cloudbase solutions

Kubernetes Multitenancy

Alessandro Pilotti | CEO @ Cloudbase Solutions

Cloudbase

+ Developers and maintainers of everything Windows / Hyper-V in **OpenStack**
+ Part of Kubernetes **#sig-windows**
+ Developers and maintainers with Microsoft of the **Kubernetes** Windows **CI**
+ Developers of **OVS/OVN** Windows components as part of the OVS community
+ **DC/OS** Windows CI
+ Founders of the **Insula** project
+ Offices in Timisoara, Bucharest, Iasi (Romania), planning to expand! ☺

Defining multi-tenancy

Provide **isolation** for multiple **tenants** within a **cluster**

What is isolation?

The ability of partitioning resources (pods in our case) assuring that a resource cannot access another one

Are Linux Docker containers isolated?

+ “Kinda” or in other words, **NO!**
+ Container isolation is based on OS primitives...
  - Cgroups
  - Network namespaces
  - Seccomp
  - LSM (SELinux / AppArmour)
+ ... which are all vulnerable to potential OS security issues
+ Also: what if you need privileged containers?

Network isolation

+ Software Defined Networking
+ VLANS, tunnels (VXLAN, GRE, STT, GENEVE, etc)
+ Kubernetes Network policies
  - Depend on network plugins
+ Network policies implemented by CNI plugin
  - Calico OVS/OVN

Kubernetes on-premise

+ Clusters belong to the same company
+ Tenants can be users from different departments (trusted)
+ Content is trusted
+ Security is important but not critical => isolation is not a primary matter

Kubernetes in public clouds

+ Google GCE/GKE
+ Microsoft ACS/AKS
+ Amazon EKS
+ ...

Users are **untrusted**, content is **untrusted** => isolation matters

K8s policies for isolation

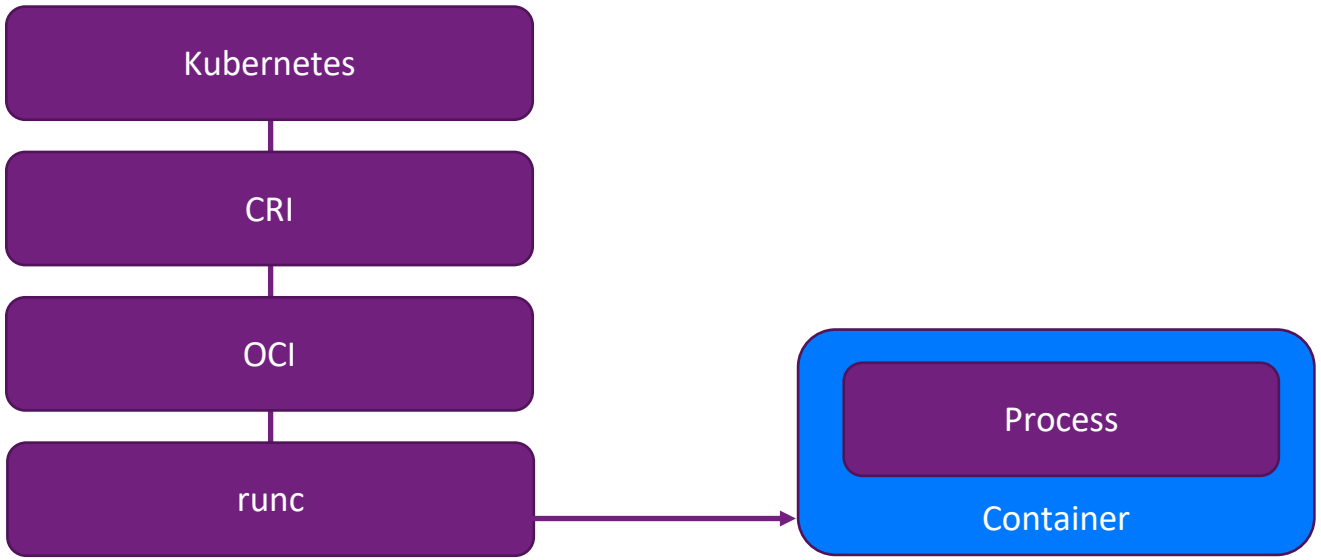
+ Role Based Access Control (RBAC)
+ Admission controllers: PodNodeSelector, ResourceQuota
+ PodSecurityPolicy
+ NetworkPolicy
+ SchedulingPolicy

How to achieve isolation?

+ A common one in current HW is good old virtualization
+ There are other solutions as well, see gvisor
+ Modern x86 CPUs include feature to handle isolation at the HW level:
  - Intel: **VT-X, VT-d**
  - AMD: **AMD-V, IOMMU**

  ARM (ARMv7-A and above)

A quick recap on the architecture!



cloudbase solutions

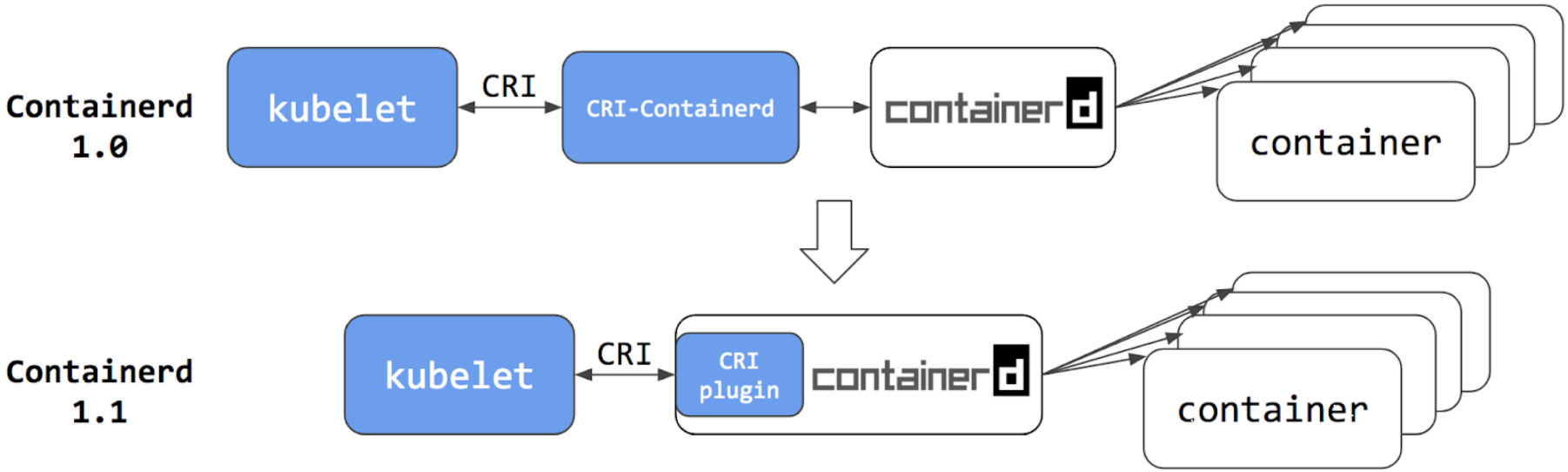
OCI runtime-spec

+ LF effort to standardize container runtimes by providing a common interface.
+ State and lifecyle operations (create, start, kill, delete), Hooks
+ **Runc**, a CLI tool for running containers, is part of this effort
+ OCI includes also **image-spec** for standardizing image formats
+ Uses: **docker, containerd, gVisor, railcar**

Kubernetes CRI – Container Runtime Interface

+ An interface to be able to use multiple container runtimes
+ Docker CRI shim
+ containerd
+ cri-o
+ Rktlet (rkt)
+ frakti

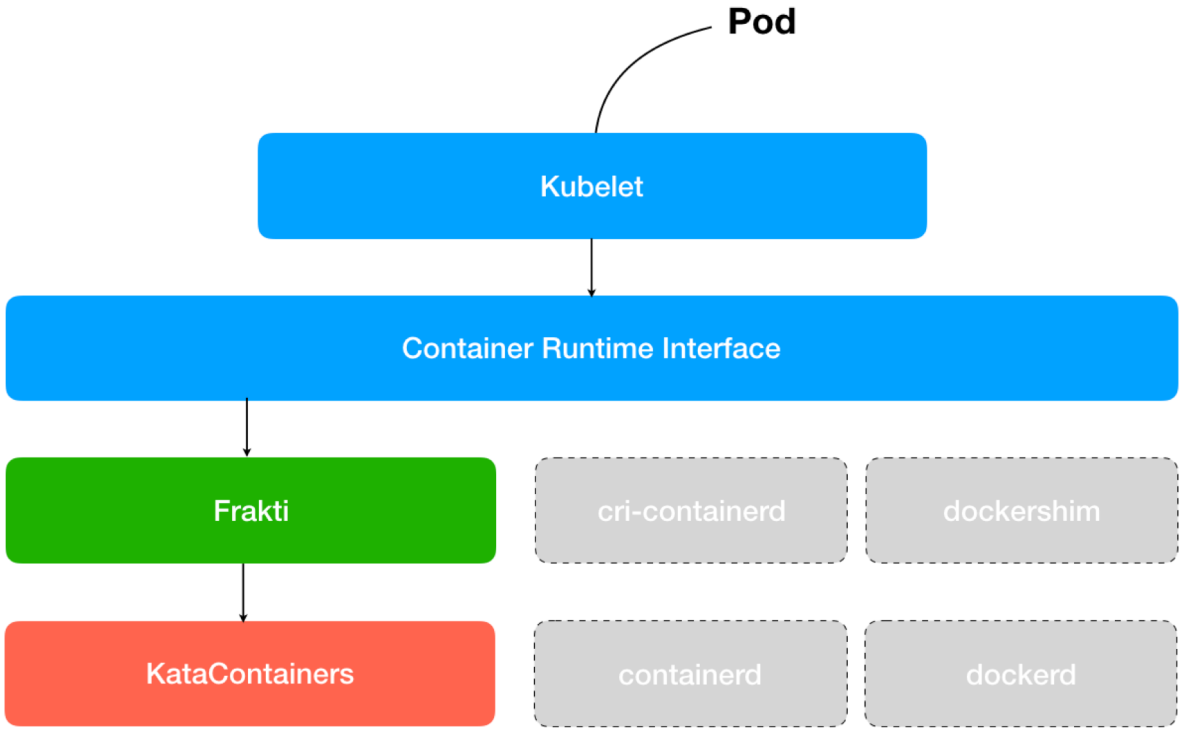
Containerd CRI plugin (new)



+ No need for CRI-Containerd anymore (k8s >= 1.10)

cloudbase solutions

Frakti

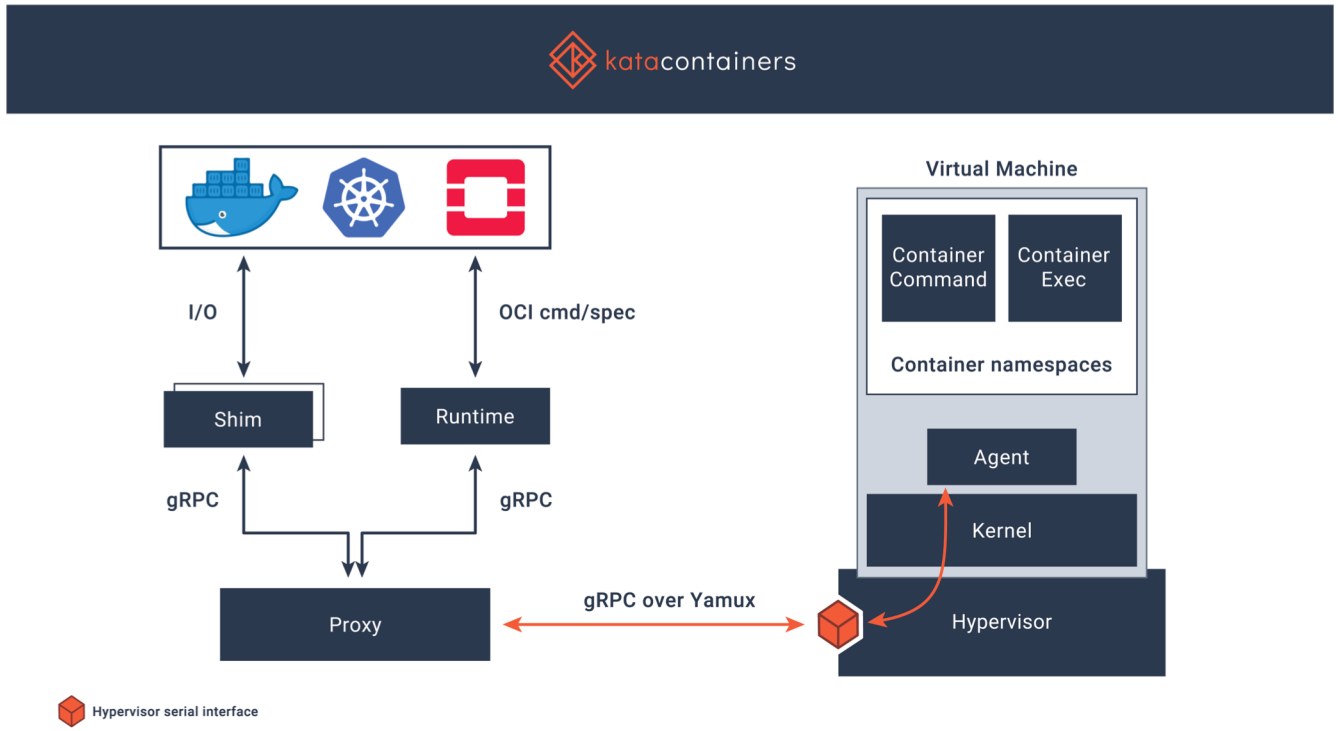


cloudbase solutions

Kata containers

+ Intel Clear Containers
+ Hyper.sh
+ Joint effort under the OpenStack foundation
+ 1 pod = 1 VM
+ VMs are lightweight
+ Various tricks are in place to optimize performance
+ Implements OCI, integrates with **frakti, containerd, CRI-O**

Kata + Kubernetes



Proxy needed only when VSOCK not available

Kata containers components

+ kata-agent
+ kata-runtime
+ kata-shim
+ kata-proxy
+ kata-ksm-throttler

+ Kata doesn’t have it’s own hypervisor (more ion this later!)

Kata containers performance

+ Spawning a VM adds a performance overhead
+ Some ways to mitigate that:
  - KSM
  - Fast VM - templating
  - Fast VM – hotplug

Kata containers devices

+ Ideally, you want to avoid emulation
+ VirtIO
+ HW passthrough
+ SR-IOV
+ Macvtap

Kata containers demo

Does Kata isolation have weak points?

+ The virtualization stack security depends on its components
+ Hardware security flaws: meltdown / spectre!
+ Hypervisor security flaws
+ Kata uses QEMU today. In OpenStack's security guide's words:

„Putting all of this together, QEMU has been the source of many security problems, including hypervisor breakout attacks."

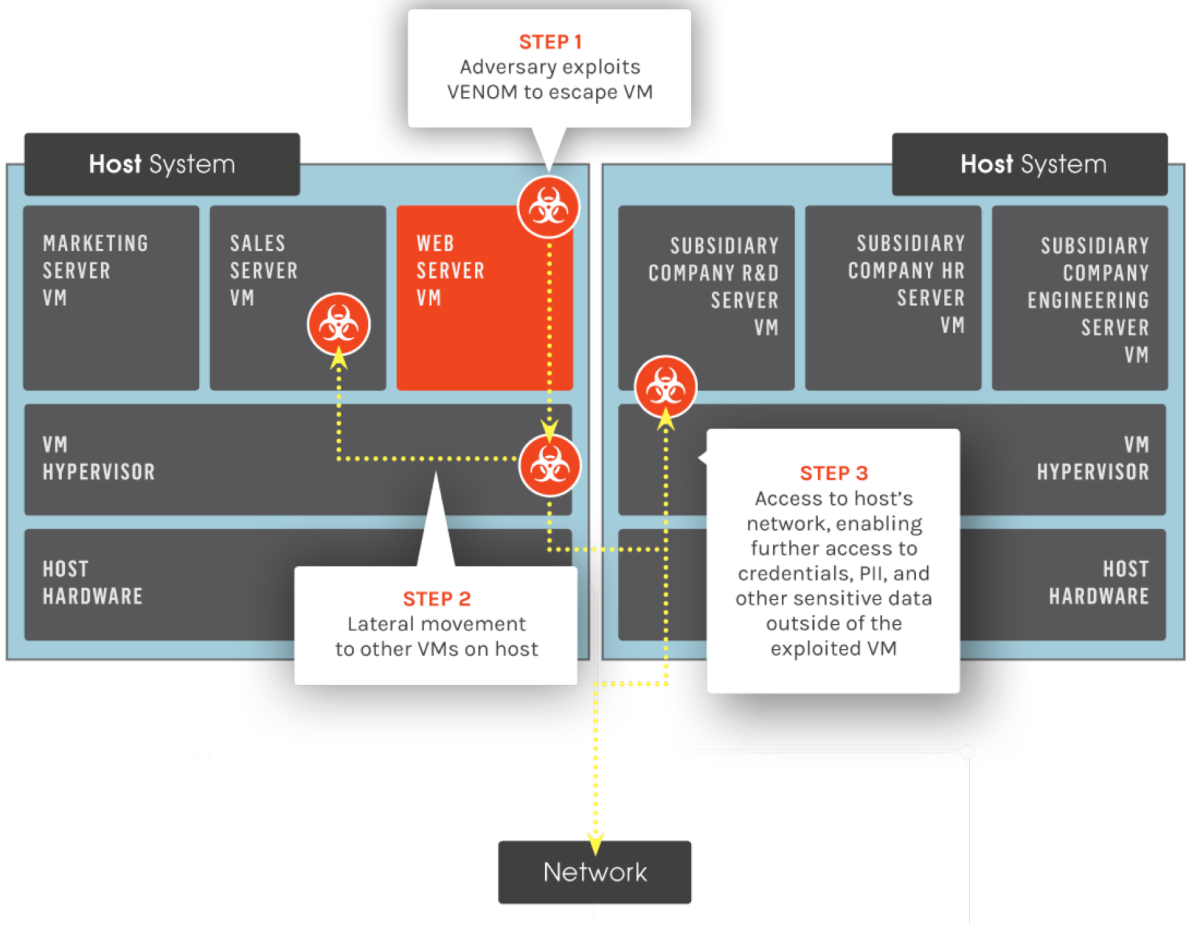
Why is QEMU insecure?

+ Large monolithic project, very difficult to maintain
+ Large attack surface
+ Single executable, usually running with high privileges

Example: VENOM

Vulnerability in the floppy disk controller!

Wait, a floppy disk in a cloud???



cloudbase solutions

What are the alternatives?

+ Hardening QEMU is a good step, but not enough
+ There are various ideas in the community, mostly PoCs:
+ Intel NEMU: https://github.com/intel/nemu
+ CrosVM: https://chromium.googlesource.com/chromiumos/platform/crosvm/
+ gVisor: https://github.com/google/gvisor

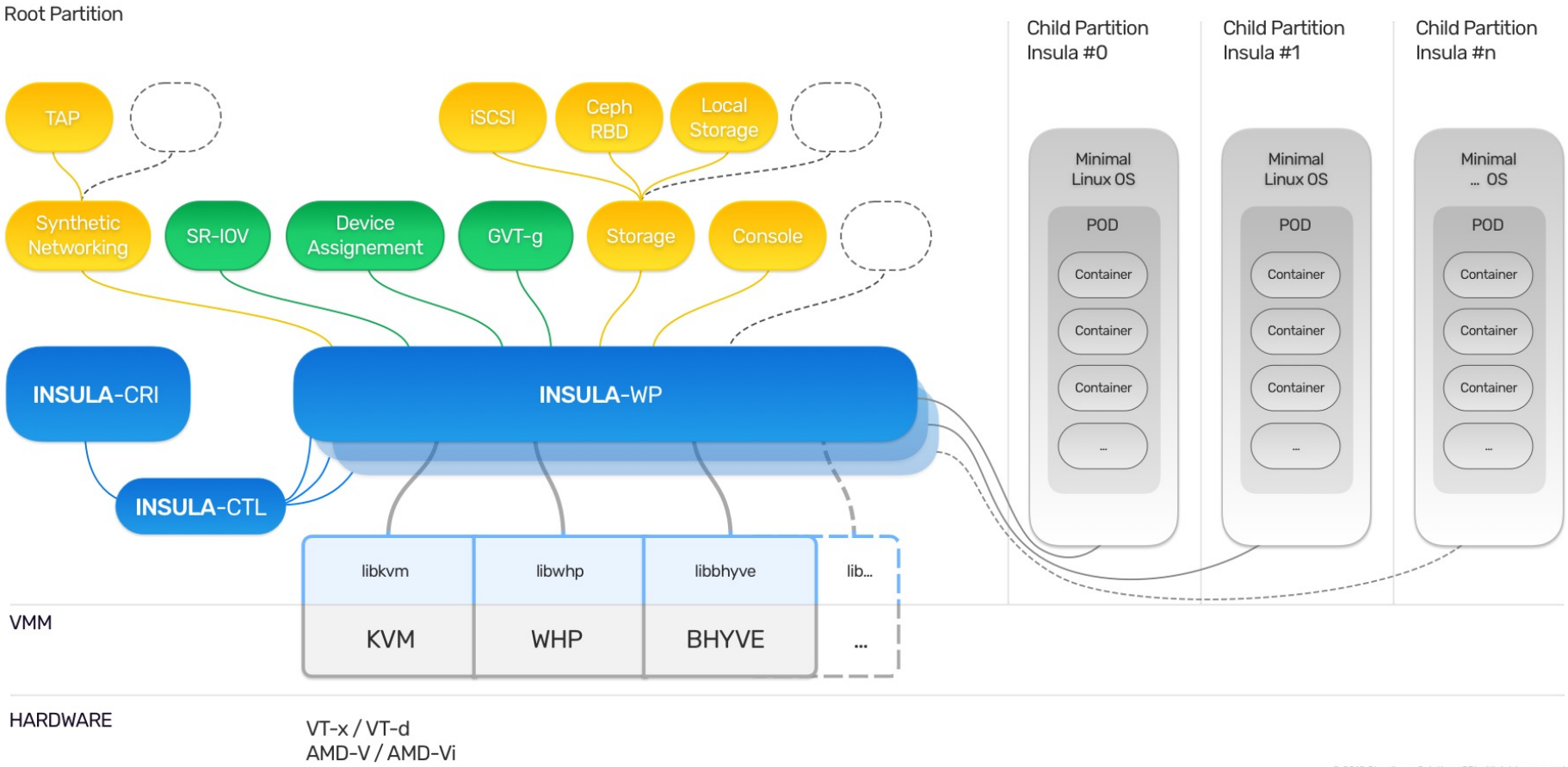
Time for a new project: Insula

+ Lightweight: designed for containers and serverless
+ Portable: KVM (Linux) and WHP (Windows)
+ Minimal device emulation
+ Modular
+ Support device assignment (VT-d, SR-IOV, FPGAs...)
+ Support Network Block Devices (RBD, iSCSI, etc) via modules
+ Integrate with SND like OVS/OVN
+ "isolation" comes from Latin "**insula**" (island)

Insula: security by design

+ Jailed process isolation per each device
+ VTL trustlets / SGX/VSM, etc
+ Reduced attack surface in the hypervisor by running MMIO, IOPort exits in userspace
+ Written in Rust

Insula: project diagram



Why Rust?

+ Kubernetes, Docker, etc are written in Go, why Rust?
+ Rust provides memory safety without a GC
+ Excellent FFI (C library calls) integration and support for “unsafe” code
+ Small runtime
+ Almost as fast as C/C++, with the added security

Some additional resources

+ https://blog.jessfraz.com/post/hard-multi-tenancy-in-kubernetes/
+ Nice Kubernetes multi-tenancy Kubecon session: https://www.youtube.com/watch?v=xygE8DbwJ7c
+ Nice Kata containers intro session:
+ https://www.youtube.com/watch?v=vK_gdy2kdPM
+

Insula project status

+ Early design stages!
+ Completed: libkvm and libwhp
+ Insula-core: under development
+ Code: https://github.com/insula-rs
+ Slack: https://insulars.slack.com

cloudbase solutions

libkvm/libwhp demo

Get in touch

Twitter: @cloudbaseit

http://www.cloudbase.it/coriolis
http://ask.cloudbase.it

Questions?