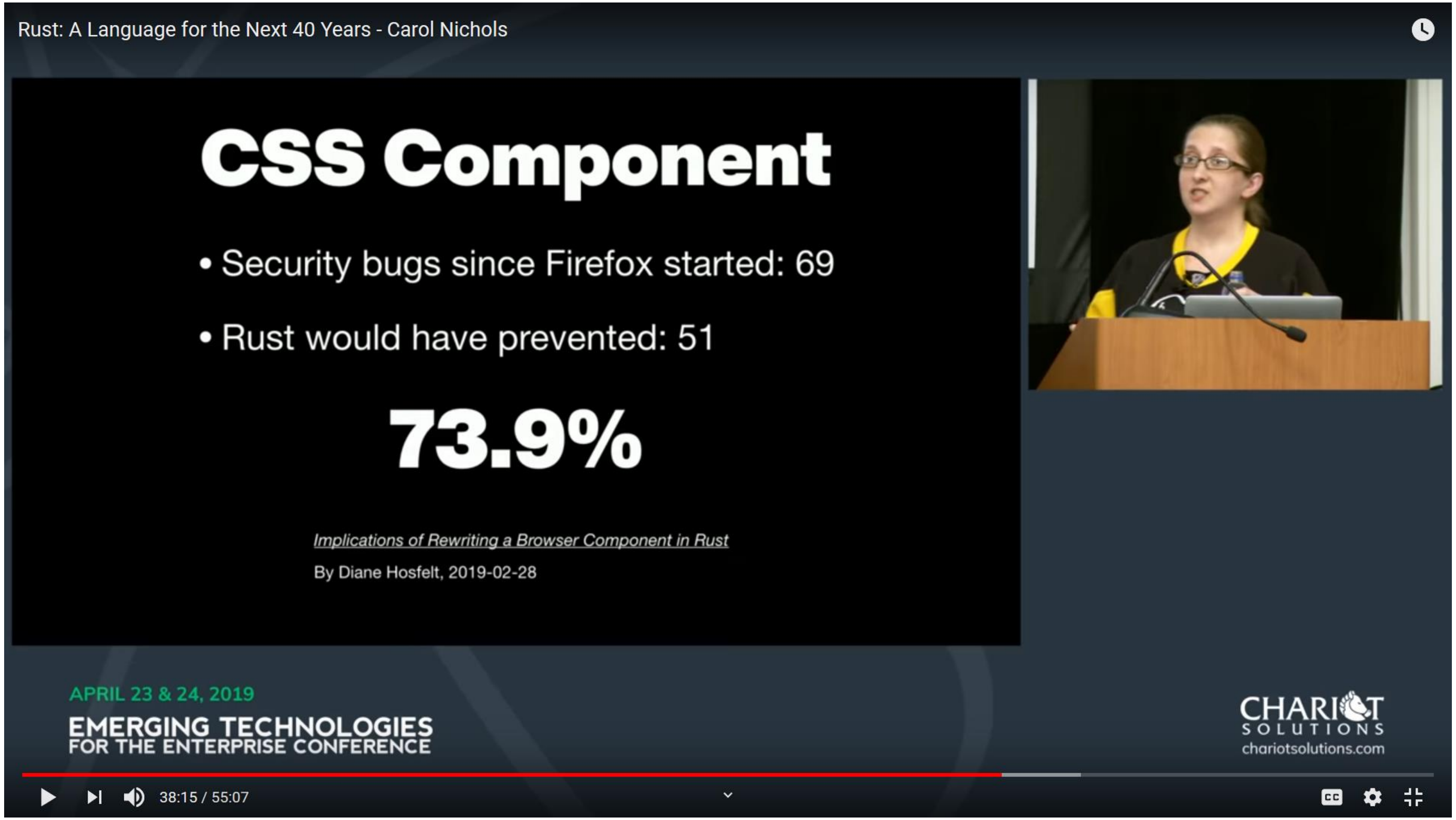
vfio-bindings

kvm-ioctls

Milestone 2

Publish crates.io

Reference: [2020] Rust-vmm Status Report by Andreea Florescu



Rust: A Language for the Next 40 Years - Carol Nichols

CSS Component

- Security bugs since Firefox started: 69
- Rust would have prevented: 51

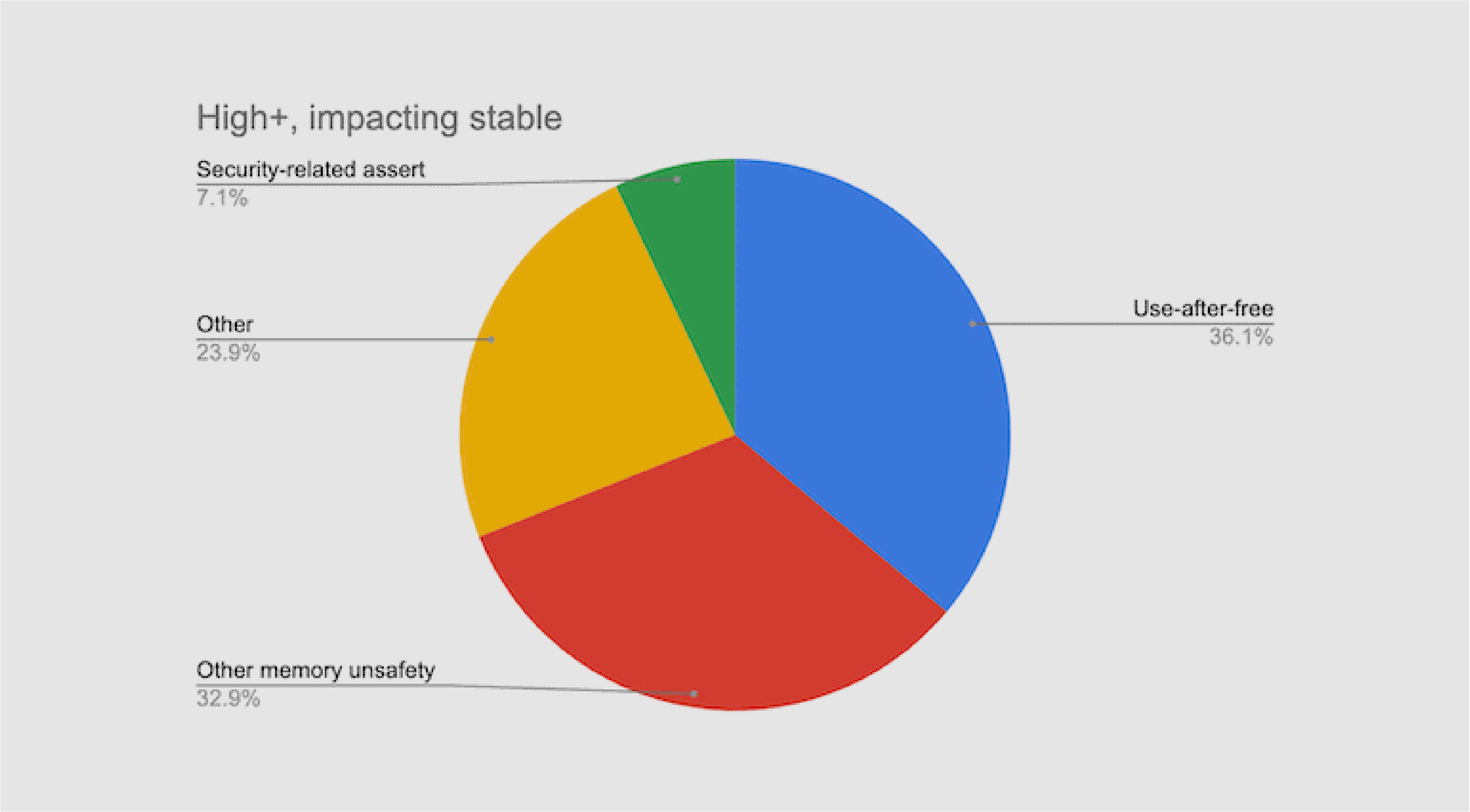
73.9%

Implications of Rewriting a Browser Component in Rust

By Diane Hosfelt, 2019-02-28

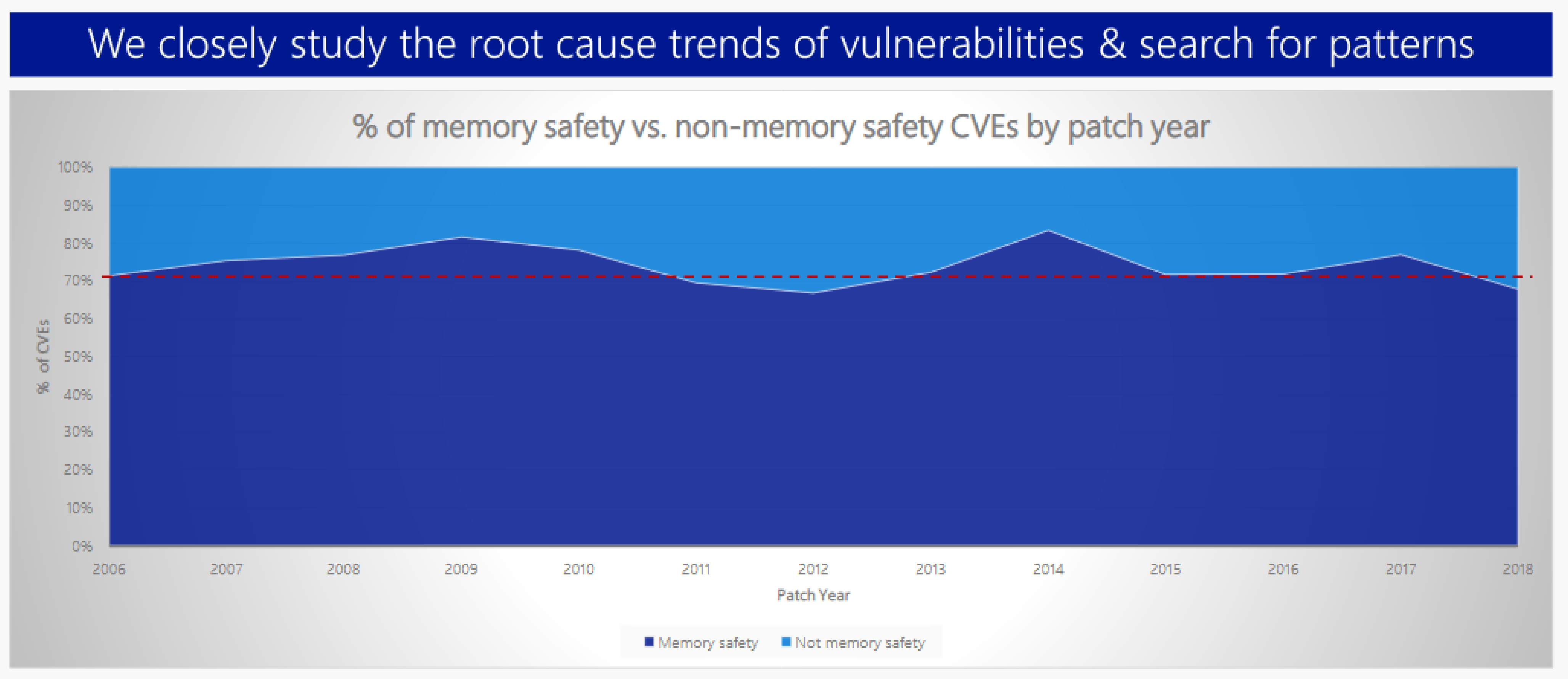
https://www.youtube.com/watch?v=A3AdN7U24iU

Chrome: 70% of all security bugs are memory safety issues



https://www.zdnet.com/article/chrome-70-of-all-security-bugs-are-memory-safety-issues

Microsoft: 70 percent of all security bugs are memory safety issues



We closely study the root cause trends of vulnerabilities & search for patterns

% of memory safety vs. non-memory safety CVEs by patch year

https://www.zdnet.com/article/microsoft-70-percent-of-all-security-bugs-are-memory-safety-issues