



The speed of containers, the security of VMs

[KataContainers.io](https://katacontainers.io)



Contents

Project Overview

Technical Details

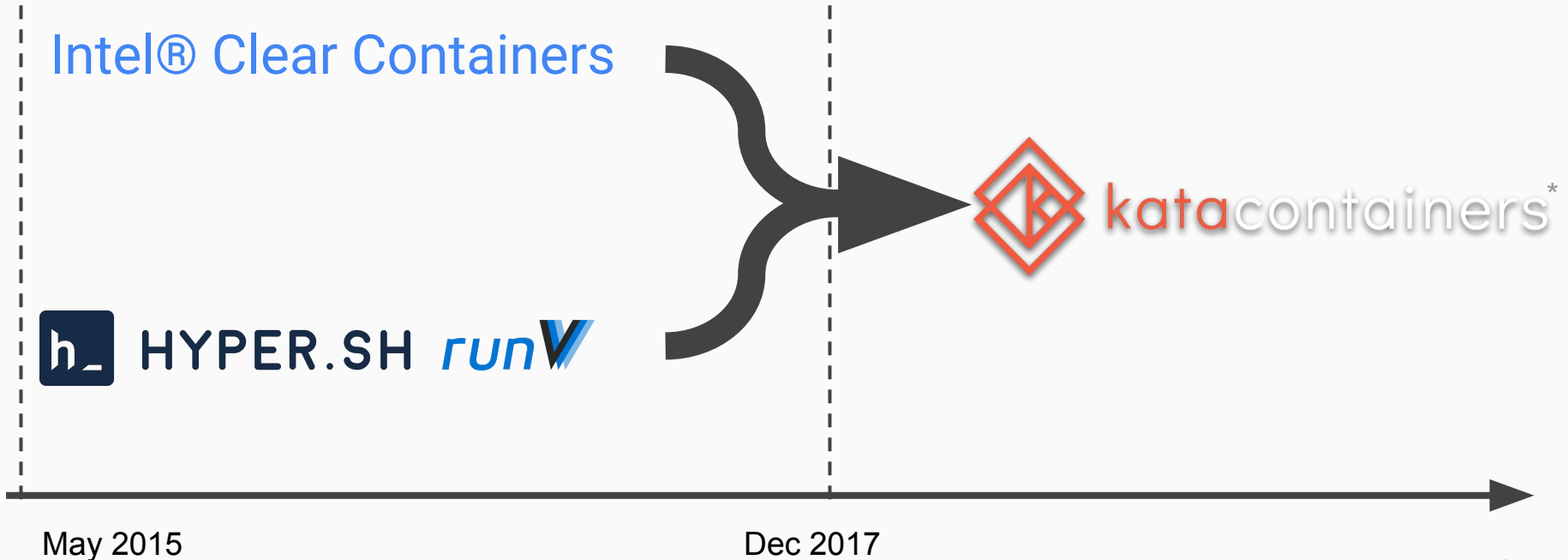
Governance

Get Involved





History



*Other names and brands may be claimed as the property of others.



Technical Vision

- Light and fast VM-based containers
- Merge Intel® Clear Containers and Hyper runV technologies
- Seamless integration with Kubernetes (CRI), Docker and Openstack
- Support multiple architectures (x86 today; others to come in the future)
- Support multiple hypervisors (KVM today; others to come in the future)

Multi Architecture
Multi Hypervisor
Full Hotplug
K8s Multi Tenancy
VM templating
Frakti native support
Traffic Controller net

Direct Device Assignment
SRIOV
NVDIMM
Multi-OS
KSM throttling
CRI-O native support
MacVTap, multi-queue net

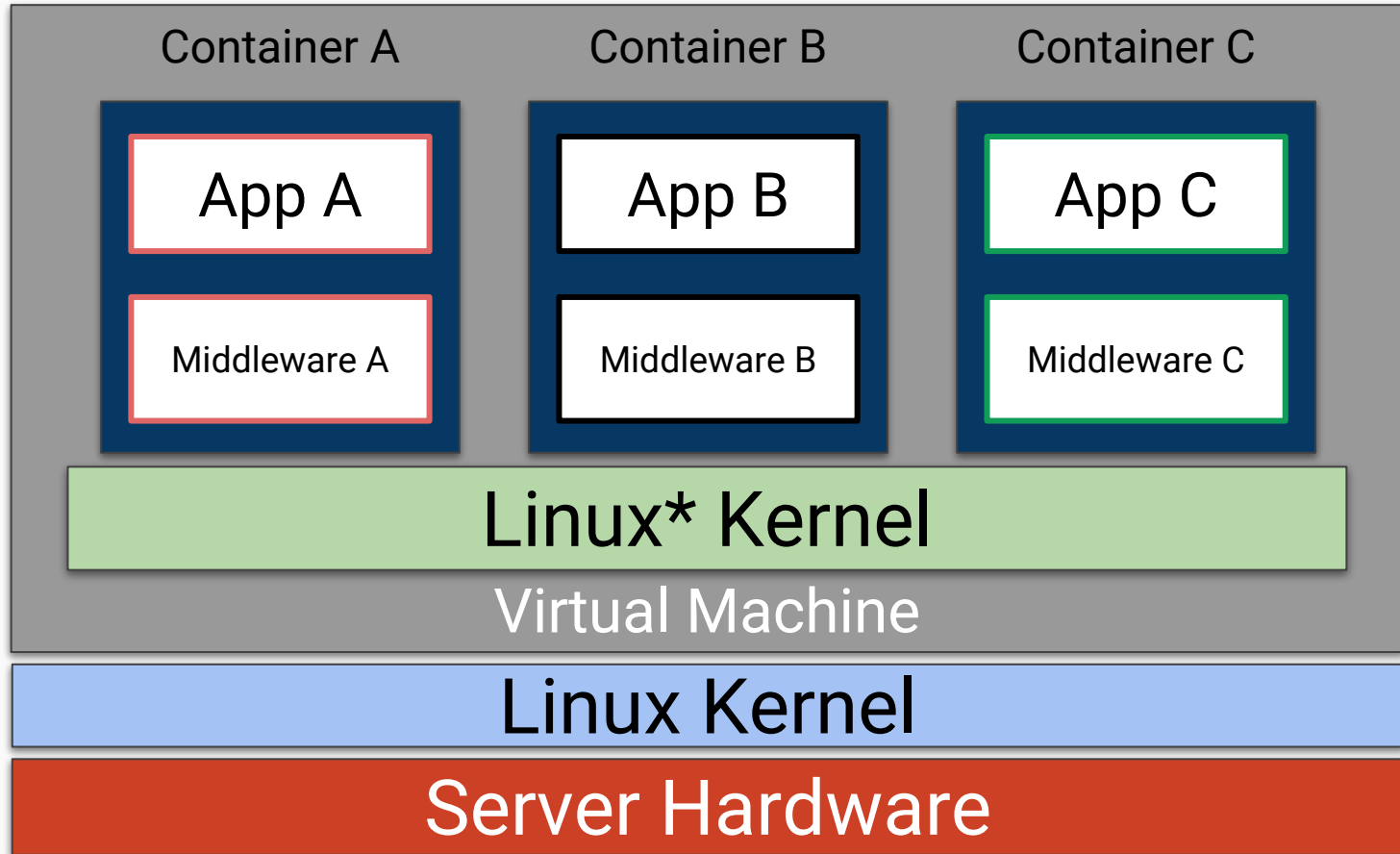




Non-Technical Goals

- Open and vendor-neutral project
- All VM based containers, users and consumers under the same project
- Managed **at** the OpenStack Foundation*
- Independent from the OpenStack* software project

Containers in Cloud

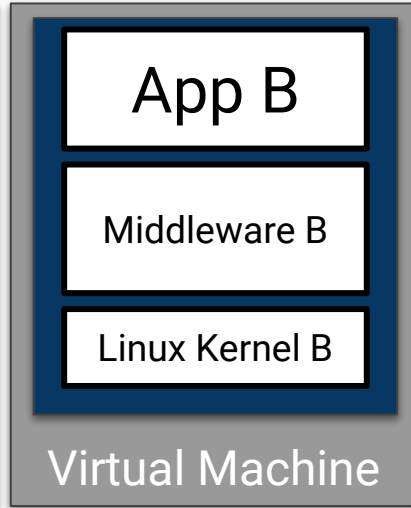


Hypervisor Based Containers

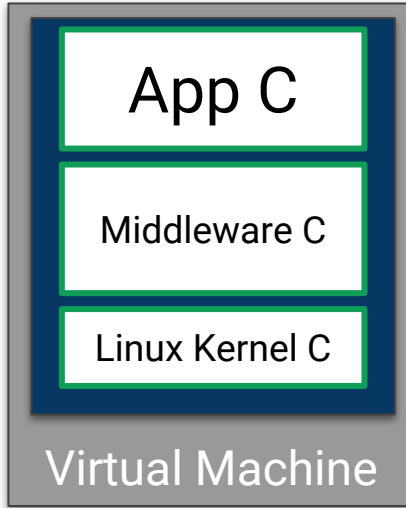
Container A



Container B



Container C



- Each container/pod is hypervisor isolated
- As secure as a VM
- As fast as a container
- Seamless integration with the container ecosystem and management layers

Linux* Kernel

Server Hardware



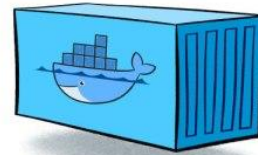
Virtual Machines



Isolation



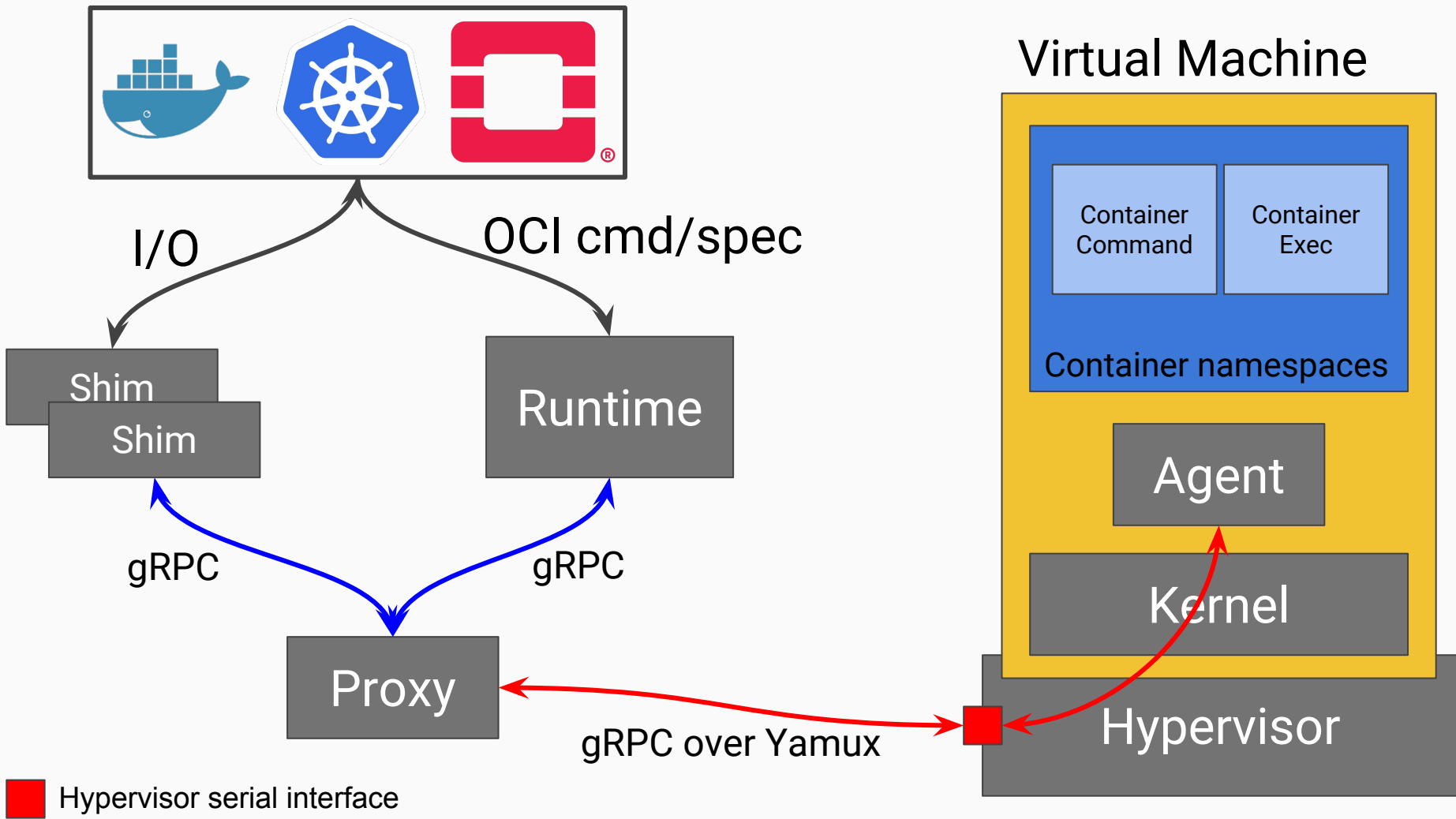
Speed




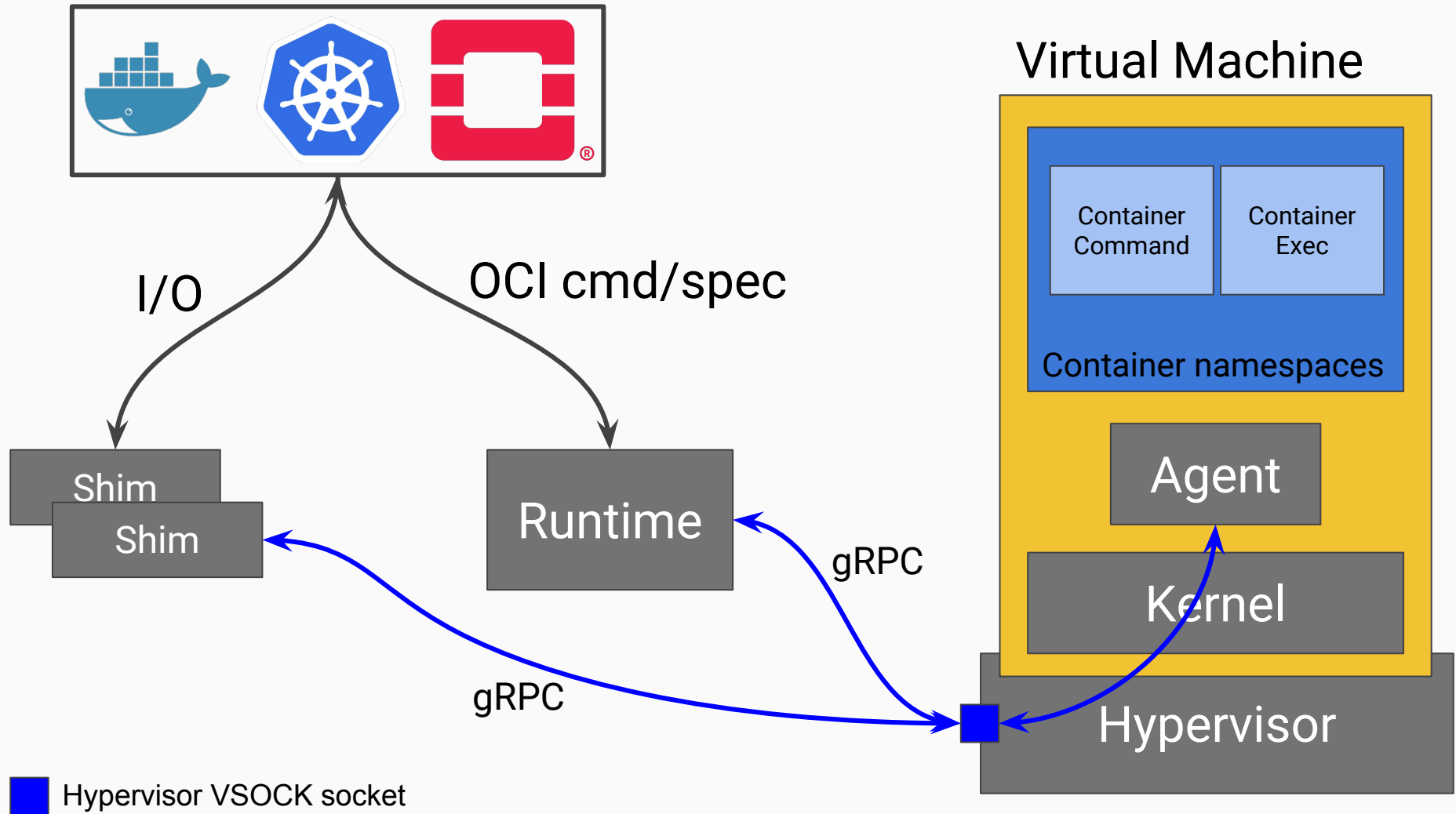


Technical Details





 Hypervisor serial interface





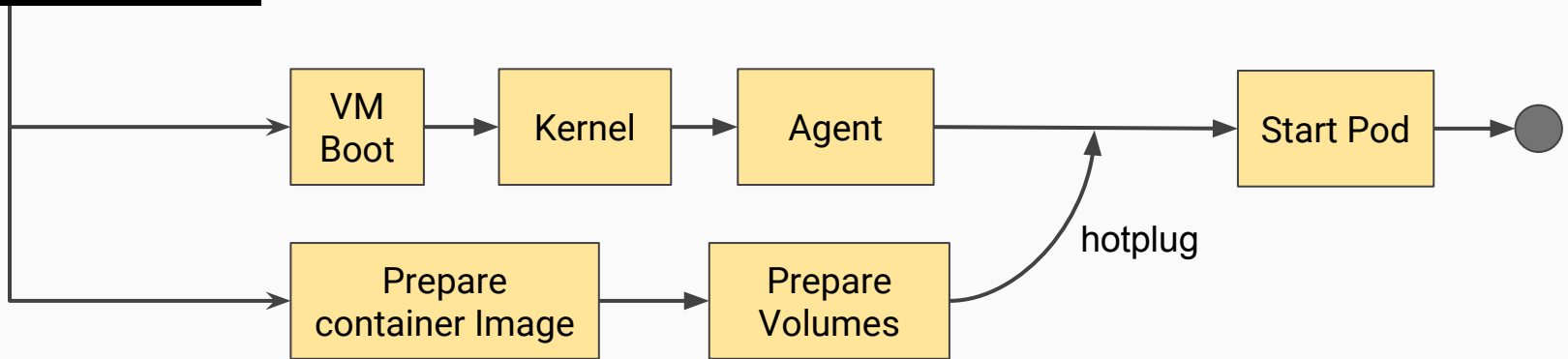
Fast as a Container

Create

Start



```
$ kubectl apply -f nginx.yml
```



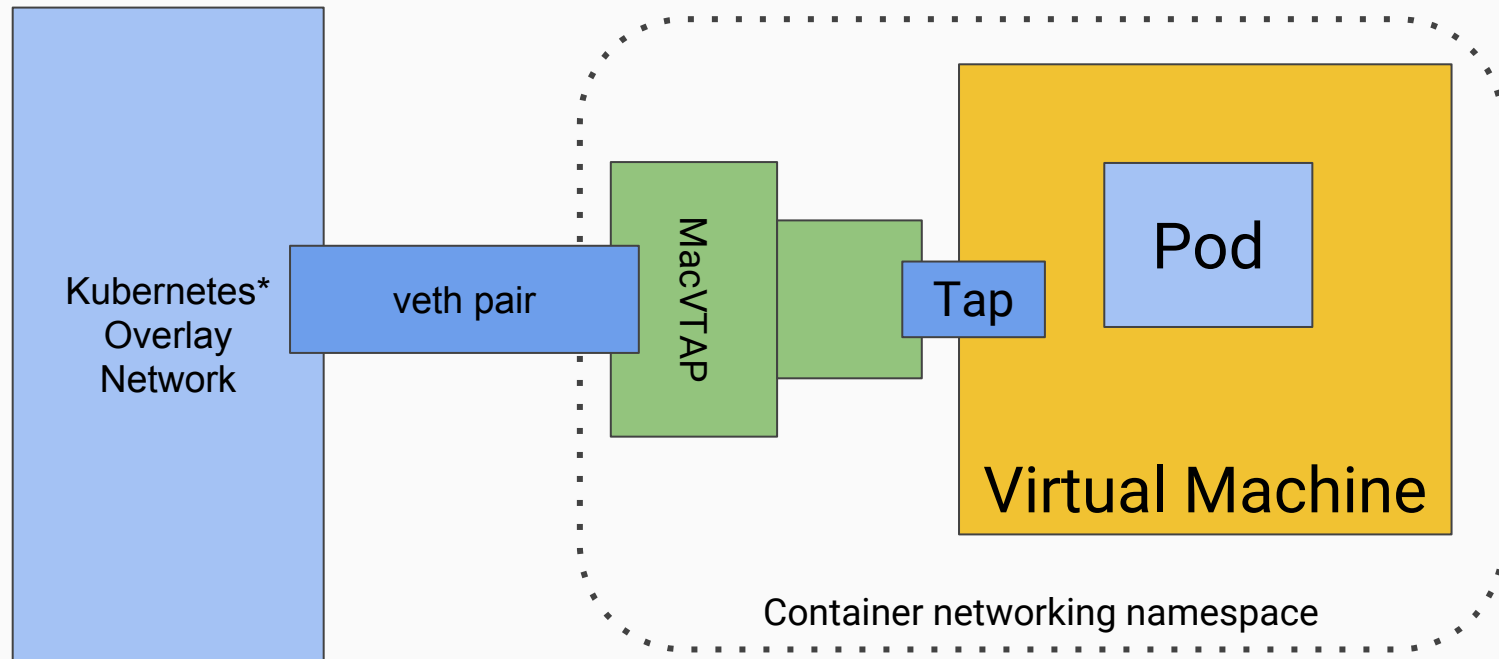


Small as a Container

- Minimize memory footprint
 - Minimal rootfs
 - Minimal kernel
 - VM Template
 - DAX/nvdim
- De-duplicate memory across VMs
 - KSM (with throttling)

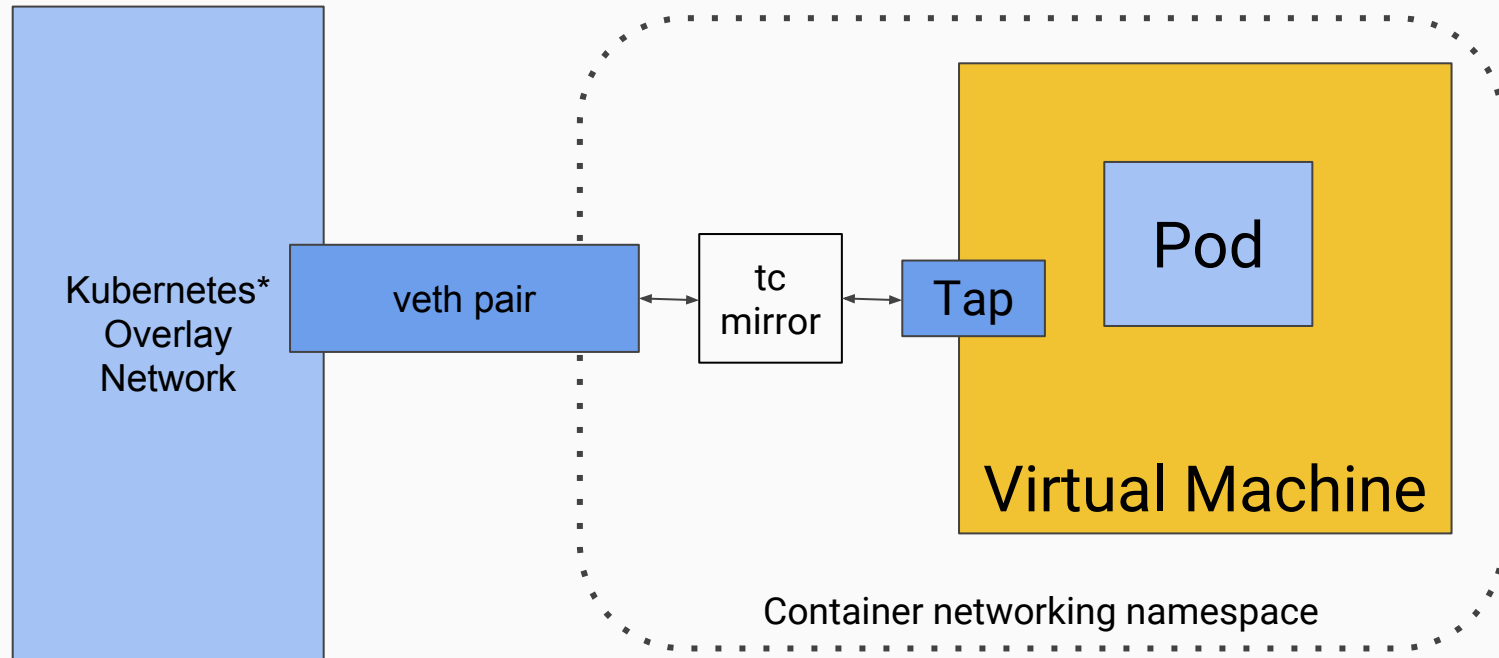


Networking



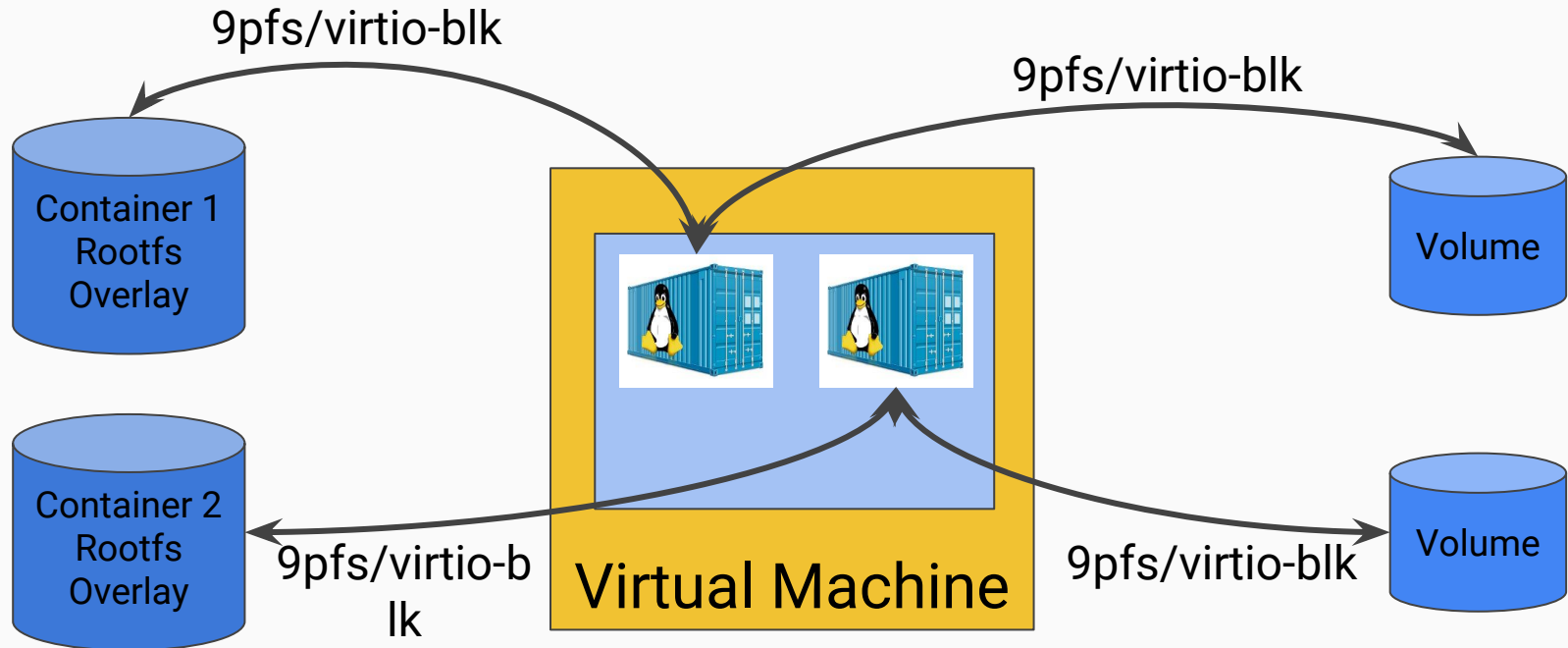


Networking



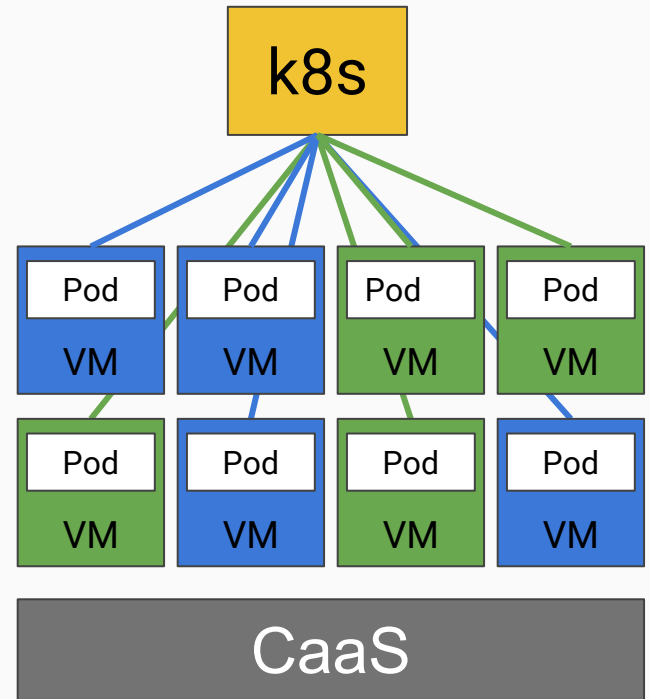
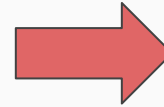
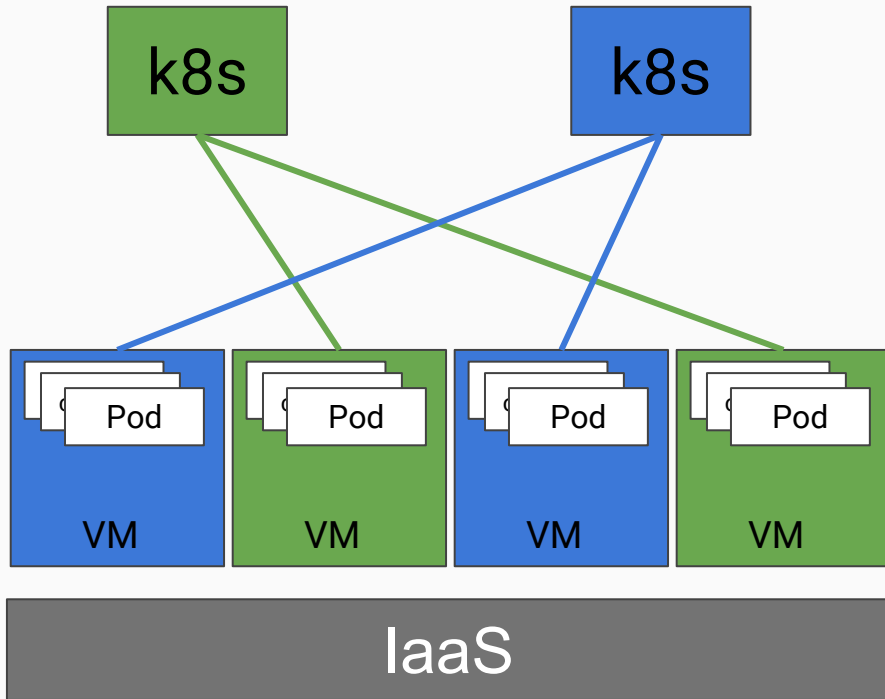


Storage



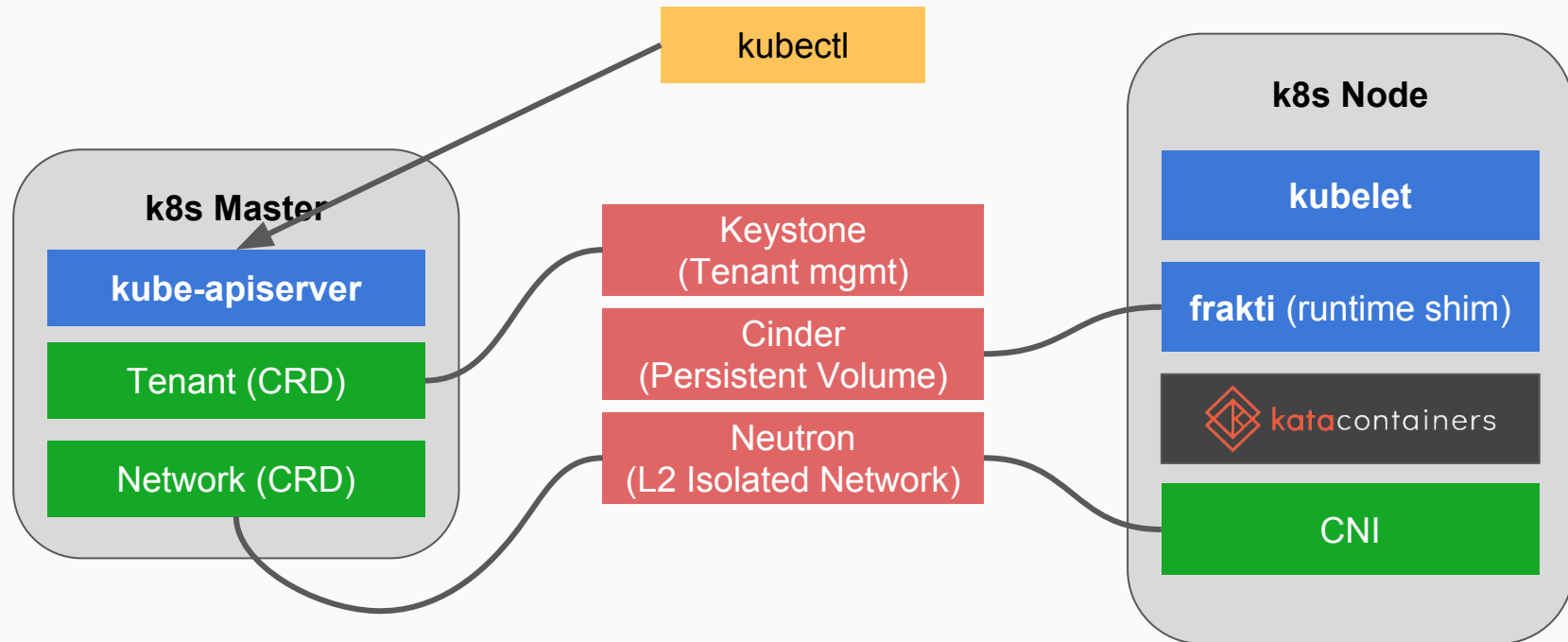


Multi-tenant Kubernetes*





Demo: Stackube - K8S with Hard Multi-tenancy





Project Status - Code

	Status	Current Work
Shim	Initial implementation merged	Terminal size, code coverage
Proxy	Initial implementation merged	Code coverage, functional testing
Agent	Initial implementation merged	Shared PID ns, sub-reaping (agent is PID 1)
QEMU	Vanilla 2.9	QEMU build config
Linux Kernel	4.13.13 + one 9pfs patch	Minimal kernel config definition
OS builder	Initial implementation merged	Initrd/initramfs support
Agent Protocol	V0.0.1 merged (pre-alpha)	More declarative APIs, network APIs improved



Project Status - Code - Runtime

- No runtime code at the moment
- Step 1: Add both Clear Containers and runV
 - Adapted to Kata Containers architecture (gRPC, new components)
 - Both runtimes will work seamlessly with all the Kata Containers components
 - Users can switch runtime implementations transparently
- Step 2: Merge runtimes into one single implementation
- Step 3: Deprecate Clear Containers and runV



Project Status - Documentation and CI

- Documentation
 - Missing at the moment
 - Clear Containers and runV documentation as a backup
- CI
 - Travis based for now: Unit testing only
 - Will move to a nested virtualization enabled and/or bare metal public cloud
 - Functional and integration test
 - From unit testing up to the higher levels of the stack (Kubernetes, OpenShift)



What's Next?

1H'2018 Horizon

- 1.0 Release (parity with RunV and CC 3.0 with upgrade path)
- CRI integration: Frakti, CRI-O, containerd-cri
- OCI runtime spec support for hypervisor based containers
- OSV support
- Documented case studies



Governance





Governance

The Kata Containers project is governed according to the “four opens,”

- open source
- open design
- open development
- open community

Technical decisions will be made by technical contributors and a representative Architecture Committee. The community is committed to diversity, openness, encouraging new contributors and leaders to rise up.



Governance

- **Contributors**
 - At least one github contribution for the past 12 months
- **Maintainers**
 - Active contributor, nominated by fellow maintainers
 - Can merge code
- **Architecture Committee**
 - Take high level architecture and roadmap decisions
 - 5 seats, elected by contributors



Governance

Architecture Committee

- The Architecture Committee is responsible for architectural decisions, including standardization, and making final decisions if Maintainers disagree.
- It will be comprised of 5 members, who are appointed by the Maintainers at launch but **fully elected by Contributors within the first year**.
- The initial Architecture committee members are Samuel Otiz (Intel), Xu Wang (Hyper), Zhang Wei (Huawei) and Tim AllClair (Google).



Governance

Working Committee

- The Working Committee is intended to make non-technical decisions and help influence the project strategy, including marketing and communications, product management and ecosystem support.
- Representatives are expected to be active contributors who are committed to the health and success of the project.
- Recognizing the project will grow and change quickly in the first six months, and in order to encourage diversity and participation, the Working Committee will be forming up and finalizing it's structure after the project launch.
- Initial appointed members include Amy Leeland (Intel) and James Kulina (Hyper). During this initial period, the participants will appoint a leader to help organize and run regular meetings, coordinate the various work streams and help define the long-term structure.
- The initial task will be to determine 2018 plans and appropriate work streams, working groups and funding to execute on those plans.
- **Anyone can join! Get involved in the #working-committee channel on Slack: bit.ly/KataSlack (case sensitive)**



Get Involved





Contribute

- Code and documentation hosted on <https://github.com/kata-containers/>
- Major releases managed through Github* Projects
- Intel (Intel® Clear Containers) & Hyper (runV) contributing initial IP
- Apache 2 license
- Slack: katacontainers.slack.com
- IRC: #kata-dev@freenode
- Mailing-list: kata-dev@lists.katacontainers.io



Where To Contribute?

- **Code**

- Unit tests for agent, shim and proxy
- PR reviews (agent, shim)
- Osbuilder support for more distros

- **gRPC**

- Input needed: Do we cover it all?
- API documentation

- **Documentation**

- Getting Started guides
- Code documentation

- **Features Requests**

- Open issues and PRs for any feature that you'd like to get from Kata Containers



Community

- You do not need to be an Individual Member of the OpenStack Foundation in order to contribute, but if you want to vote in the annual OpenStack Foundation Board of Directors election, you may join: openstack.org/join
- If you are contributing on behalf of an employer, they will need to sign a corporate contributor license agreement, which now covers all projects hosted by the OpenStack Foundation (same model as Apache and CNCF)
- Independent contributors may be submitted with a sign off header under the DCO



Communication

- KataContainers Slack bit.ly/KataSlack (case sensitive)
- #kata-dev IRC freenode (Slack and IRC have a gateway to share messages)
- Mailing Lists: lists.katacontainers.io
- Twitter: @KataContainers
- Email: info@katacontainers.io



Thank you!

[KataContainers.io](https://katacontainers.io)

