

KubeCon | CloudNativeCon

North America 2026



KubeVirt's Architectural Leap: Pluggability, AI Hardware, and Maturing Governance

Vladik Romanovsky & Itamar Holder, Red Hat



KubeCon CloudNativeCon North America 2026

10 Years of KubeVirt & Graduation



Ten Years of KubeVirt

AUG 2016

First commit

VMs on Kubernetes.

SEP 2019

CNCF Sandbox

Neutral governance.

APR 2022

CNCF Incubating

AUG 2025

Graduation application filed

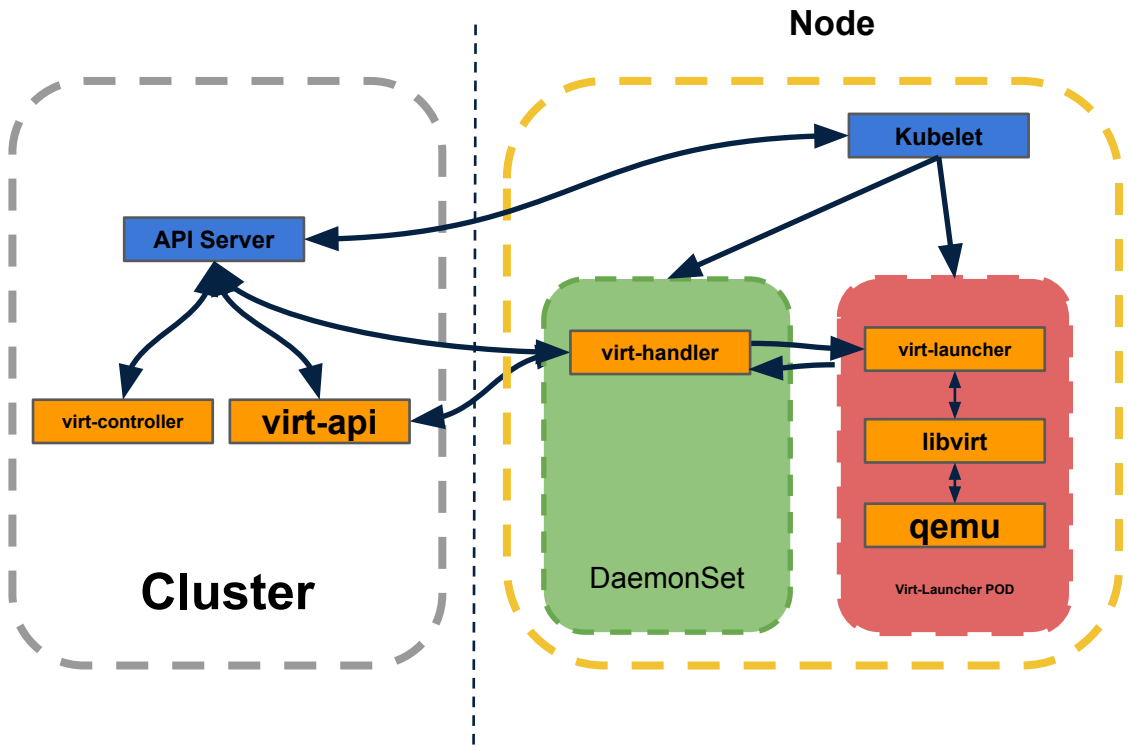
cncf/toc#1822 under TOC review

TODAY

Graduation!



Architecture



Each KubeVirt component is granted only the minimal privileges required, on both the cluster and the host, reducing the impact of a potential compromise.

- Control plane components are granted minimal, justifiable permissions.
- Restricted Virt-Handler
- Virt-Launcher is untrusted
- Greater security and isolation

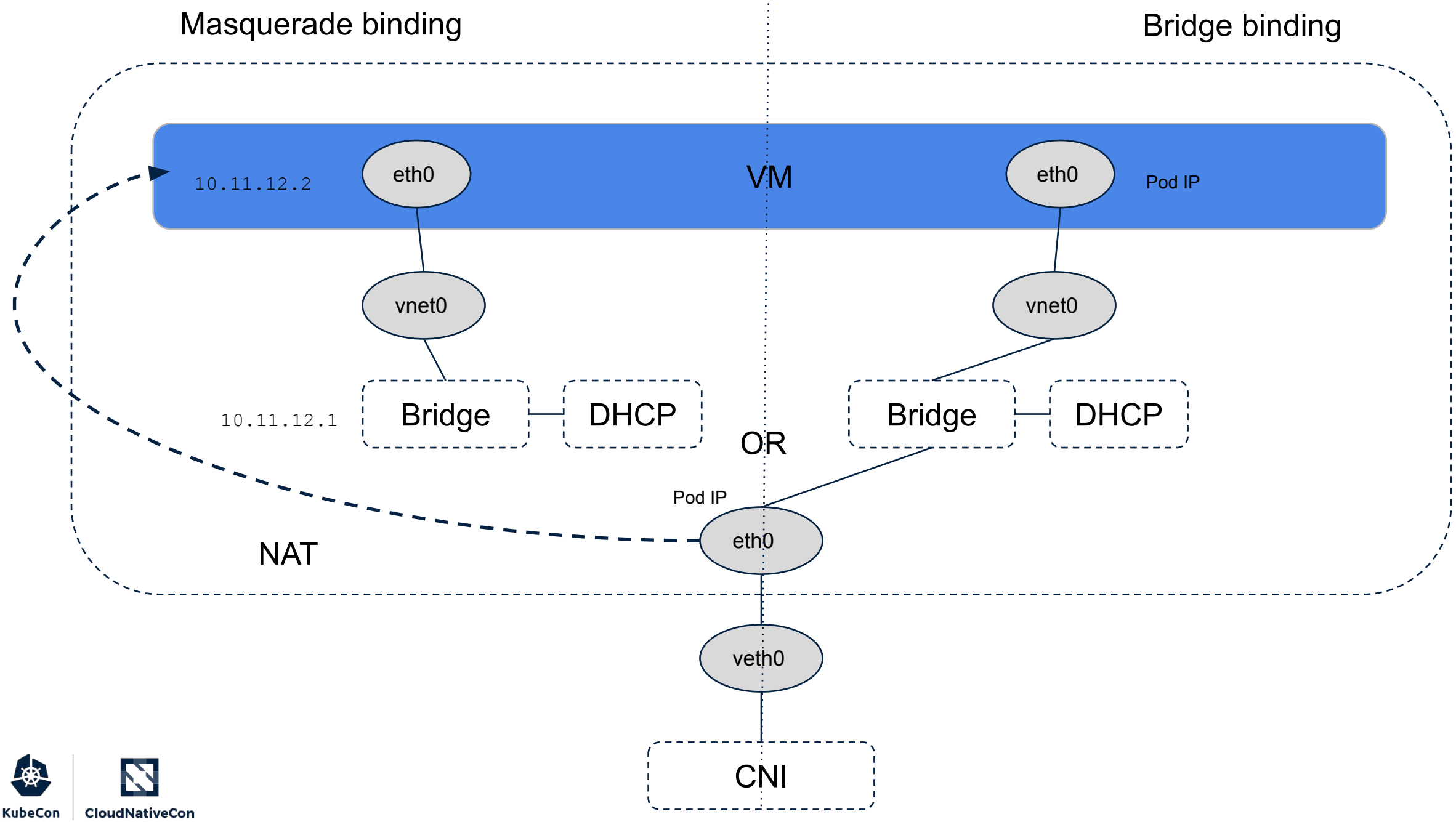
Thesis: Reuse and Contribute Back

Re-use, don't rebuild.

- Using CNI, CSI, DRA and core primitives.
- VM snapshots – thin wrapper over CSI VolumeSnapshot
- CPU pinning & NUMA: handled by the kubelet's CPU, Topology and Memory Managers

Contributed back

- KEP-2400: Node swap, by KubeVirt maintainers.
- KEP-5304: by NVIDIA for KubeVirt's GPU needs.
- KEP-6438: Resize Gates for In-Place Pod Resize
- KEP-5365: ImageVolume with an image digest


Masquerade binding
Bridge binding
10.11.12.2
eth0
VM
eth0
Pod IP
vnet0
vnet0
10.11.12.1
Bridge
DHCP
OR
Bridge
DHCP
Pod IP
eth0
NAT
veth0
CNI

Philosophy

Make it easy to extend KubeVirt!

OUR GROWING ECOSYSTEM

End Users

arm
Aussie Broadband
ByteDance
CHILD RESCUE COALITION
CIVO
CLOUDFLARE
CLOUDRAFT
CoreWeave
Genesis Cloud
intel GAUDI
THE LINUX FOUNDATION
KLLK CODA
NEBIUS
NVIDIA
SBNS PERFORMANCE RENEVATION COMPCANCE
SK telecom

Vendors

Ænix
alauda.io
Deckhouse Kubernetes Platform
EQUINIX METAL
Kube-DC
KUBERMATIC
KUBESPHERE
Microsoft
ORACLE
PLATFORM9
puzl.cloud
RAFAY
Red Hat
spectro cloud
SUSE
swisscom
TrueFullstaq

Integrations

cloudcasa
veeAM Kasten
ISOVALENT
minikube
oVirt
okd
pits does matter
portworx
TRILIO

Projects Building on KubeVirt

Tekton Tasks

CI/CD pipelines for VMs:
create, customize, disk ops, libguestfs, Windows install.

HyperShift / CAPK

KubeVirt VMs as Kubernetes worker nodes.

OSAC

Open Sovereign AI Cloud

VM-aaS, future Cluster-aaS with KubeVirt as subtrace

Flightctl

Edge device management

Runs KubeVirt VMs as standalone virt-launcher Pods

KubeVirt is becoming the shared virtualization layer that higher-level platforms compose with - not just a tool that end users adopt.

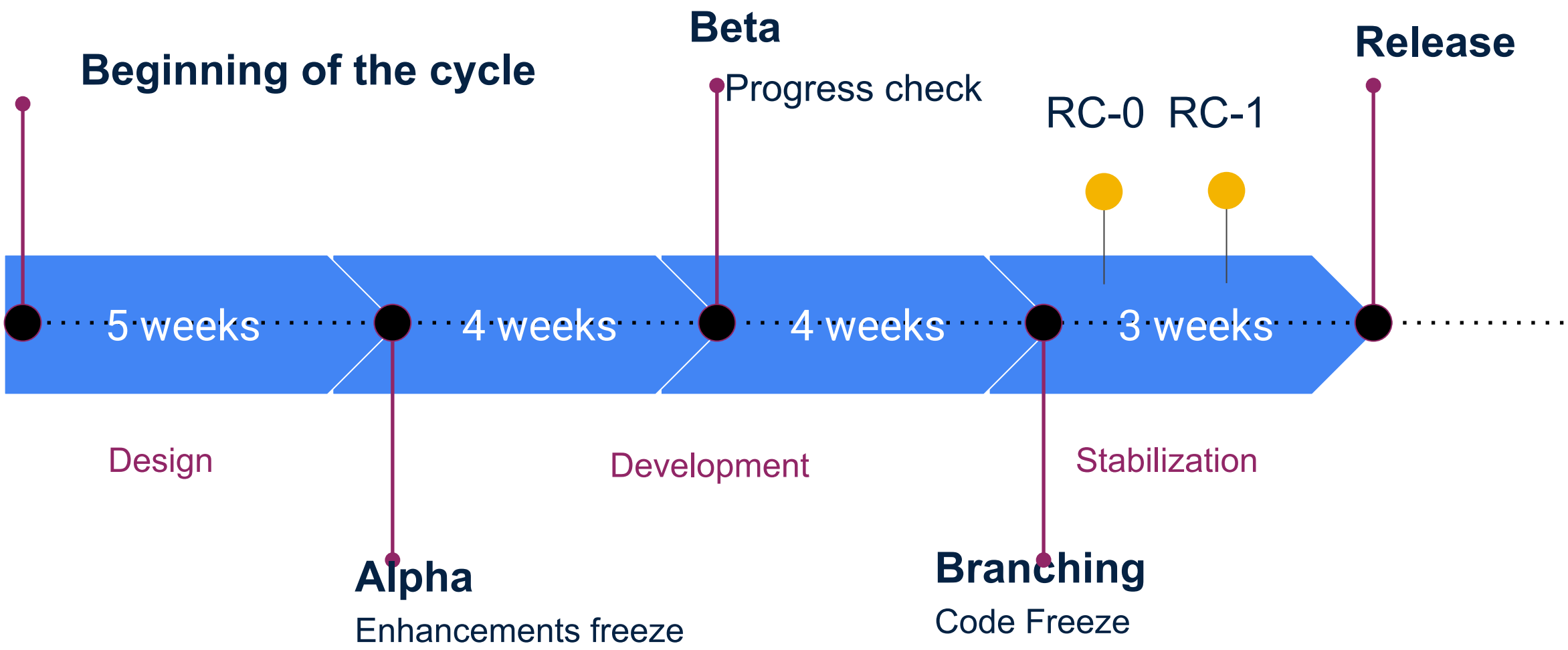
A community that outgrew informal process



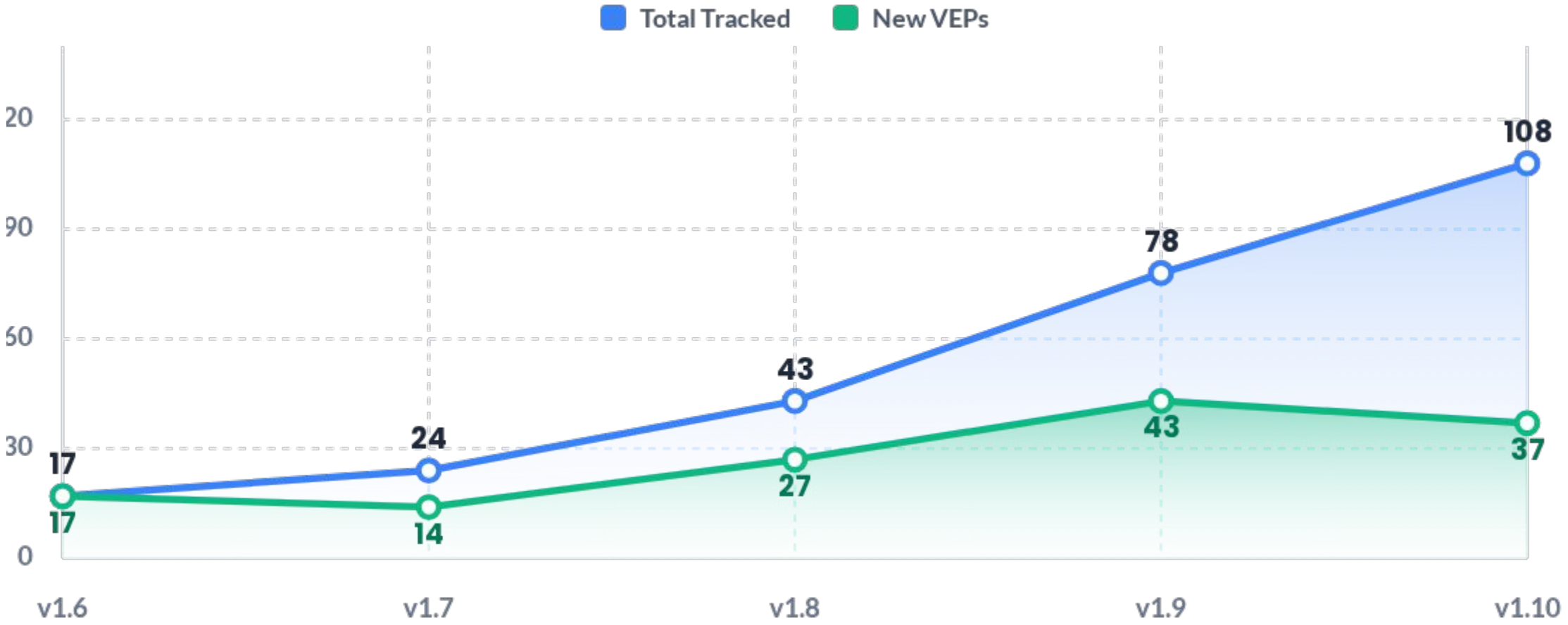
VEPs (Virtualization Enhancement Proposal) mirrors Kubernetes KEPs.

16–week release cycle with enhancement freeze and code freeze.

Development cycle



High Growth in Scope



Massive 6.4x Increase: Tracked VEPs grew exponentially from 17 items in v1.6 to 108 items in v1.10. The community contribution pipeline is expanding rapidly, peaking at 43 new proposals introduced simultaneously during the v1.9 cycle alone.

KubeVirt 1.9 Enhancements Tracking

VEP agent manages the status and likelihood to merge in time
https://github.com/kubevirt-agents/vep-police-agent

Add status update | Insights | Workflows

Enhancements | VEP Planning | Docs | + New view

-status:"Removed from Milestone" -promotion-phase:Remaining

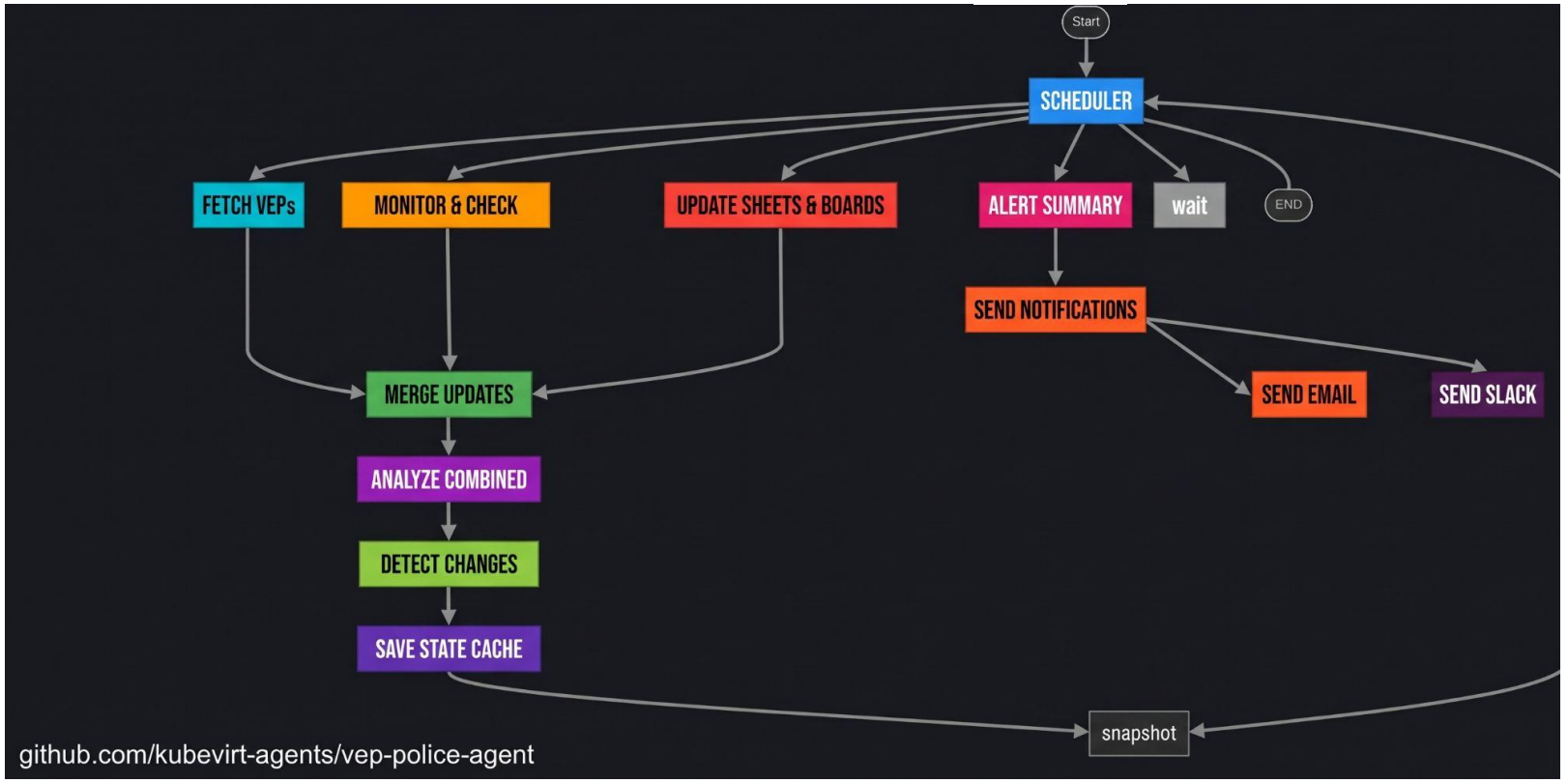
35 | View

	Title	Status	Assignees	Agent ...	Agent Comment	Proposal PRs	Impl PRs
1	VEP 24: Decentralized live migration #24	Exception R...	awels	YELLOW	Moderate risk (50%) - concerned	https://github.c...	https://github.com/kubevirt/kubevirt/pull/1815
2	VEP 14: Track the second phase of the firmware UUID implementation fix #14	At risk	dasionov	RED	At risk (5%) - blocked [STALE]	https://github.c...	https://github.com/kubevirt/kubevirt/pull/1739
3	VEP 192: Decentralized Live Migration LM network proxy support #192	At risk	awels	YELLOW	Moderate risk (50%) - concerned [STALE]	https://github.c...	https://github.com/kubevirt/kubevirt/pull/1792
4	VEP 222: VSOCK network namespace confinement #222	At risk	0xFelix	YELLOW	Moderate risk (60%) - positive	https://github.c...	https://github.com/kubevirt/kubevirt/pull/1795
5	VEP 291: Live Migration - Dynamic Downtime Tuning and Iteration-Based Co... #291	At risk	michalskrivanek	YELLOW	Moderate risk (50%) - concerned	https://github.c...	https://github.com/kubevirt/kubevirt/pull/1748
6	VEP #140: Live update NAD reference #140	Complete	frenzyfriday	GREEN	Likely to merge (100%) - positive	https://github.c...	https://github.com/kubevirt/kubevirt/pull/1704
7	VEP 141: Target-Side Pre-Migration Hooks for Domain XML Modification #141	Complete	Barakmor1	GREEN	Likely to merge (100%) - positive	https://github.c...	https://github.com/kubevirt/kubevirt/pull/1798
8	VEP 143: Monitoring Stack Separation via Hybrid Architecture #143	Complete	machadovilaca	GREEN	Likely to merge (100%) - positive	https://github.c...	https://github.com/kubevirt/kubevirt/pull/1755
9	VEP 156: Expose Memory Overhead on VMI Status #156	Complete	Barakmor1	GREEN	Likely to merge (100%) - positive	https://github.c...	https://github.com/kubevirt/kubevirt/pull/1780
10	VEP 160: Option to opt-out kubevirt RBAC aggregation #160	Complete	orenc1	YELLOW	Moderate risk (50%) - concerned	https://github.c...	https://github.com/kubevirt/kubevirt/pull/1754
11	VEP 168: Masquerade Port Ranges #168	Complete	fra2404, g-nitti, and ...	GREEN	Likely to merge (100%) - positive	https://github.c...	https://github.com/kubevirt/kubevirt/pull/1809
12	VEP 171: Recreate VMI on guest reboot #171	Complete	MarSik	GREEN	Likely to merge (100%) - positive	https://github.c...	https://github.com/kubevirt/kubevirt/pull/1762
13	VEP 172: Cross-Architecture Virtualization #172	Complete	jschintag and lyarwo...	RED	At risk (30%) - concerned	https://github.c...	https://github.com/kubevirt/kubevirt/pull/1719
14	VEP 183: NetworkDevicesWithDRA #183	Complete	oshoval	YELLOW	Moderate risk (50%) - concerned	https://github.c...	https://github.com/kubevirt/kubevirt/pull/1698
15	VEP 62: Generalized Migration Priority in KubeVirt #62	Complete	fossedihelm	GREEN	Likely to merge (100%) - positive	https://github.c...	https://github.com/kubevirt/kubevirt/pull/1781
16	(NO VEP): PanicDevice feature graduation #63	Complete	varunrsekar	GREEN	Likely to merge (100%) - positive	-	https://github.com/kubevirt/kubevirt/pull/1651
17	VEP 76: Native Support For VirtualMachine Templates #76	Complete	0xFelix	GREEN	Likely to merge (100%) - positive	https://github.c...	https://github.com/kubevirt/kubevirt/pull/1791
18	VEP 80: Enable Intel TDX and AMD SEV-SNP #80	Complete	alancaldelas	GREEN	Likely to merge (100%) - positive	https://github.c...	https://github.com/kubevirt/kubevirt/pull/1763
19	VEP 109: Implement vGPU Enabled Live Migration #109	Complete	Aseeef and csomani1	RED	At risk (46%) - concerned [STALE]	https://github.c...	-
20	VEP 115: PCIe NUMA Topology Awareness #115	Complete	mresvanis	YELLOW	Moderate risk (50%) - concerned	https://github.c...	https://github.com/kubevirt/kubevirt/pull/1702

+ You can use Control + Space to add an item

The VEP Police Agent





Talk @ **DevConf.CZ 2026**

Steers maintainer attention to at-risk work before freeze deadlines hit.

Confidential Computing Support

AMD

SEV-SNP (Beta - v1.9)

Intel

TDX - Alpha, v1.9; TDX attestation (VEP-108)

IBM

Z Secure Execution (GA - v1.9)

NVIDIA/Arm

CCA - Planned (VEP388/VEP418)

VEP-340: Inject SNP hostData and TDX mrConfigId into the guest registers for meaningful attestation.

ACCELERATED INFRASTRUCTURE

Supporting Newest NVIDIA Hardware

Grace Hopper & Blackwell today · Vera Rubin next · building on VEP-115 PCIe NUMA topology.

Parallel Enablement

Temporary kubevirt-aie branch let Red Hat + NVIDIA move fast on Grace/Blackwell.

Validated inference model brought back to main.

Inference First

Builds on VEP-115: reconstruct host PCI-NUMA in the guest.

virt-launcher reads sysfs and extends it for Grace (GI nodes, SMMUv3).

Still Unprivileged

iommufd-device-plugin opens /dev/iommu and passes the FD.

virt-launcher stays untrusted. No new caps.

Common case should just work without extra config

VEP-115 gave us the base: mirror host PCIe NUMA topology into the guest. Grace/Blackwell extend that inference model for GI nodes, SMMUv3 isolation, and large PCI holes — still without privileged operations inside virt-launcher.

DYNAMIC RESOURCE ALLOCATION · VEP-183 · VEP-152 · VEP-300

First-Class DRA, Minimizing User Complexity

AI Hardware Driver: Correlate GPU, NIC, CPU on one NUMA node.

Avoid VMI Spec Inflation

VEP-183 · Networking

First-class DRA networks support live migration.

Binding plugins read DRA metadata.
ResourceClaimTemplate-backed claims.
SR-IOV deferred.

VEP-152 · CPU Driver

Controller synthesizes CPU claims from existing cores + dedicatedCpuPlacement.

Accounts for vCPUs, emulator, and IO threads.
Topology via KEP-5304 metadata.
Low-level override only.

VEP-300 · Claim Manager

Provisioner automates multi-device NUMA alignment.

ManagedClaimProvisioner CRD for admin policy.
Deterministic claim names.
Third-party provisioners supported.

WHY WE HIDE THE COMPLEXITY

Users express intent. Claim Manager generates the claim.

WHAT THE USER WRITES

```yaml
apiVersion: kubevirt.io/v1
kind: VirtualMachine
spec:
  template:
    spec:
      domain:
        cpu:
          cores: 8
          dedicatedCpuPlacement: true
        devices:
          gpus:
          - name: gpu1
          interfaces:
          - name: nic1
      networks:
      - name: nic1
      resourceClaims:
      - name: ai-devices
        managedClaimProvisionerName: ai-default
```

WHAT DRA ACTUALLY NEEDS

```yaml
apiVersion: resource.k8s.io/v1
kind: ResourceClaim
spec:
  devices:
    requests:
    - name: gpu
      exactly:
        deviceClassName: gpu.nvidia.com
        count: 1
        selectors:
        - cel:
            expression: |
              device.attributes["gpu.nvidia.com"]
                .productName == "H100" &&
              device.capacity["gpu.nvidia.com"]
                .memory >= quantity("80Gi")
    - name: nic
      exactly:
        deviceClassName: nic.example.com
        count: 1
    - name: cpu
      exactly:
        deviceClassName: cpu.example.com
        capacity:
          requests:
            "dra.cpu/cpu": "10"
    constraints:
    - requests: ["gpu","nic","cpu"]
      matchAttribute: "resource.kubernetes.io/numaNode"
```

Claim Manager turns the left into the right. Users never see DeviceClasses, CEL, capacity math, or topology constraints.

Offloading Complexity: The Three-Way Win

Users: Clean Intent

Express standard VMI device intent only.

No ResourceClaims.
No DeviceClass names.
No accounting math.

+

Vendors: Policy Control

Inject allocation rules via ManagedClaimProvisioner.

DeviceClass mappings.
Topology policies.
StorageClass-like discoverability.

KubeVirt: Absorbs DRA

Synthesizes deterministic claims across NUMA nodes.

Built-in topology aligner.
Finalizers & owner refs.
No status handshake.

Three-Way Win: Vendors set policy · Users keep a clean VMI · KubeVirt handles DRA

Talk @ **KubeVirt Summit 2026**

VEP-190: STRUCTURED PLUGINS

Real features.

Fully out-of-tree. Natively integrated.



Kubernetes runs on contracts

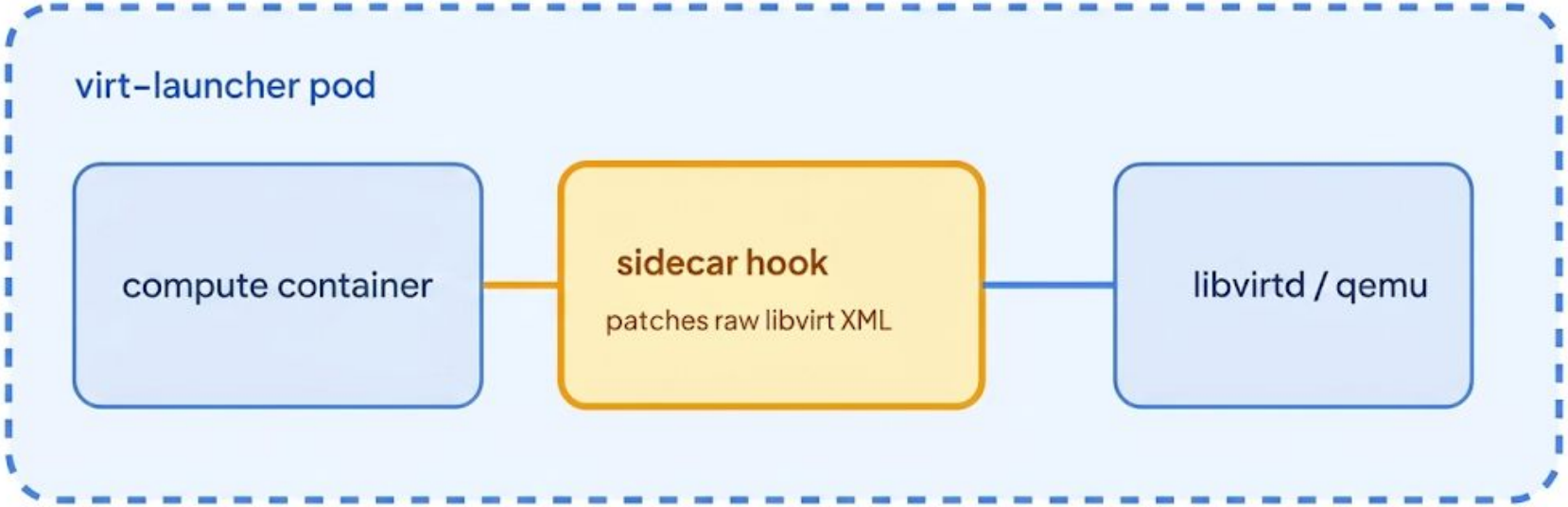
Every major subsystem relies on well-defined plugin boundaries.



KubeVirt deserves the same.

Sidecar hooks: served us well

The XML handoff is the only touchpoint.



Where it breaks

Four critical structural limitations of the legacy model.



Raw XML: fragile



No node access



No admission say



All-or-nothing gate

One Plugin CRD, three touchpoints

Alpha – v1.9 / Alpha 2 – v1.10

Level 1 • CEL

Deploy: nothing

Add a CEL expression to modify guest definitions.

Level 2 • LAUNCHER HOOKS

Deploy: one side-car container

GuestDefinition, PreBoot, PreMigrationSource sidecar hooks.

Level 3 • NODE HOOKS

Deploy: DaemonSet

gRPC calls from virt-handler.

STAGE 1

GuestDefinition

Typed domain spec

NEW v1.10

PreBoot

e.g. Cloud-init ISOs

NEW v1.10

PreMigrationSource

Tune bandwidth & flags

VM START

PreVMStart

PostVMStart

MIGRATION

PreMigrationSource

PreMigrationTarget

PostMigrationTarget

VM STOP

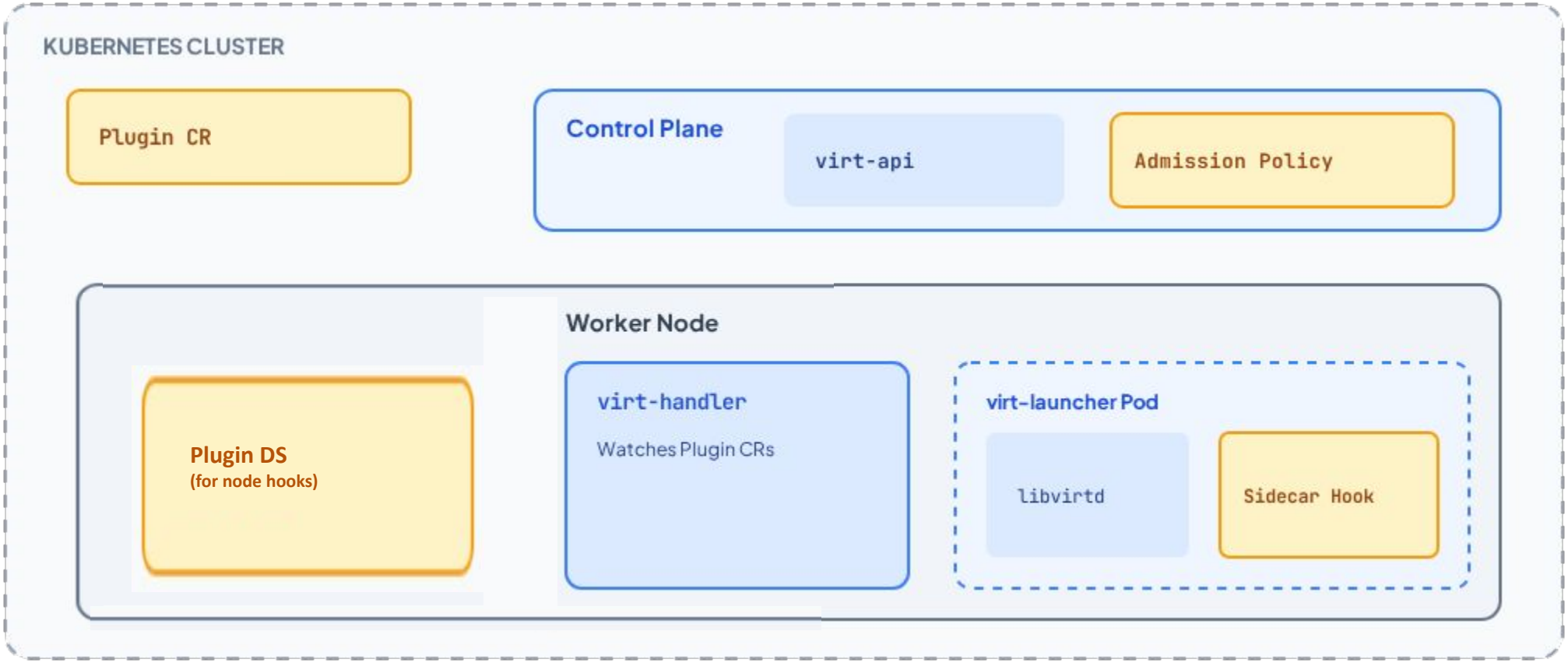
OnVMStop

PostVMStop

One plugin, three touchpoints

Admission, launcher pod, and node-level lifecycle.

Core Plugin K8s



CEL · Admission Policy · DaemonSet + RBAC

kubevirt/plugins

Introducing the KubeVirt Plugin SDK.

The whole plugin

Fluent Go builder pattern with automatic dispatch.

same binary: serve + generate

```
type handler struct{}
func (h *handler) MutateDomain(ctx context.Context, d *libvirtxml.Domain, vmi *v1.VirtualMachineInstance) error {
      d.Sysinfo.System[0].Value = "KubeVirt Summit"
      return nil
}

p := plugin.New("smbios-injector").WithDomainHook(plugin.ForLibvirt(&handler{}).WithFailureStrategy(plugin.Fail))
p.Execute()
```

generate

CR · admission · DaemonSet · RBAC · image

```
$ plugin generate
```

Emits ready-to-apply manifests into deploy/ directory with distroless multi-stage Dockerfile.

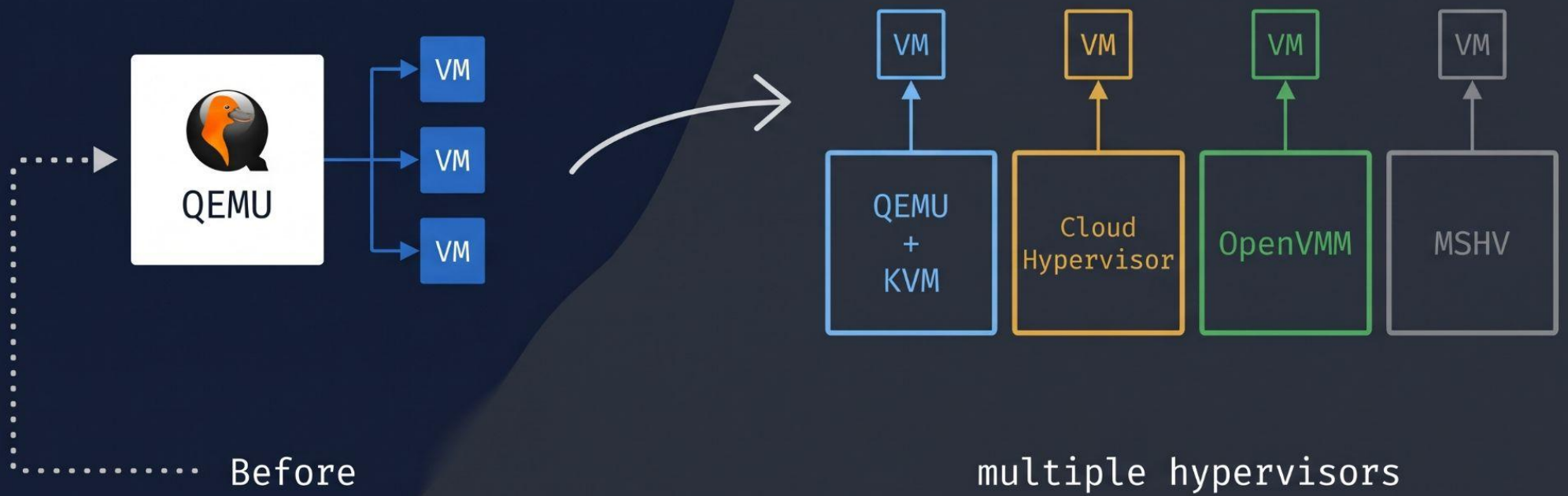
- 01-plugin.yaml (Plugin CR)
- 02-mutating-admission-policy.yaml
- 03-policy-binding.yaml
- 04-daemonset.yaml + 05-rbac.yaml
- Containerfile + Makefile

Market-place!



Beyond QEMU: multiple hypervisors

Virtual machine management (KubeVirt)



Step 1: Hypervisor Abstraction Layer

Alpha - v1.8

VEP-97: KVM out of the core

Before: KVM assumptions hard-coded across virt-controller, virt-handler and virt-launcher.

Now: per-hypervisor contracts. Microsoft's MSHV runs in-tree: **/dev/mshv** instead of **/dev/kvm**, same libvirt + QEMU.

```
kind: KubeVirt
spec:
  configuration:
    hypervisors:
    - name: hyperv-direct   # feature gate: ConfigurableHypervisor
```

One hypervisor per cluster, still QEMU. Next: make the whole stack pluggable.

Step 2: pluggable virtualization stacks

In progress

VEP-82 (kubevirt/enhancements PR #311): libvirt becomes an implementation choice, not part of the contract.



Goal: VMs on different stacks, side by side on the same node.

Two kinds of pluggability

Extend what KubeVirt does to a VM. Swap what runs it.

KubeVirt VM API (v1) — Unchanged

Network Binding Plugins — fold into VEP-190? — GA

Structured plugins VEP-190 — CEL | Launcher hooks | Node hooks — EXTEND — Alpha - v1.9 / Alpha 2 - v1.10

Hypervisor VEP-97 — KVM | MSHV — SWAP THE HYPERVISOR — Alpha - v1.8

Virtualization stack VEP-82 — libvirt + QEMU (default) | Cloud Hypervisor | OpenVMM — SWAP THE VMM — In progress

One API. Three layers you can extend or swap.

Teaser to the future: Agent-Sandbox integration

Use KubeVirt as the isolation backend

- Highly customizable
- Supports DRA and complex hardware configurations
- Auto-scalers, de-schedulers and other Kubernetes facilities
- Live-migration
- Confidential computing
- Much better than the current alternatives :)

Get involved

Ten years of virtualization on Kubernetes.

Still a lot of work ahead — join us.



KubeVirt Community Calendar



github.com/kubevirt/kubevirt



github.com/kubevirt/enhancements



github.com/kubevirt/plugins

The First Hard Problem: Networking

One binding, then multiple bindings, then pluggable network binding plugins built outside KubeVirt.



[TODO Vladik: final wording of the bindings story]

TEE Support in KubeVirt

AMD SEV-SNP	**Beta**	launchSecurity.snp	Since KV 1.7. Mutually exclusive with SEV.
Intel TDX	**Alpha**	WorkloadEncryptionTDX	Launch in 1.7; QGS support in 1.9.
IBM Secure Execution	**GA**	On by default	GA in KV 1.9. Offline migrate only.
ARM CCA	**Planned**	VEP-338 / 418	Under discussion.

ADVANCING: InitData (VEP-340 / PR #346)

Inject SNP hostData and TDX mrConfigId for meaningful attestation.

virt-controller watches LaunchData (owned by external operator), gates virt-launcher pod creation until ready (DataVolume pattern). Immutable initDataRef on VMI. Prefer pod env var for resolved digest. No extra Committed condition. Feature gate: InjectInitData.

Maturity through adoption

OSAC

KubeVirt as the base for VM-as-a-Service, and in future Cluster-as-a-Service.

Standalone VM-pods (edge)

Rendered from a VM object and run with plain Podman outside Kubernetes.

AI hardware: the newest NVIDIA platforms

Grace Hopper & Blackwell

Alpha – v1.9

GPU passthrough: GH200, GB200, GB300 – Alpha, v1.9 (VEP-199).

Vera Rubin

Planned