

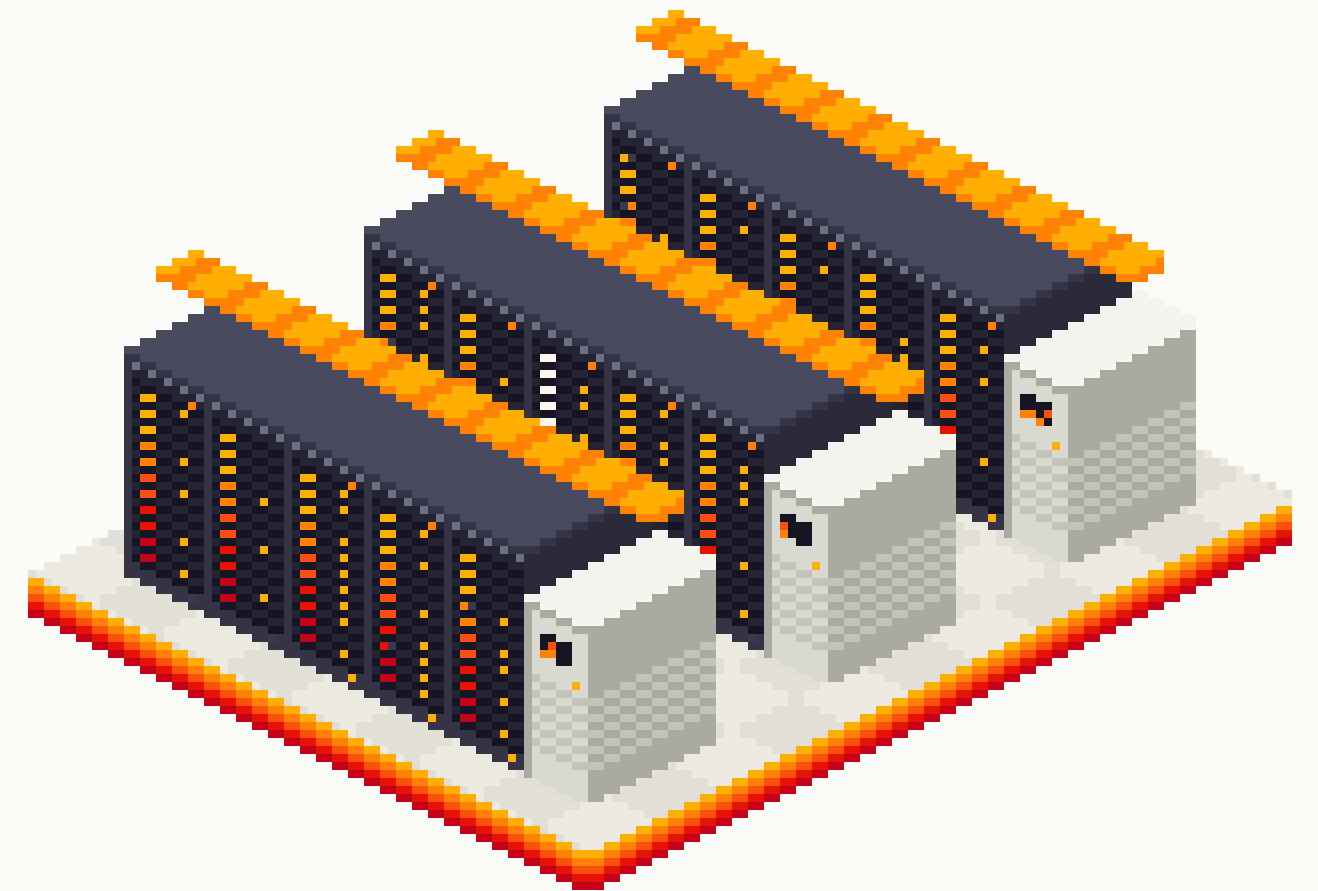


From zero to GPU cloud

12 months building Mistral Compute on top of
open source

Antoine Roy · Léonard Suslian · Mistral Compute

Wed 11 Nov 2026 · Salt Palace · Level 2 · 255 EF



5,000 bare-metal hosts under management
GB200 · GB300 · B300 · CPU

We build and operate Mistral Compute



Antoine Roy

linkedin.com/in/ant-roy

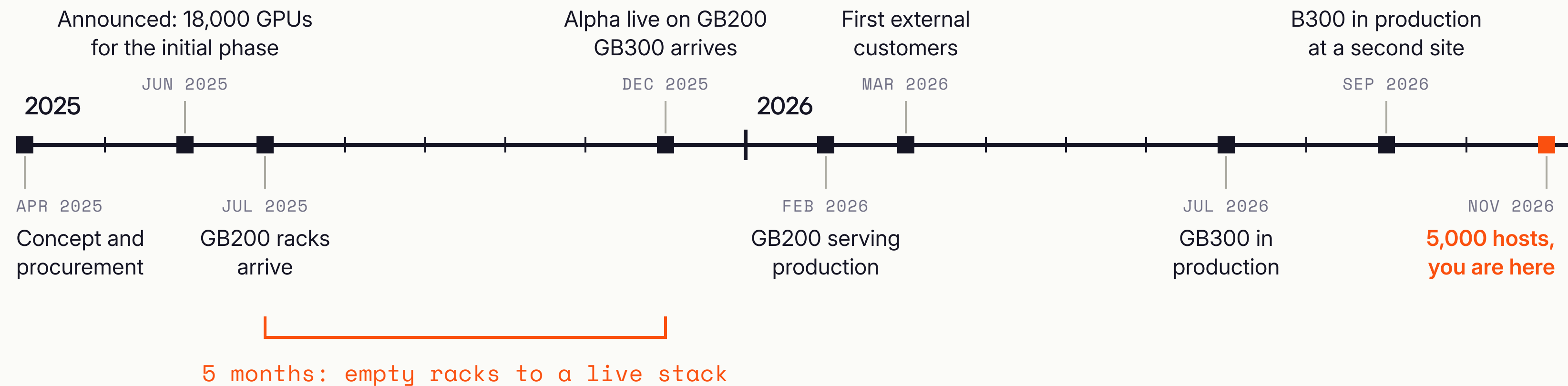


Léonard Suslian

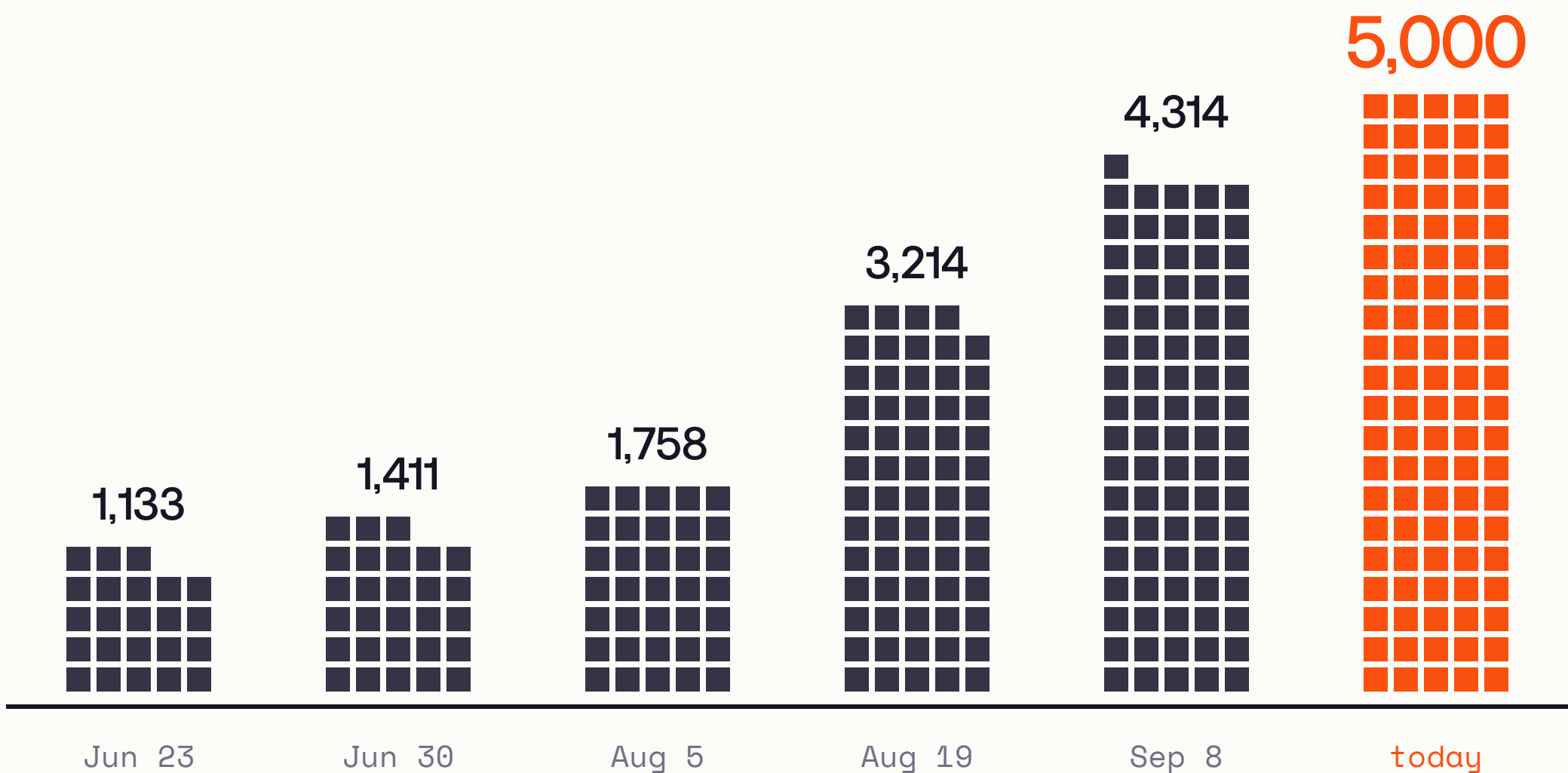
[@synthe102](https://twitter.com/synthe102)

Our job: make GPU infrastructure usable through **a consistent cloud interface.**

July 2025: empty racks. Today: a GPU cloud.



1,133 to 5,000 hosts in under five months



bare-metal hosts under management, 2026 · 1 square = 50 hosts

MIX · SEPTEMBER 2026

90% GPU hosts

GB300 NVL72

Grace · arm64

GB200 NVL72

Grace · arm64

B300 HGX

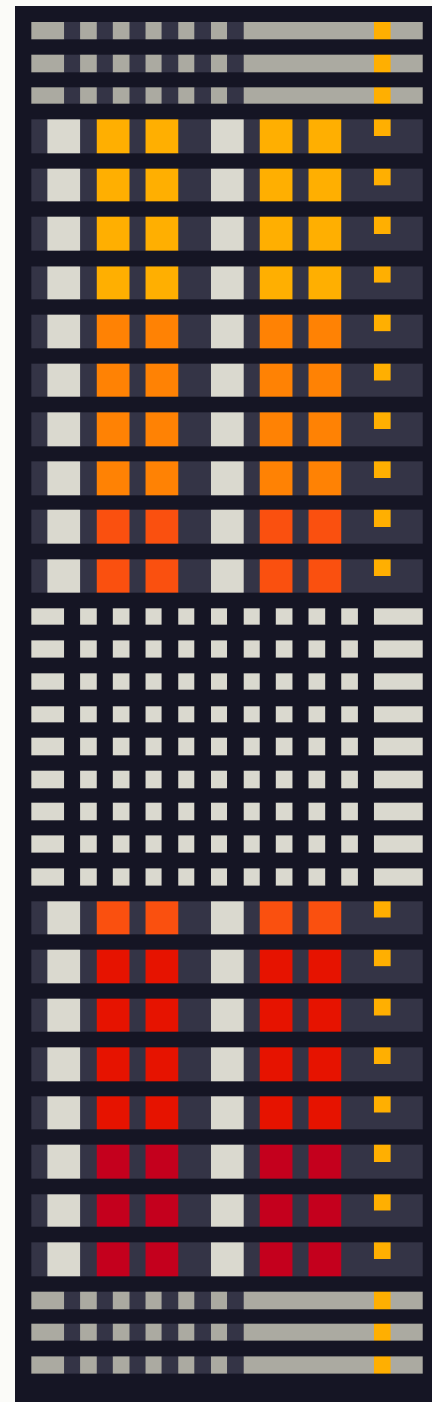
x86

10% CPU hosts

CPU nodes

x86 and Grace

One NVL72 rack is 18 Kubernetes nodes



■ 18 compute trays

each one is a Linux host: 2 Grace CPUs + 4 GPUs

= 18 Kubernetes nodes

□ 9 NVLink switch trays

all 72 GPUs share one NVLink domain, across 18 hosts

■ Power and liquid cooling

shared by the whole rack

72

GPUS

36

GRACE CPUS

18

HOSTS

1

NVLINK DOMAIN

B300 HGX, for contrast: 8 GPUs inside one x86 host.

Built in Europe, growing to 200 MW

200 MW

of compute capacity across Europe by the end of 2027



operating announced planned

PARIS REGION

Operating

 44 MW expansion under way

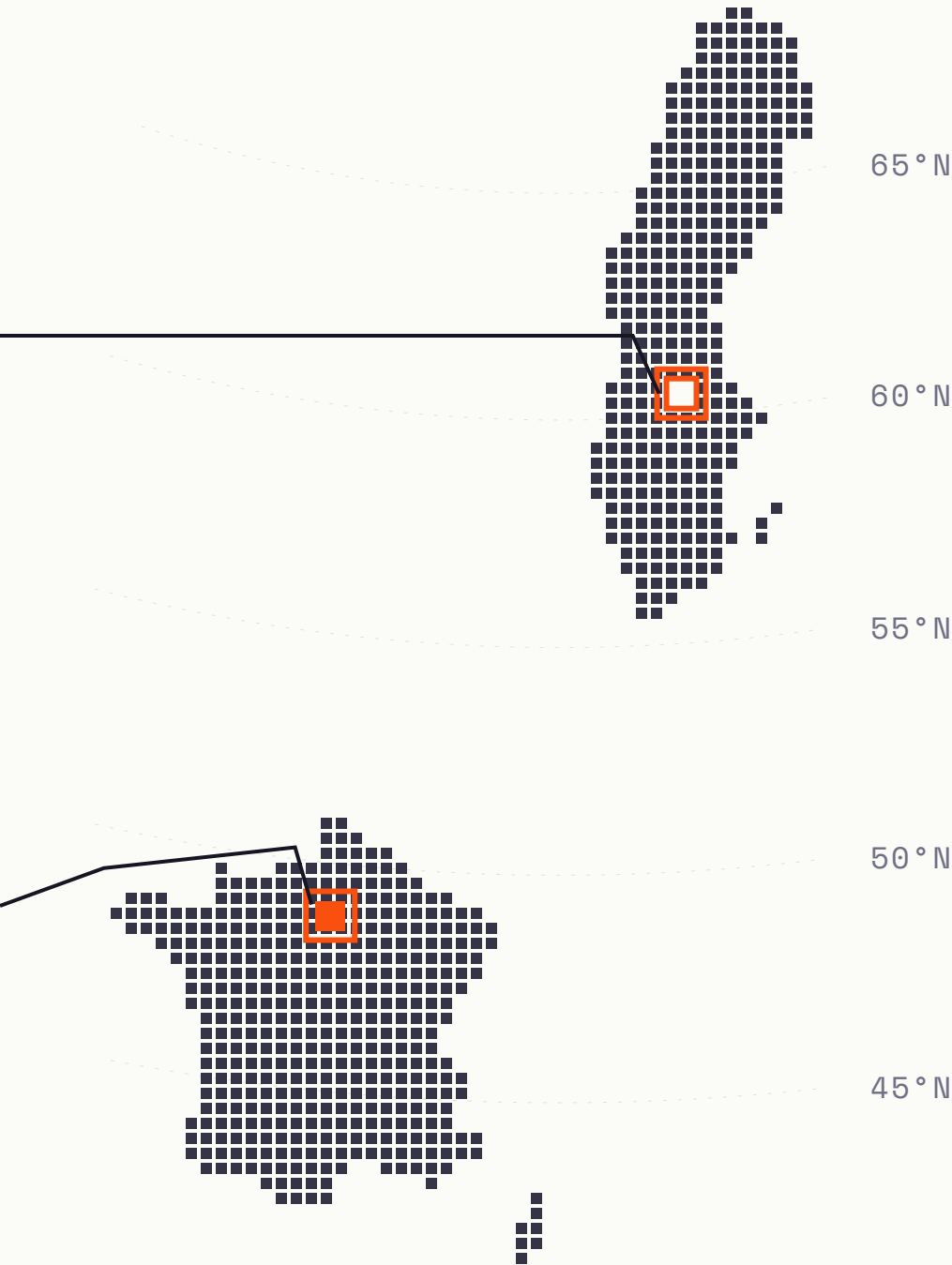
Announced

 an inference site

SWEDEN

Planned

 for 2027



Standing on the shoulders of open source

LIFECYCLE



Cluster API



Metal3



Kamaji



etcd

WORKLOADS



Slinky



Slurm



NVSentinel

DELIVERY



Talos



Argo CD



Sveltos

OBSERVABILITY



Prometheus



Grafana

NETWORK



Cilium



NetBox



Kea DHCP

OURS, IN BETWEEN

Compute API
product API

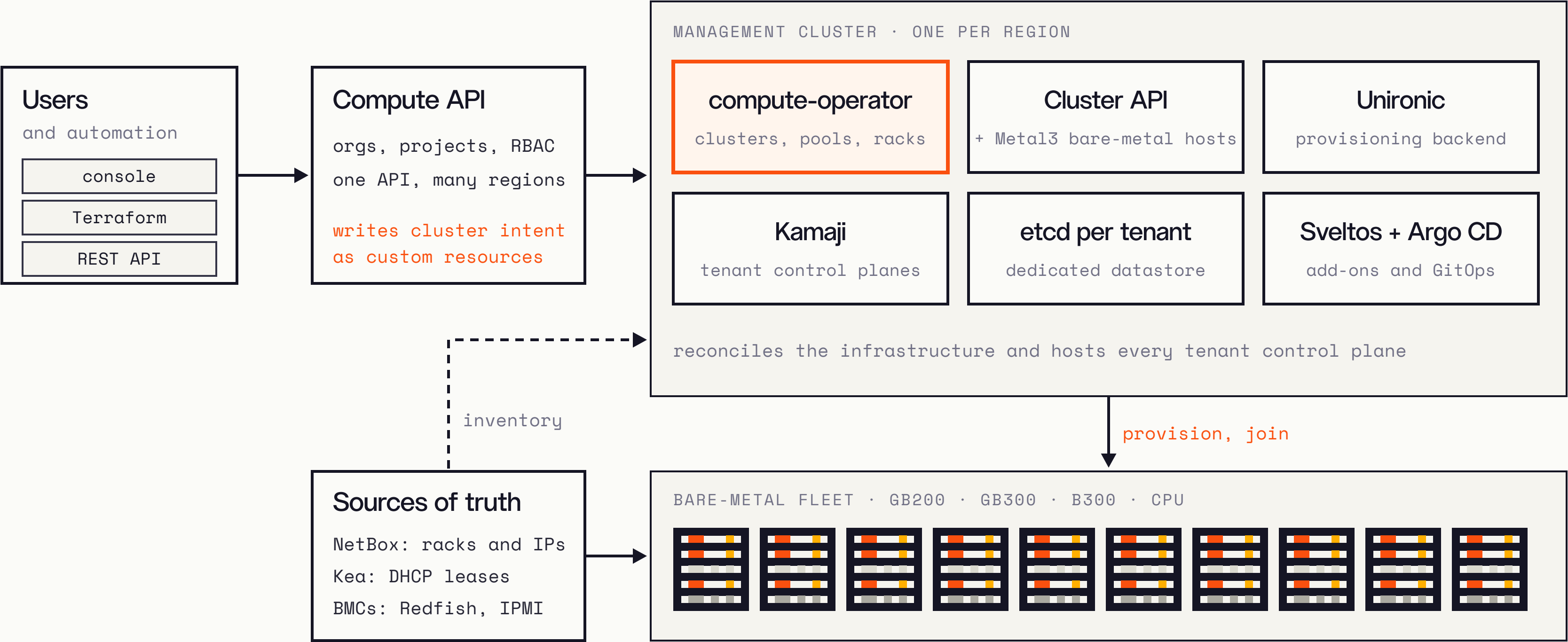
compute-operator
clusters, racks



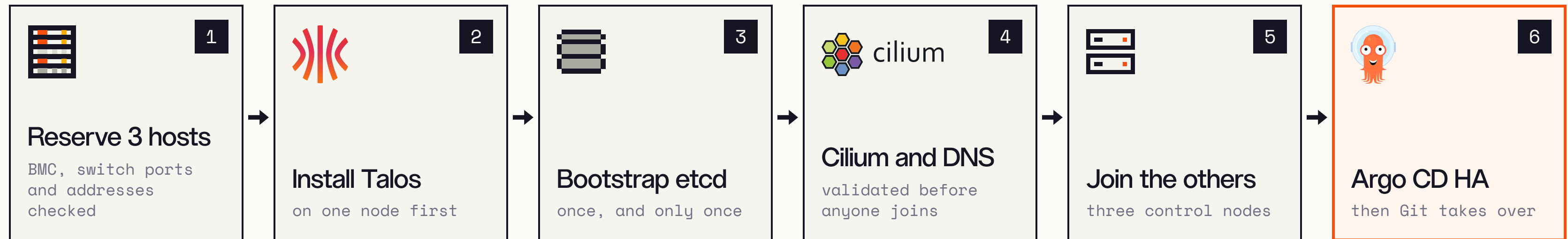
Unironic

HPC operator
Slurm health

Mistral Compute at a glance



One Talos node to a management cluster

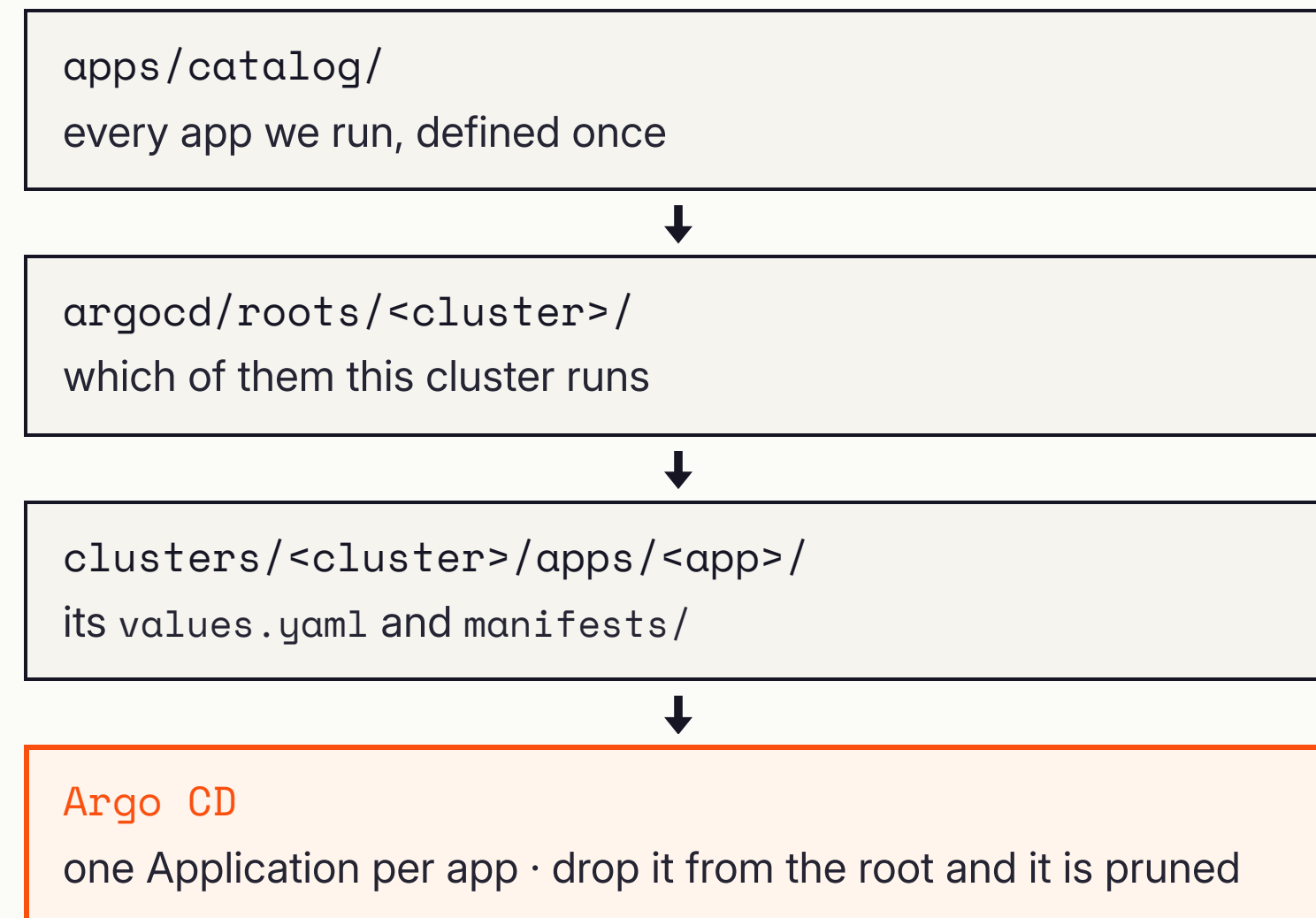


 **Credentials stay outside the checkout.** A private repository is not a credential store.

 **No kind cluster, no clusterctl move.** Our pivot left BareMetalHost network data behind, so we dropped it.

Management clusters run Talos. Tenant nodes run Ubuntu 24.04 and kubeadm, from images we build for x86 and arm64.

After day 0, the management cluster is Git

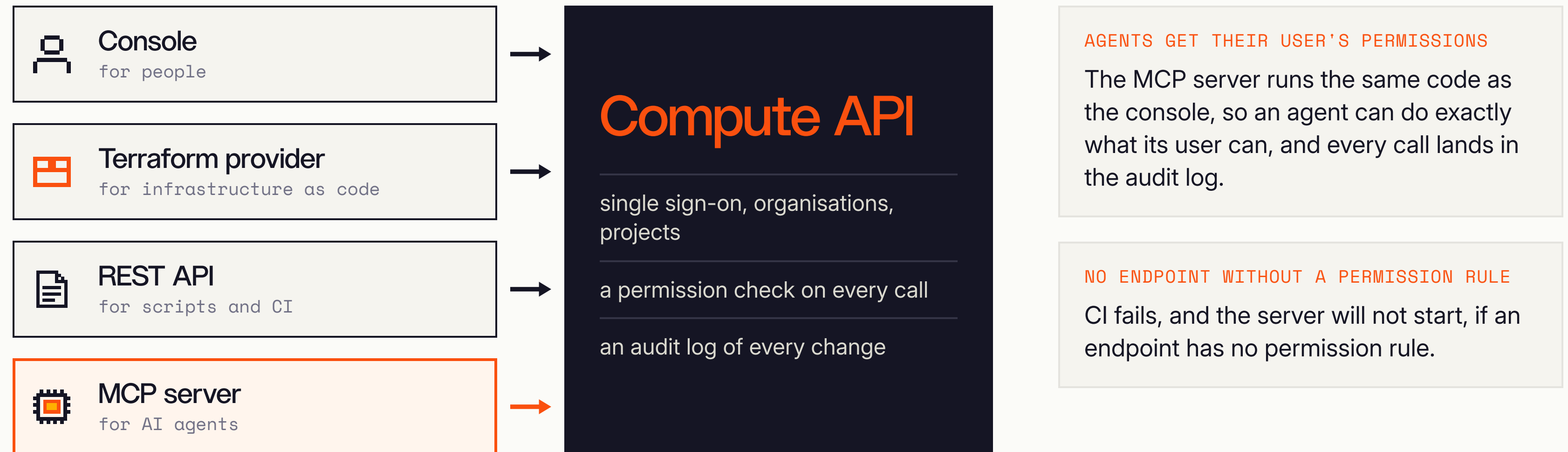


```
$ git log --oneline
88219a8 unironic: bump to v0.2.94
821862e kea: add a 3-node Galera database
0bd961e alerts: NVMe over TCP link down
ab21742 compute: deploy v0.0.245
3e989c2 nvsentinel: bump our janitor provider
d44a4a9 fabrik: scale up the switch provider
```

Every change to a management cluster is a reviewed pull request, and Argo CD applies it.

Management clusters are Git. Tenant clusters are labels.

The console, Terraform and AI agents all use the same API



An AI agent never gets more access than the person it acts for.

From an API call to a running cluster

POST /v1/projects/{project}/clusters

```
MANAGEDCLUSTER · SPEC EXCERPT
kind: ManagedCluster
spec:
  kamaji:
    replicas: 3           # hosted control plane
    datastore:
      storageType: etcd
      etcdReplicas: 3    # dedicated etcd
```

+ ManagedNodePool per worker pool
replicas · rack selector · OS image · Kubernetes version

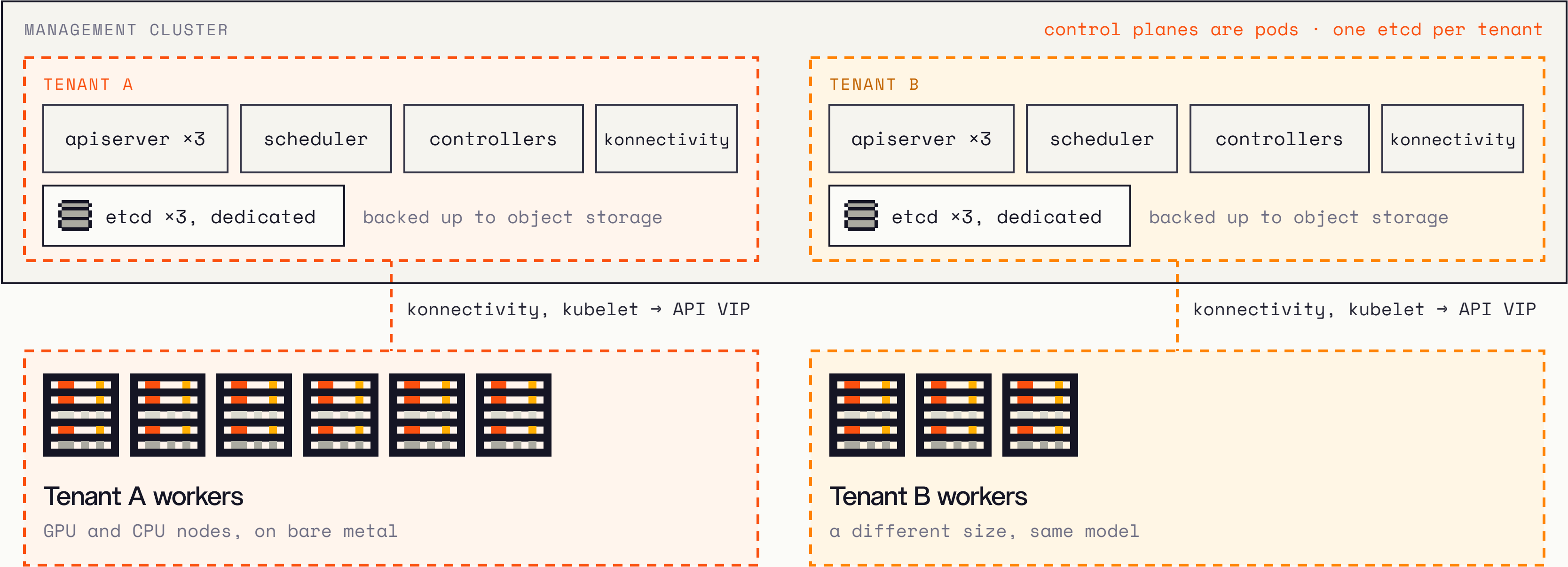


COMPUTE-OPERATOR RECONCILES INTO

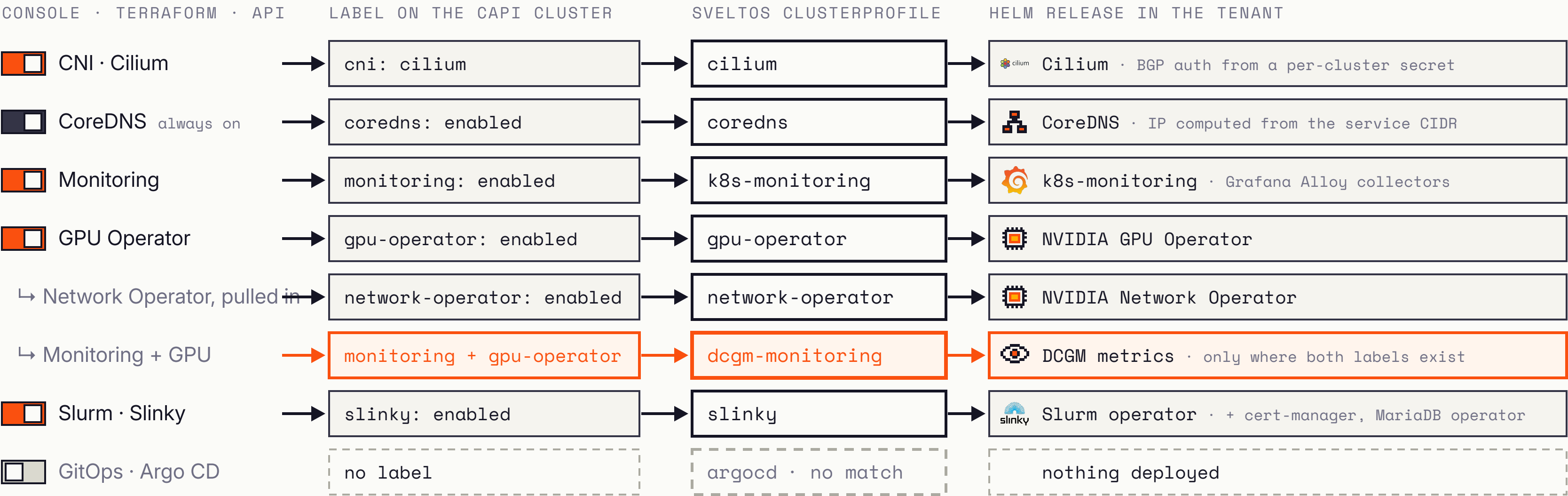
-  **Cluster API Cluster**
+ Kamaji control plane
-  **etcd cluster**
3 members, backed up
-  **Machine templates**
Metal3, one set per pool
-  **Plugin labels**
on the Cluster, for Sveltos
-  **Machines**
claim bare-metal hosts

Users see a cluster and node pools. Never a Machine.

Hosted control planes, physical workers



Plugins: a label in, a Helm release out



TEMPLATED PER CLUSTER

Values read the Cluster: Cilium's BGP auth from its own secret, CoreDNS's IP from its service CIDR.

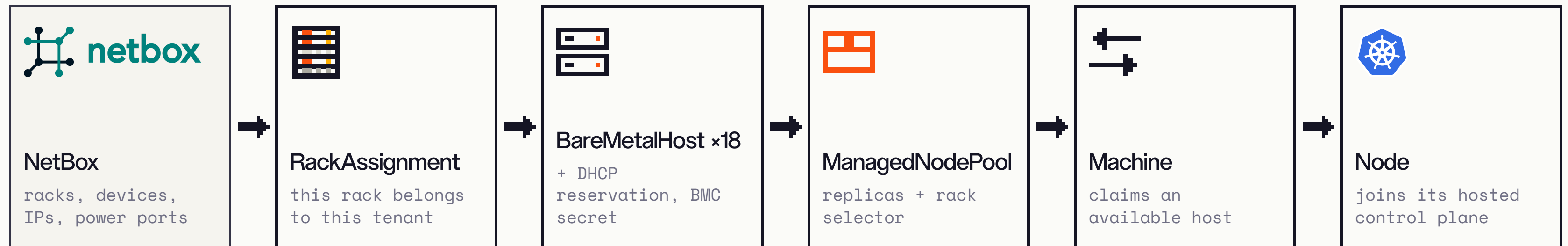
DRIFT DETECTION

Edit an add-on inside a tenant and Sveltos puts it back. Untick a plugin and most releases are withdrawn.

EVERY PROFILE IN GIT

Reviewed like any other change. Admin-only plugins never reach the public API.

A rack is a Kubernetes object



DeviceAssignment

overrides the rack for one host: moving capacity is a one-line change

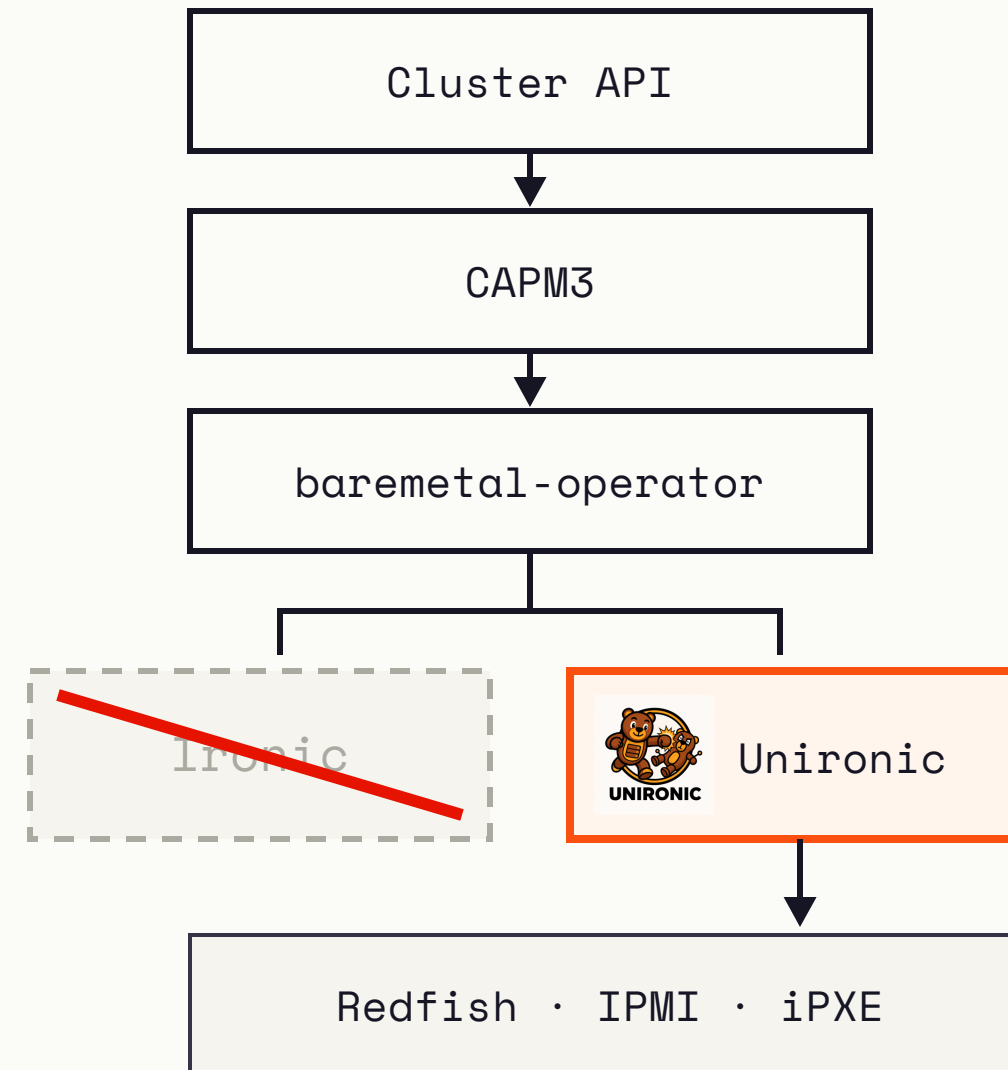
Topology labels

parsed from NetBox names: dc · hall · pod · rack · tray, so schedulers can see racks

We kept Metal3. We replaced IroniC.

WHY

- 🕒 Hard to debug, and our deployment hit its scaling limits
- 🔧 Delicate GPU hosts need exact provision and clean steps
- 🐟 Every BMC model behaves a little differently



350k

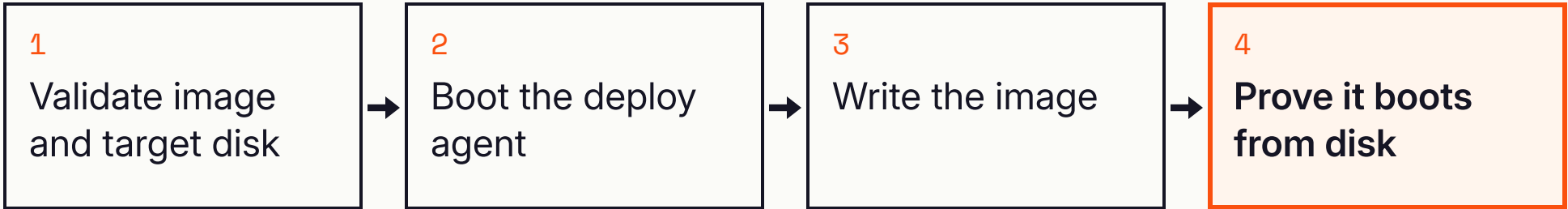
events handled per day

every boot, power action and check, on our largest site

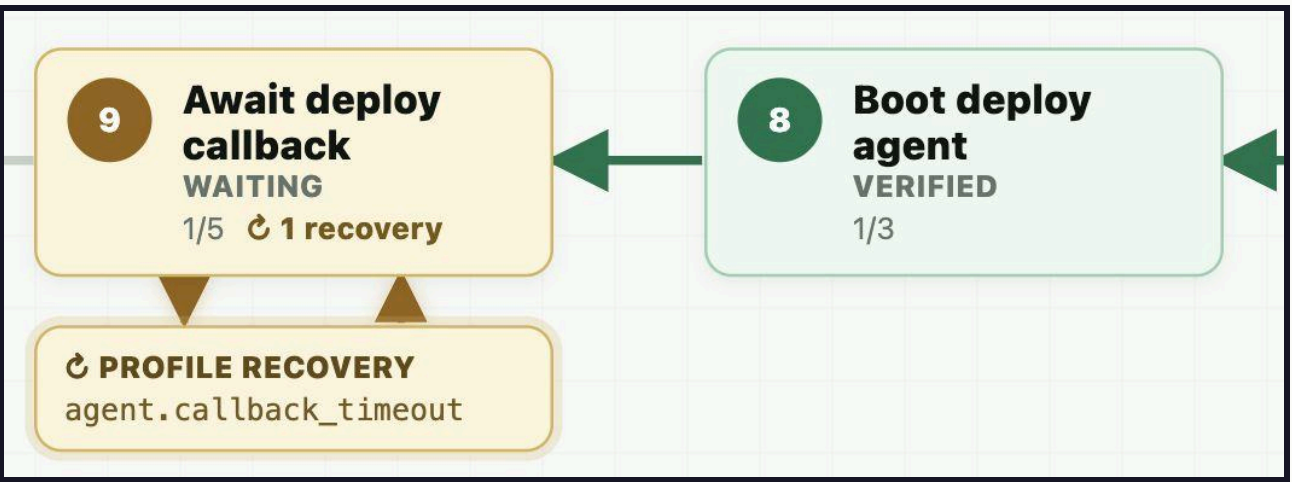
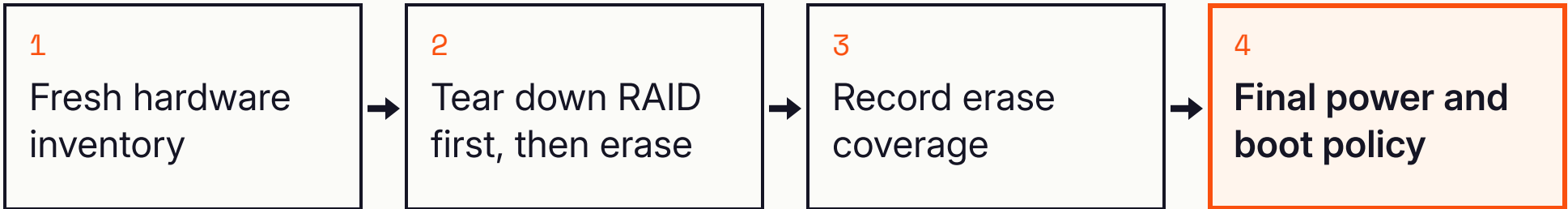
Same BareMetalHost API, same IPA ramdisk. Go, PostgreSQL, durable workflows. From the first commit to production in 19 days.

Every provisioning step is explicit

PROVISION



CLEAN AND REUSE



Unironic UI · each step, attempt and recovery branch · synthetic data

SAFETY GATES, AS CONFIG IN GIT

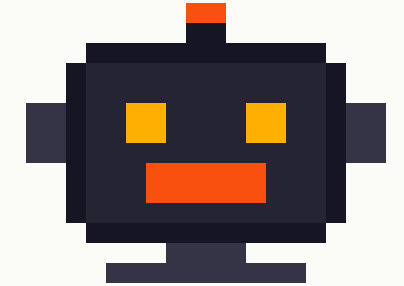
300 power actions at once across the fleet, at most 18 per rack

64 disk wipes at once, at most 4 per rack

5 failures of one kind in 5 minutes trip a circuit breaker

 One boot profile per hardware model: exact-MAC PXE, one-shot boot, cold cycles

203 production hosts held back by one audit: a host in a running cluster is never rebooted by automation



Hosts our automation can't fix go to an AI agent

WHAT THE AGENT IS ALLOWED TO DO

-  It runs in its own pod, one per host, with no access to the node it runs on
-  It gets temporary credentials for that one host, and nothing else
-  An admission policy lets it change a single field on that host
-  Inside our network, it can only reach the host's BMC and the few APIs it needs
-  Kubernetes stops it after 30 minutes
-  **It cannot declare the host fixed: the host has to pass our normal checks**

IN PRODUCTION SINCE SEPTEMBER

184 agent runs in one week,
on our largest site

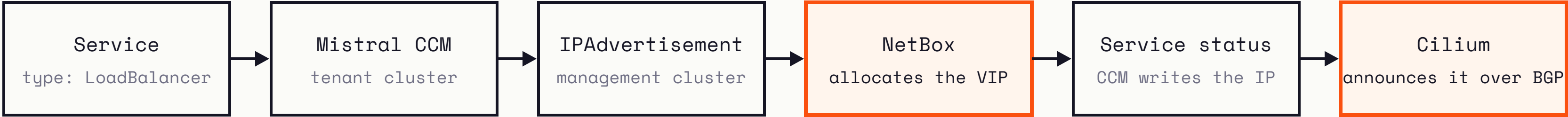
It reads the BMC and the console logs, can take a screenshot of the remote screen, and fixes what it finds, like a broken boot entry.

It started when a coding agent repaired a broken boot loader through a BMC console. We shipped the feature the next day.

 A host only goes back into service after passing the same checks as any other.

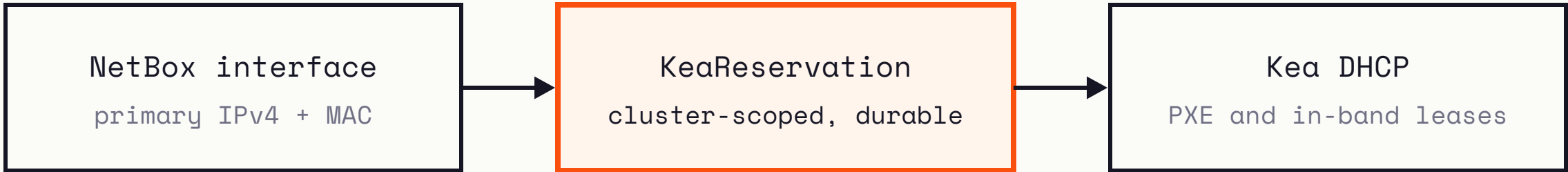
How a Service gets a public IP on bare metal

A LOADBALANCER SERVICE



an allocated IP is not a reachable IP: the routers still need the BGP route

EVERY HOST'S DHCP IDENTITY

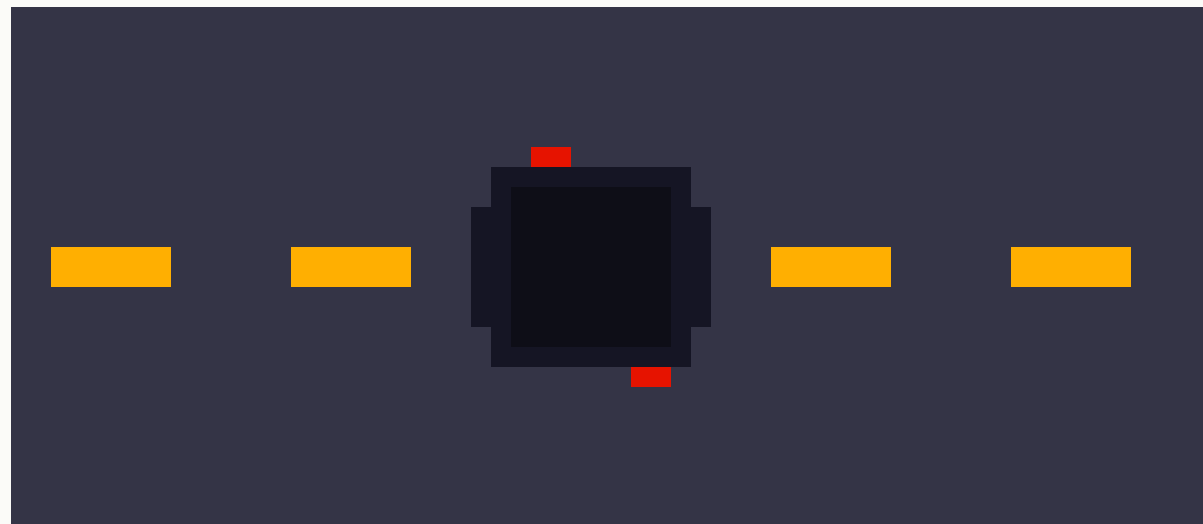


moving a host between racks or tenants keeps its address

GOTCHA

externalTrafficPolicy:
Cluster SNATs the client IP. A
customer's allowlist rejected its
own traffic.

The potholes

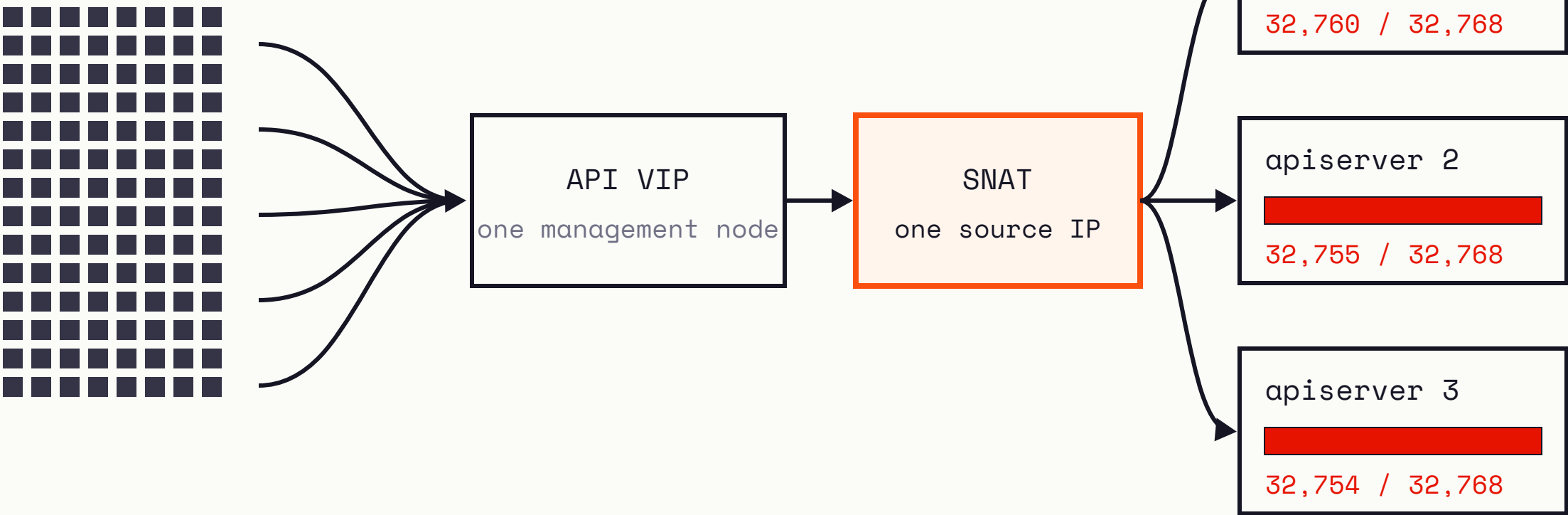


- 01 A control plane past 4,000 hosts
- 02 The network you did not know you had
- 03 arm64 on Metal3
- 04 BMCs, Redfish and NVL72 racks
- 05 Fleet monitoring, and our own bugs

Our first scaling problem was SNAT, not etcd

kubelets

every watch is a connection



98,269

NAT mappings, 99.96% of the table

433/s

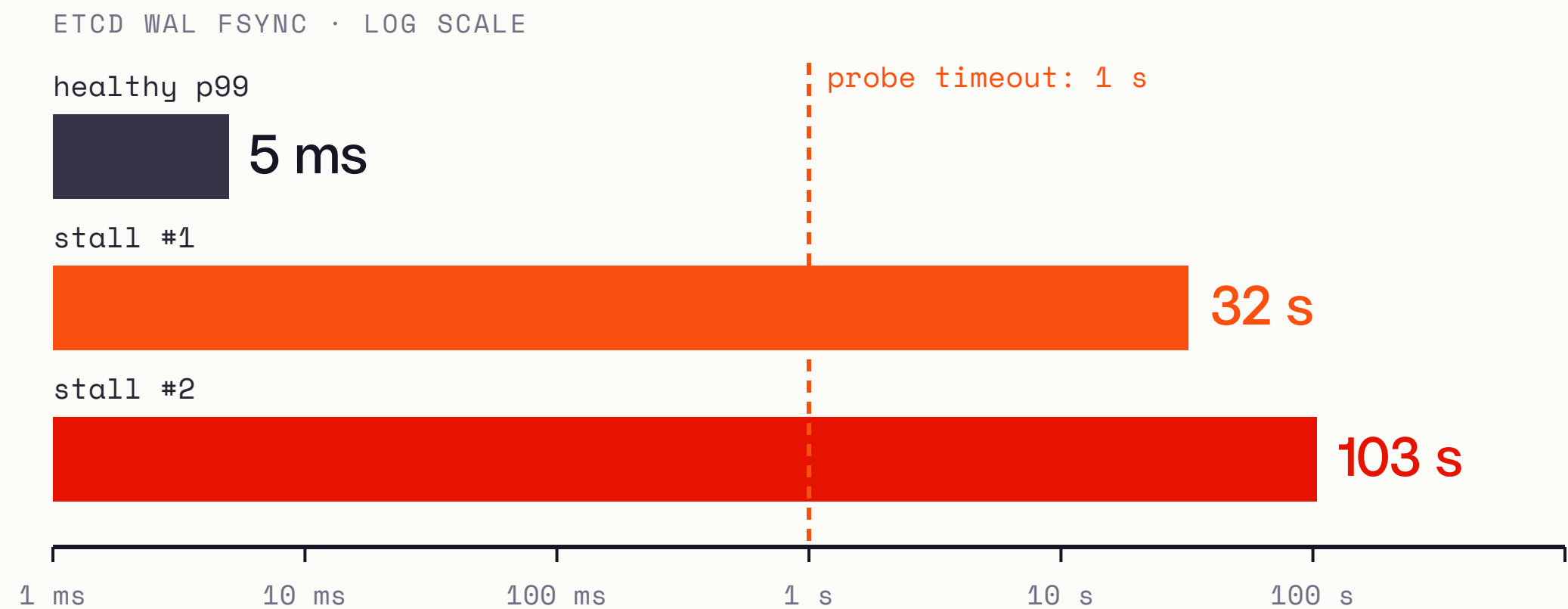
drops: "no mapping for NAT masquerade"

0

etcd leader changes · fsync p99 5 ms · API 5xx

 With hosted control planes, every kubelet goes through a few source IPs: alert on the NAT table.

One slow disk restarted two control planes

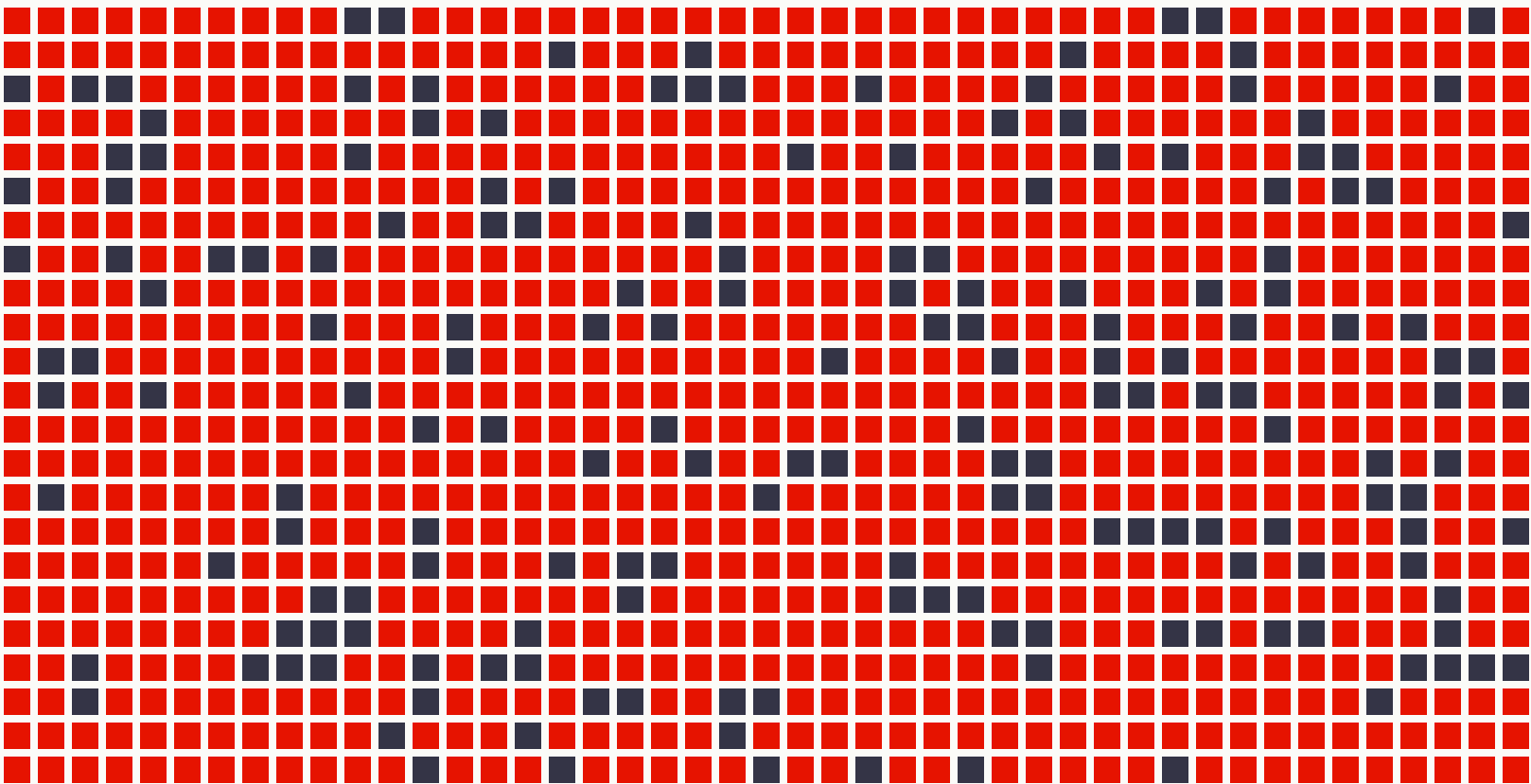


Two tenants on the same storage, at the same second. Each API server restarted 4 times in 4 minutes.

WHAT WE LEARNED

-  etcd gets isolated, low-latency disks
-  Probes must survive an fsync stall. Ours timed out after 1 s.
-  Requests on every control-plane pod. Ours were BestEffort.
-  An explicit etcd StorageClass. Ours silently fell back to the default.

A DHCP problem that looked like a control-plane outage



857 / 1,035 GB300 nodes Unknown

-  10-minute DHCP leases, renewed at 5
-  Kea flooded: 5 M+ DISCOVERs from BMC and DPU ports with no pool
-  Nodes lose their IP and their default route
-  Lease PUTs time out, an API server at 20 CPU / 32 GiB: it looks like the control plane
-  Recovered kubelets register a Cilium IP: no --node-ip

 Long leases or static in-band IPs. A DHCP server sized for every port. Always pin `--node-ip`.

What broke when we booted arm64 Grace hosts

STAGE	WHAT X86 ASSUMED	WHAT GRACE NEEDED
NETWORK BOOT	the x86 iPXE binary	snponly-arm64.efi, over HTTP
DEPLOY AGENT	an x86 IPA ramdisk	an aarch64 IPA ramdisk
FIRST DISK BOOT	firmware falls back to the disk	the disk sat behind ~14 PXE entries: the agent writes the UEFI entry
TOOLING	efibootmgr --full-dev-path	efibootmgr 16 rejects it: caught by an arm64 VM in CI

83% of our Metal3 hosts are arm64

same CAPI and Metal3 objects for arm64 and x86

all UEFI

A BMC update made GB300s boot from the network forever

```
$ PATCH Systems/System_0 Boot.BootOrder=[Hdd, ...]  
403 Forbidden # even as Administrator  
$ PATCH Systems/System_0/Settings Boot.BootOrder=[Hdd, ...]  
204 No Content # accepted, never applied  
$ PATCH Systems/System_0 BootSourceOverride=Continuous/Hdd  
204 No Content # wiped at the next boot  
$ PATCH Systems/System_0 BootSourceOverride=Once/Hdd  
204 No Content # works. Exactly once.
```

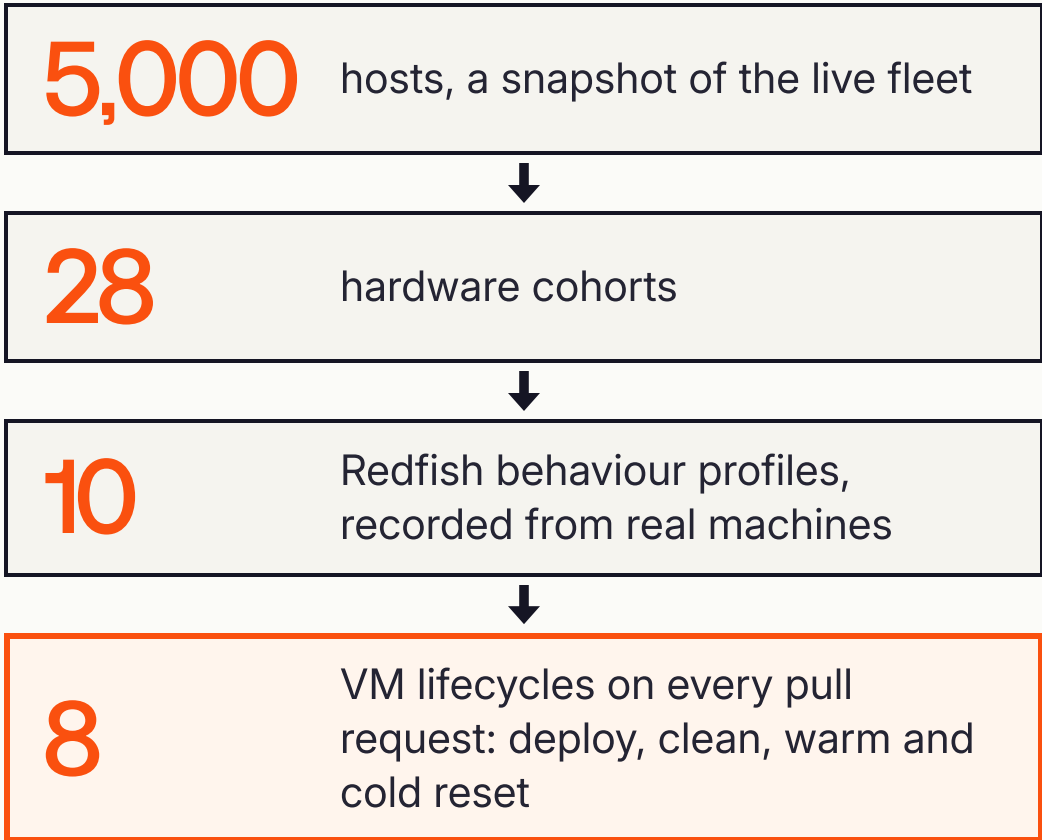
WHAT UNIRONIC DOES NOW

- 1 Re-arm a one-shot disk boot before every boot
- 2 Cold power cycle: a soft reset ignores the override
- 3 Trust the agent API and kubelet Ready, not BootProgress: OSRunning



Same Redfish spec, different behavior: we keep a boot profile per hardware model.

Testing BMC behavior in CI, without the hardware



RECORDED QUIRKS

GB NVL tray

Pending boot order: accepted, never applied

x86 GPU server

BootNext reads back fine, then is ignored at boot

x86 CPU server

No BootNext at all

REAL BMC FIRMWARE, IN QEMU

- Vendor BMC images boot and serve their own Redfish API
- Signed boot: the original image boots, a tampered one is rejected
- We patched QEMU so the BMC's crypto engine computes real RSA
- Images and charts publish only if this passes

Each BMC quirk we hit in production became a CI test.

When the BMC hangs, the PDU decides

✓ Web UI	200 OK
✗ Redfish	500
✗ IPMI	no answer

NO REMOTE RESET PATH LEFT

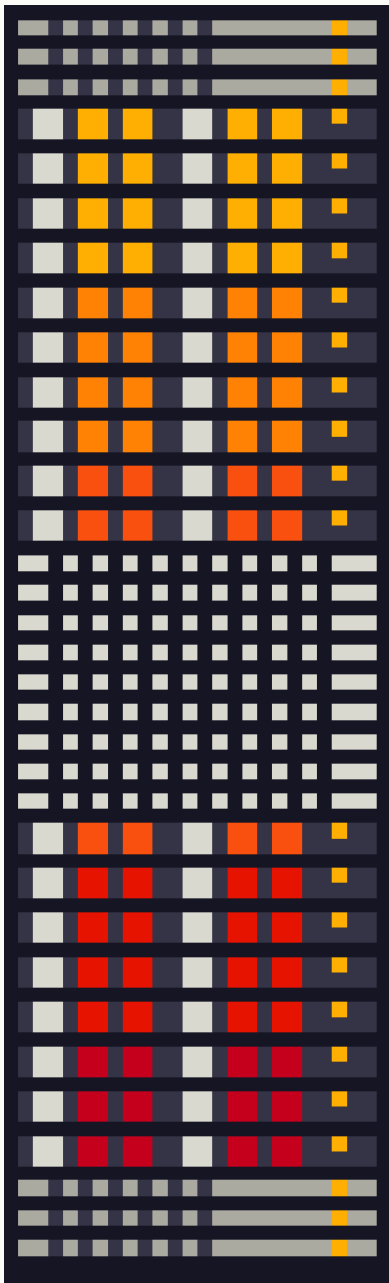
- ✚ Find the host's outlets in NetBox, by PDU name: PDU IPs collided with OOB switches
- ⚡ Cut **both** feeds at once, wait 60 s
- ✓ The BMC comes back

MEANWHILE, IN OUR NIGHTLY BMC
RESET JOB

72 / 72 FAIL

The BMCs had reset fine. They dropped their
network before sending the IPMI ACK.

On NVL72, the rack is the unit



Capacity

One RackAssignment = 18 hosts. Pools scale in racks: 144 = 8 × 18.



Power

Power and destructive actions are budgeted per rack, not only per fleet.



Telemetry

NVSwitch, CDU and PDU metrics are per rack, next to the node exporters.



Break-fix

Done means 4 GPUs, all 72 NVLinks up, a healthy fabric.

We monitor water, not just GPUs

NODES	DCGM, NVSentinel, kernel panics, kubelet
FABRIC	NVLink and InfiniBand managers, switch config drift
BMCS	firmware inventory over Redfish, every serial console
POWER	power shelves, rack PDUs, every feed
COOLANT	CDUs: flow, pressure, leaks, dew point

144 alert rules in Git, all tested in CI

21 of them watch the coolant: leaks, dew point, pumps



Exporters we wrote for hardware that shipped without one



An SLA alert per rack, from Cluster API and Metal3 state

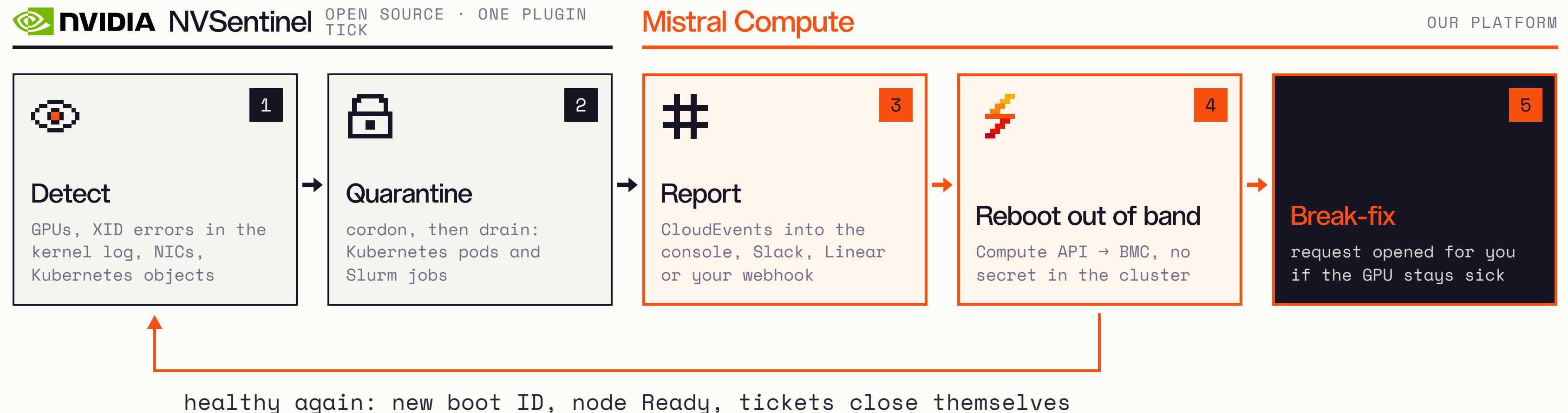
CDUWaterLeak

CDUBelowDewPoint

CDUPodPumpRedundancy

RackBreachedSLA

NVSentinel finds GPU failures, our platform handles the rest



From a failing GPU to a break-fix request, with no human in the loop.

Monitor what is missing, not just what is red

desired: 0

60 arm64 nodes, no storage client

The DaemonSet selected nothing, so nothing turned red. A pool label had drifted.

Alert when coverage < inventory

99.96%

A full NAT table

The metric was there all along. Now it pages.

Alert on saturation, not only on errors

new boot ID

A 204 is not a reboot

```
ready := bootID != "" &&  
bootID != op.BootID &&  
nodeReady(node)
```

NVSentinel remediation ends on a new boot ID + Ready

Alloy via Sveltos → Mimir → Grafana · DCGM · NVSentinel · customers get their own NVSwitch metrics

Four bugs you will write too

`MaxConcurrentReconciles: 1`

One hot loop, **4,900 reconciles in 10 minutes**, starved another controller. Two racks never got their BareMetalHosts.

→ size workers per controller, then fix the loop

`--flags in a random order`

A Kamaji build rendered kube-apiserver flags differently on every reconcile: **~8 Deployment revisions per second**.

→ pin a known-good pair, diff the rendered spec

`ignoreMissingValueFiles: true`

A values file moved. Argo CD rendered Cilium with defaults, **the control-plane VIP vanished**, tenant nodes went NotReady.

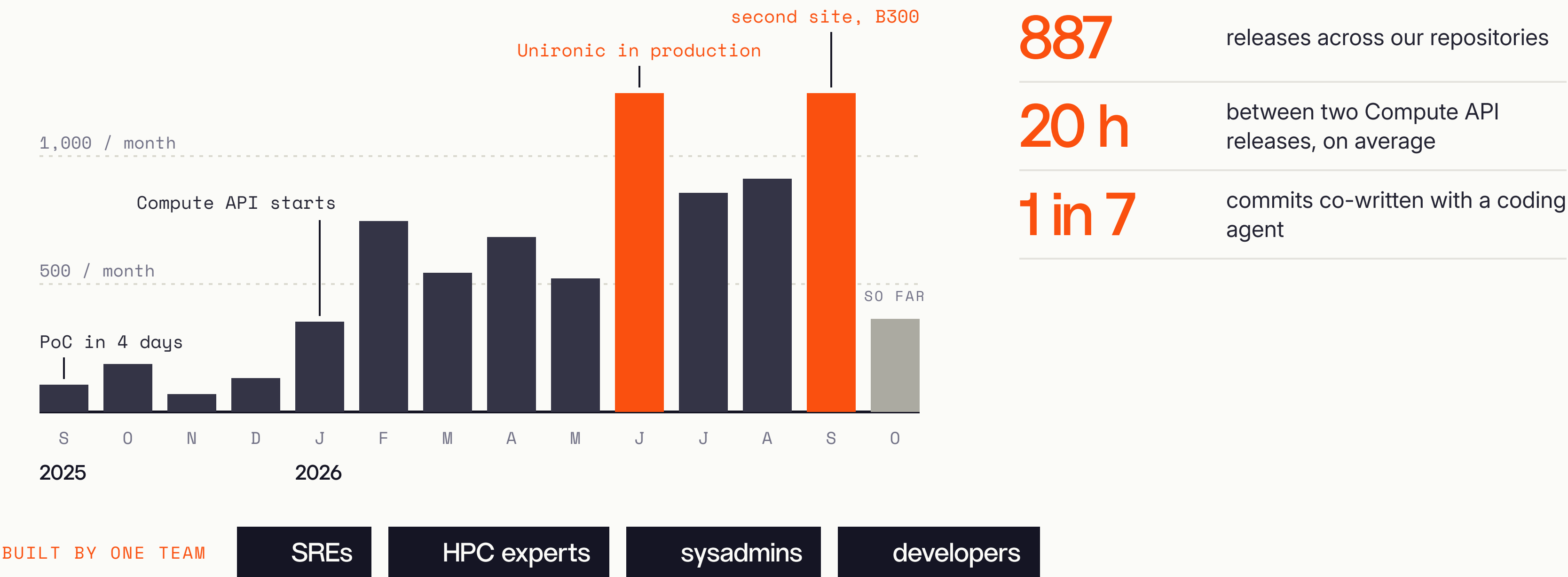
→ fail loudly, sync the CNI first

`?reprovision=2026-09-18`

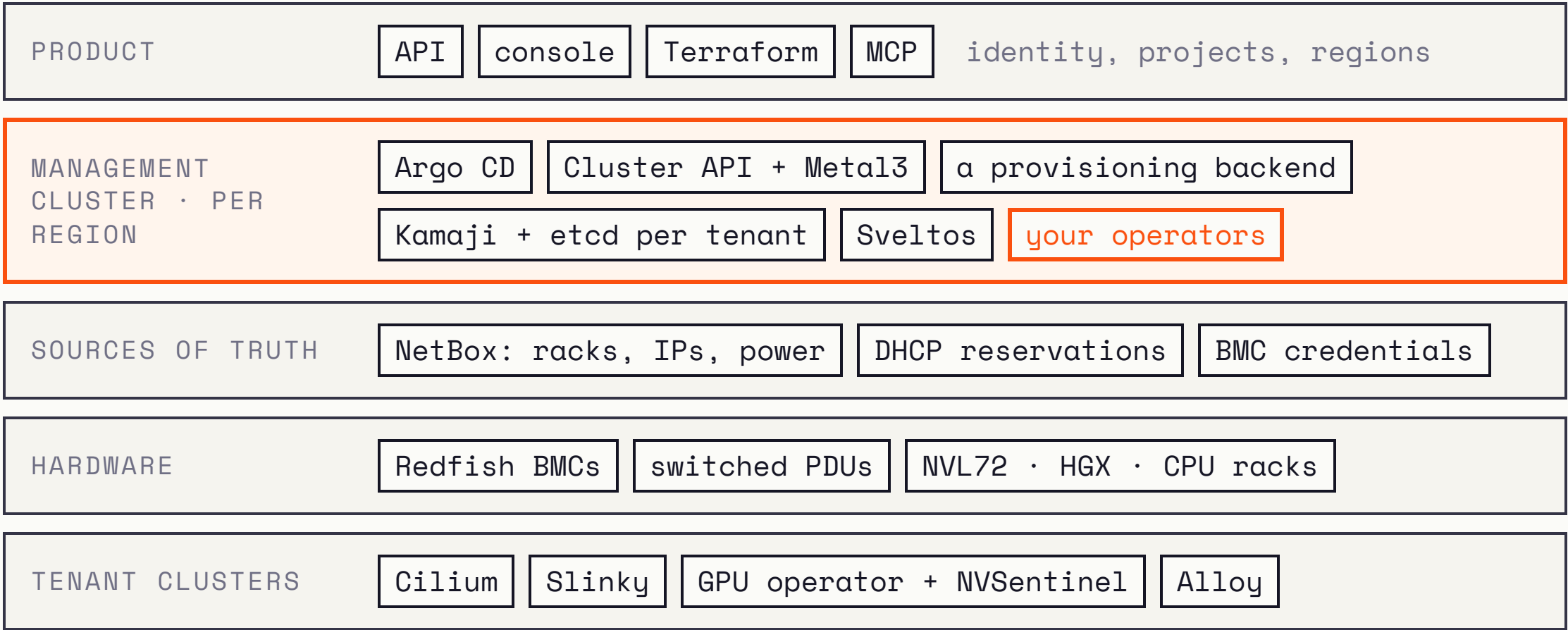
A cache-busting query string on image URLs: **52 GB300s failed every re-image for ~12 hours**. The error lived only in controller logs.

→ errors belong in `.status`

13 months, 7,969 commits, one team



Reference architecture: KaaS on bare metal



FIVE RULES

- 1 Hosts and racks are API objects
- 2 Control planes are pods, etcd is per tenant
- 3 NetBox is the truth, controllers read it
- 4 One hardware profile per model
- 5 Every error reaches .status

From a kind cluster to 5,000 hosts

IN KIND		AT 5,000 HOSTS
A node appears in seconds	BOOT	Firmware, a BMC, a DHCP lease and a disk write
etcd shares the node	ETCD	Its own disks, probe budgets, one per tenant
The pod network is the network	NETWORK	SNAT ceilings, BGP, DHCP, several fabrics
One architecture	CPU	arm64 Grace and x86 in the same fleet
Reboot = delete the pod	REBOOT	Redfish, a cold cycle, sometimes a PDU
Defaults are fine	CONTROLLERS	Worker counts, flag order, ignore flags
Errors show up in your terminal	ERRORS	Errors in .status, alerts on what is missing

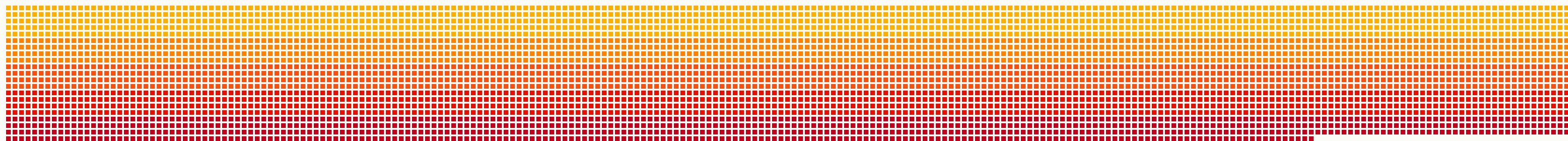


Thank you

Find us after the talk: what is the hardest layer in your GPU platform?

Antoine Roy · [linkedin.com/in/ant-roy](https://www.linkedin.com/in/ant-roy)

Léonard Suslian · [@synthe102](https://twitter.com/synthe102)



Sources

MISTRAL AND PARTNERS

Mistral Cloud timeline and hardware portfolio
`mistral.ai/cloud`

Mistral Compute announcement, 11 Jun 2025
`mistral.ai/news/mistral-compute`

Inference site, Paris region · Mistral, 28 May 2026
`mistral.ai/news/ai-now-summit-2026`

Sweden, 2027 · EcoDataCenter, 11 Feb 2026
`ecodatacenter.tech/press`

NVIDIA AND UPSTREAM

18,000 Blackwell GPUs, initial phase · NVIDIA, 11 Jun 2025
`blogs.nvidia.com/blog/france-sovereign-ai-infrastructure`

Paris-region site, 44 MW, 200 MW across Europe · NVIDIA, 17 Jun 2026
`blogs.nvidia.com/blog/france-advances-europes-ai-future`

Vera Rubin · NVIDIA, 21 Jul 2026
`blogs.nvidia.com/blog/vera-rubin`

GB200 NVL72 and GB300 NVL72
`nvidia.com/en-us/data-center/gb200-nvl72`

Kubernetes node heartbeats and Leases
`kubernetes.io/docs/reference/node/node-status`

Kamaji datastores
`kamaji.clastix.io`