



KubeCon



CloudNativeCon

Europe 2026

#KubeCon #CloudNativeCon

Locking Down Ray Serve: How to Secure Ur ML Models?



Nice to meet you!

Hey there, everyone!

I'm Катерина, Senior Software Developer Consultant,
Lecturer at KPI and KSE, WTM Ambassador, Lead and
co-founder Google Developer Groups

Background: Senior Java Development & DevOps
Capability pool consultant, security champion within
Platform team

And most important -- I'm totally in love with ML. So I
do a lot of MLOps lately :)

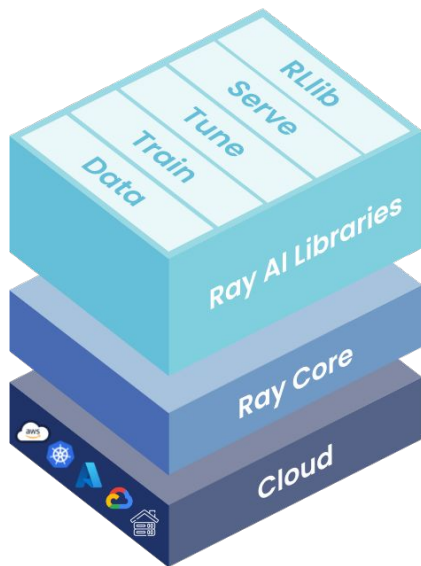


I have quick question for you :)



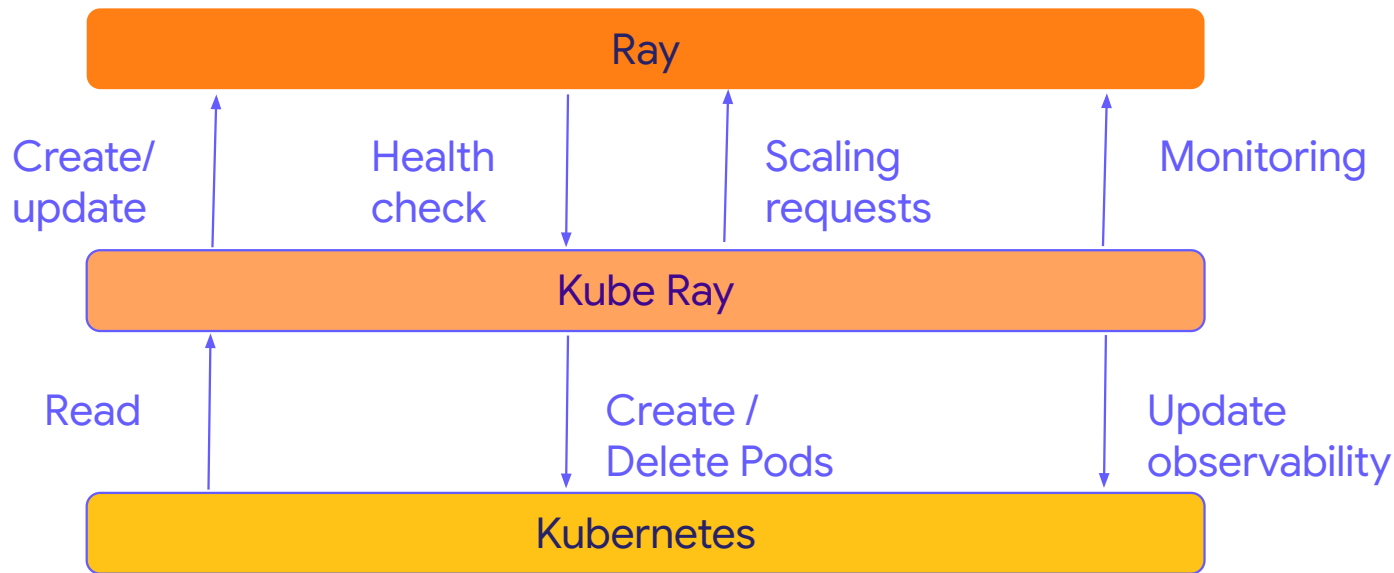
Who here works with Ray or heard of it?

What is



Open-source framework for ML. It can do a lot, but mainly it's managing low-level processes and the resources they operate on, using a Python API

Encapsulates the functionality of a distributed process manager



01 Ray Cluster

02 RayService =
Ray Cluster + Ray
Serve

03 RayJob = Ray
Cluster + Job

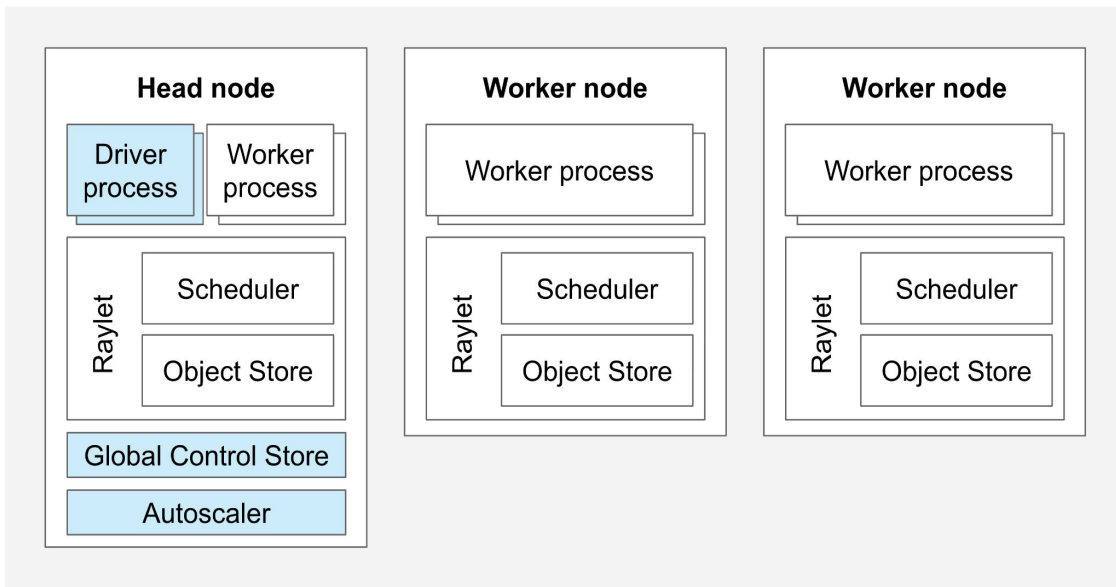


Insides of Ray cluster

Look oddly familiar

Like a K8s Control Plane, Ray utilizes a Head Node to manage cluster state and resource scheduling

But under the hood it's just worker node. So technically you may run the regular processes there. **But you should not**



KubeCon



CloudNativeCon

Europe 2026

U may often hear Ray cluster being referred as “flat”, which means that...

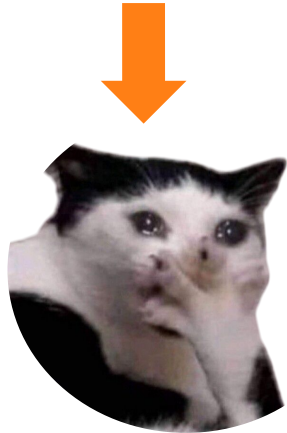


And additionally, as we know already - **Ray allows to run arbitrary code** with Ray Jobs, Ray Client or Ray Dashboard APIs

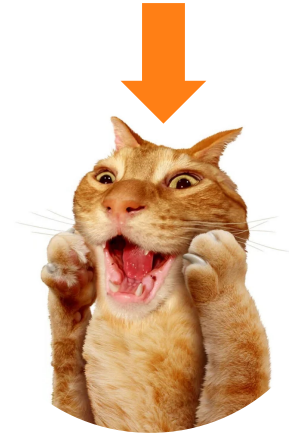
So if you leave Ray cluster to be as it is, with default config, it may be interpreted... **As an open invitation for anyone to execute code on your GPUs**

So sth like this might happen

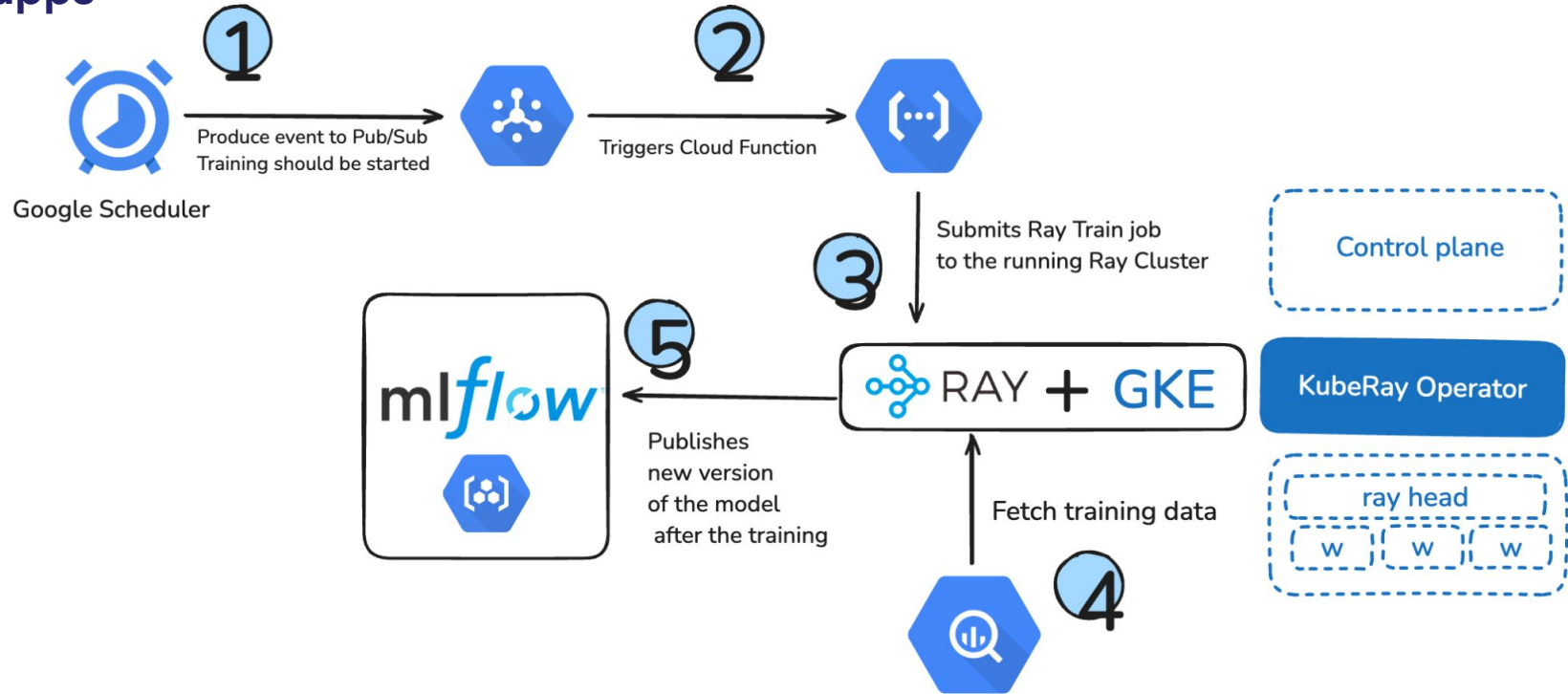
The one who
configured Ray
cluster



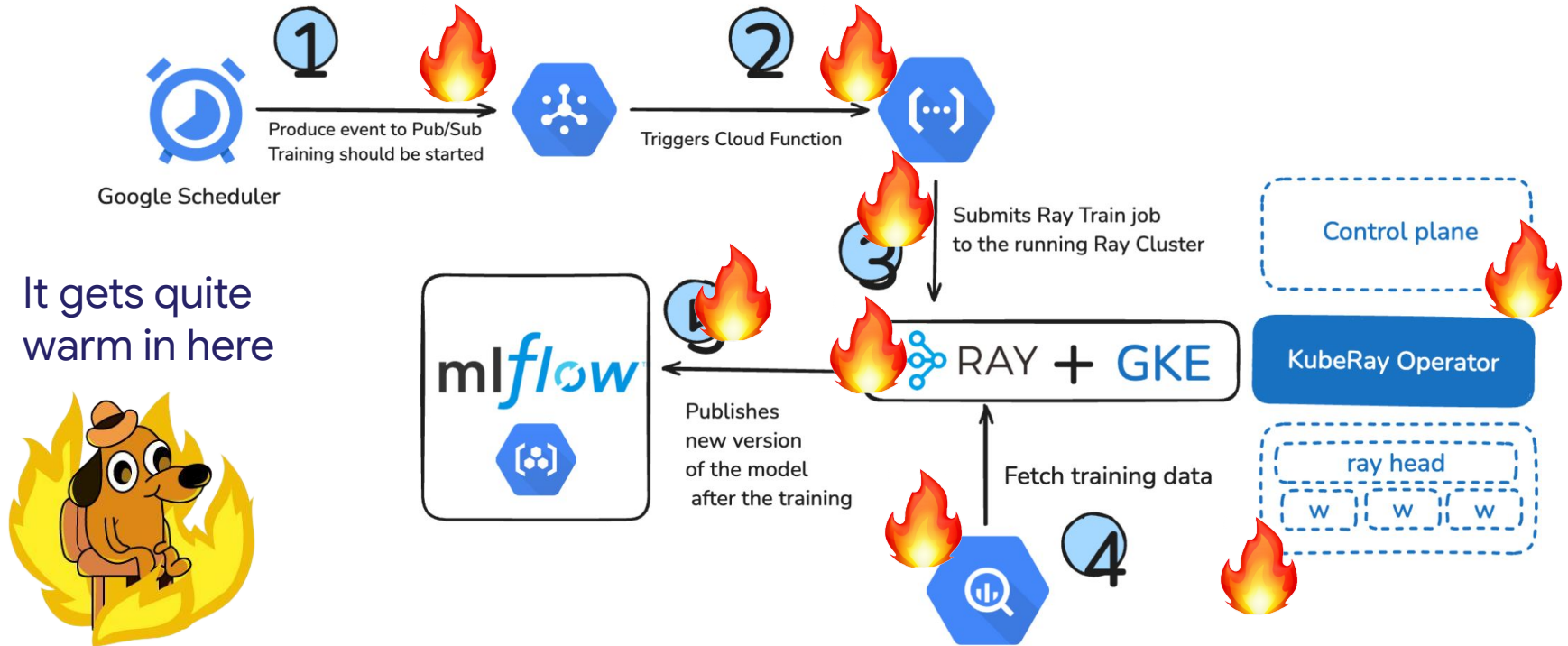
The one who will
pay the bill



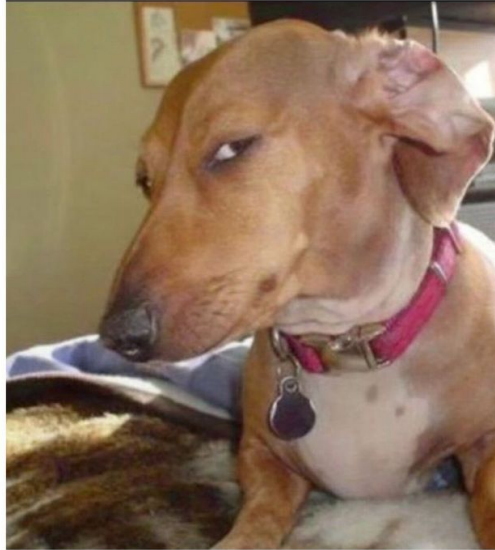
When we talk about ML apps security, we need to remember. **ML apps are still apps**



So they burn in all places regular apps would and in some extra as well

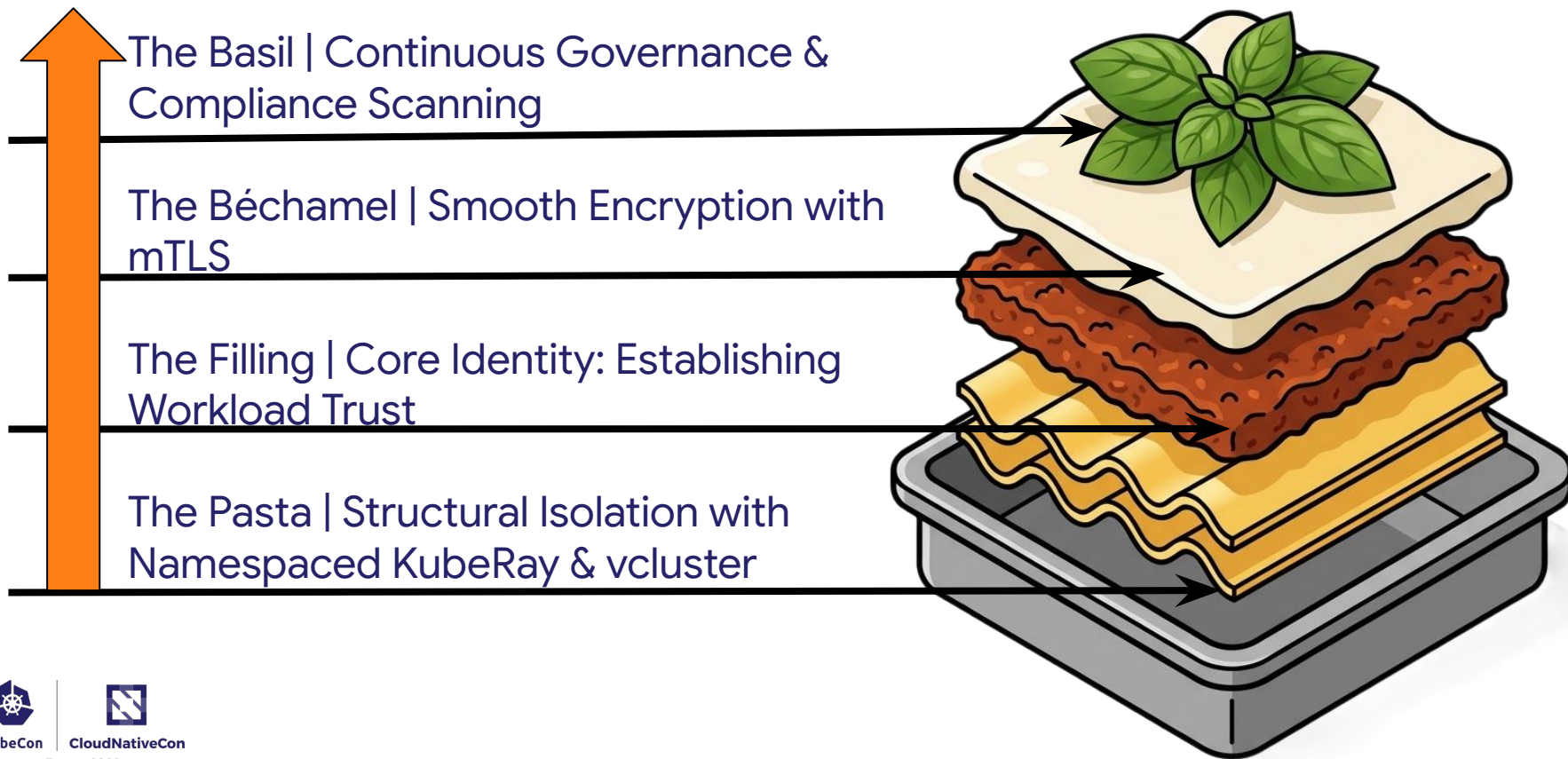


So what we gonna do about it?



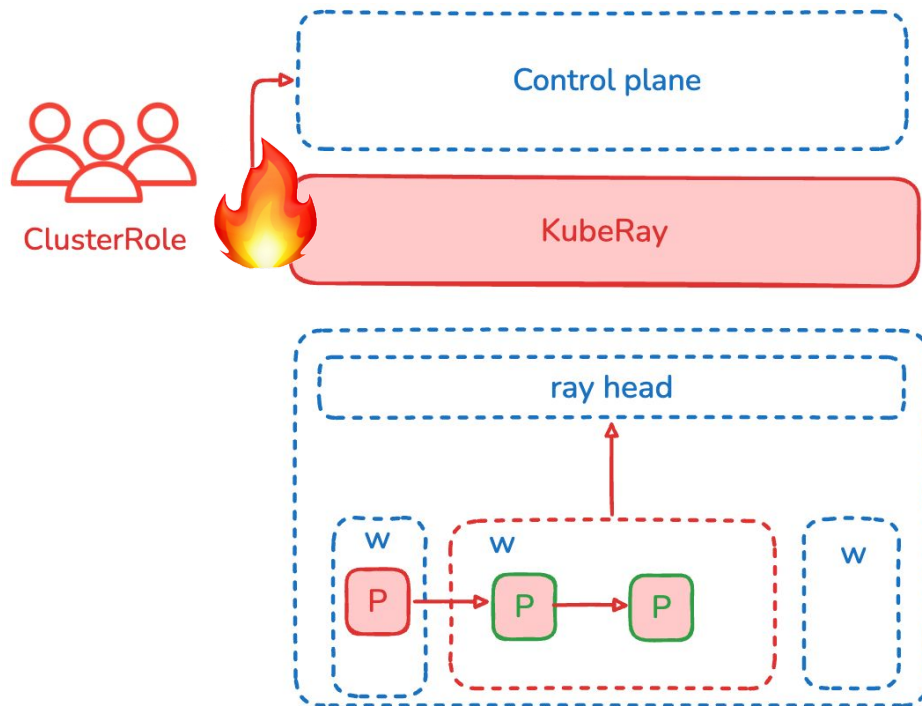
... well :)

Let's add some layers :) Meet fabulous security Lasagna





The Pasta: On a Mission to Protect Ur Control Plane



KubeCon

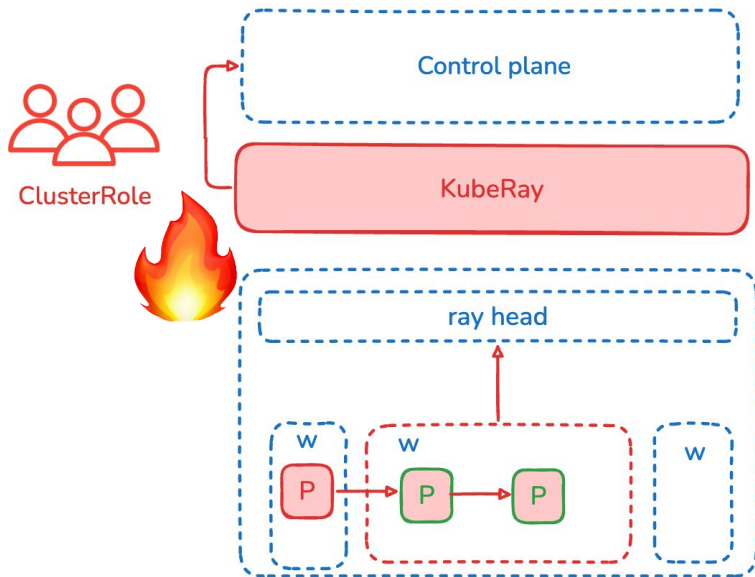


CloudNativeCon

Europe 2026



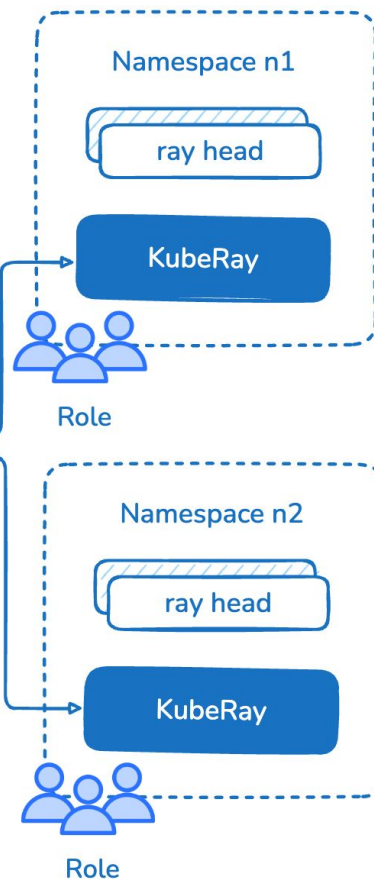
The Pasta: On a Mission to Protect Ur Control Plane



Much better
now... huh?

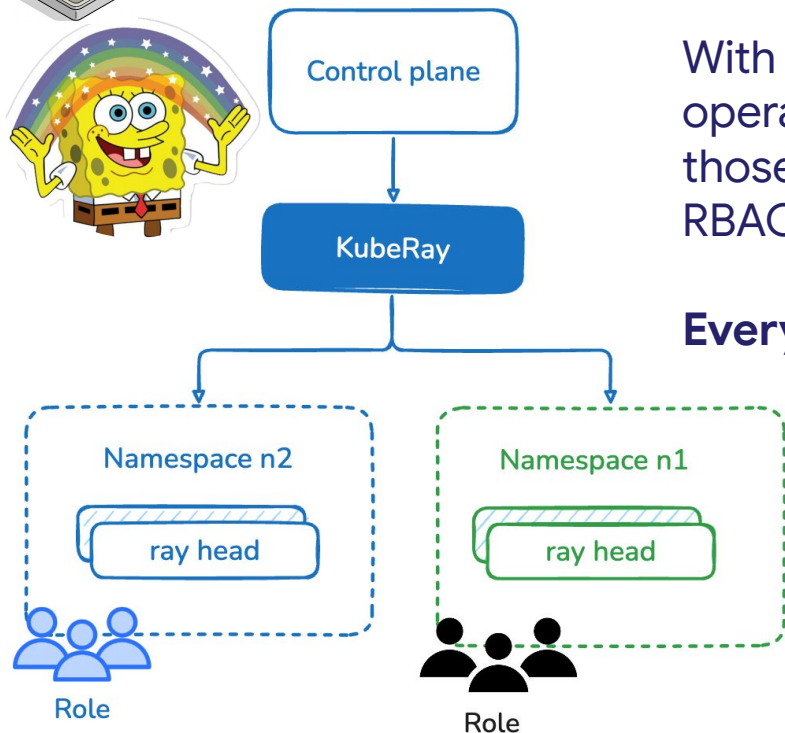
rbacEnable

singleNamespaceInstall





No limit to perfection: 1 KubeRay, many Namespaces

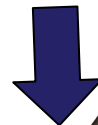


With **watchNamespace** we enforce Kuberay operate within couple of namespaces. With those namespaces can also configure some RBAC, Network Policies, Limit Range, etc...

Everyone should be happy right?

Not exactly, because we overseen another problem. Global CRDs is least of them

SRE



KubeCon



CloudNativeCon

Europe 2026



So let me get this straight... U want isolated environment to do all the ML stuff BUT do that without providing KubeRay access to the control plane???

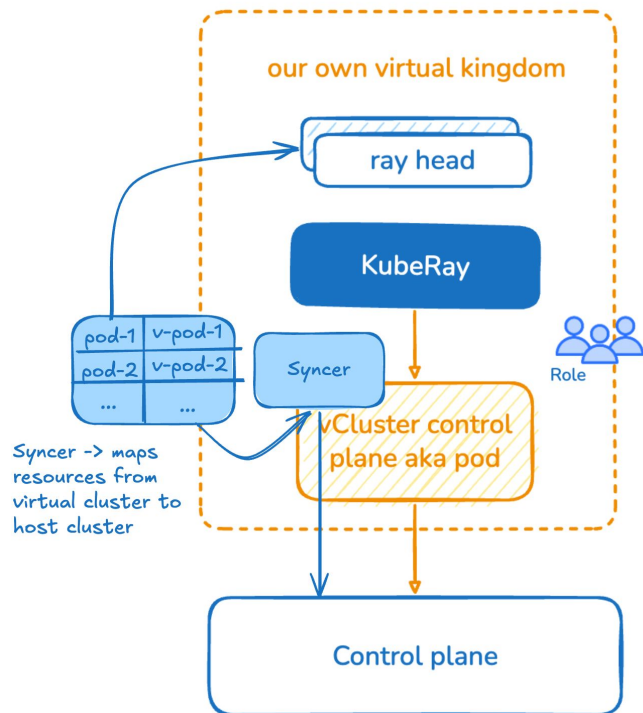


Very much
Yes. I do.



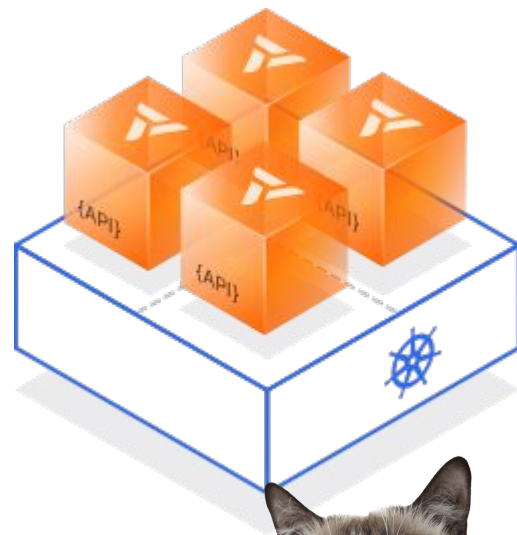


Good news: there is a solution to do just that



Open-source tooling, that allows us to do whatever we please inside our little kingdom

Our own namespaces, policies, CRDs, and our own RBAC. We are 100% decoupled from outside cluster



Yay for that!



Kyverno



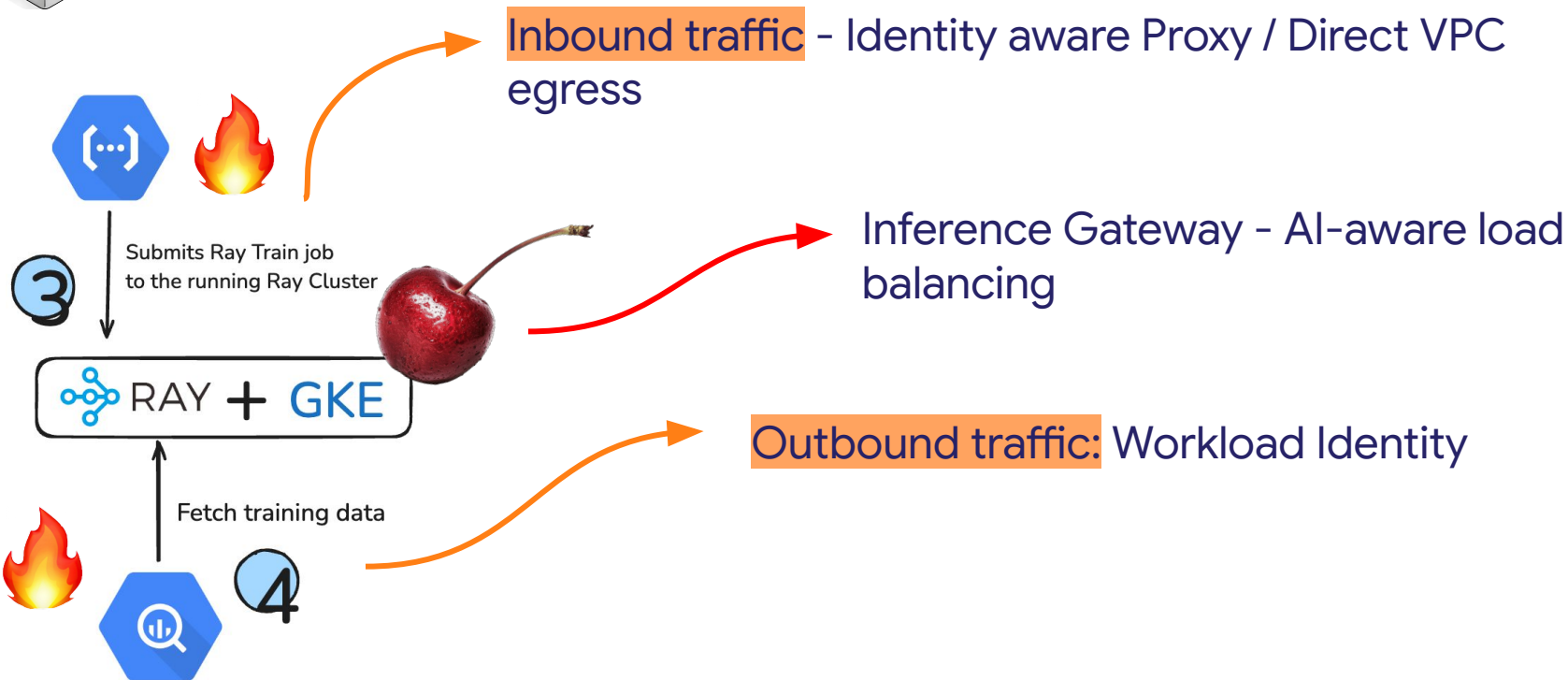
3 things we should consider

- **Network:** Label-based micro-segmentation (Workers can communicate with Head only, head node - can communicate with control plane).
- **RBAC:** API-less Workers. Preventing lateral discovery between jobs.
- **Resource Governance:** LimitRange as the final safety valve for the podTemplate that KubeRay uses





The Filling: some doors meant to be closed. So let's do it!



KubeCon

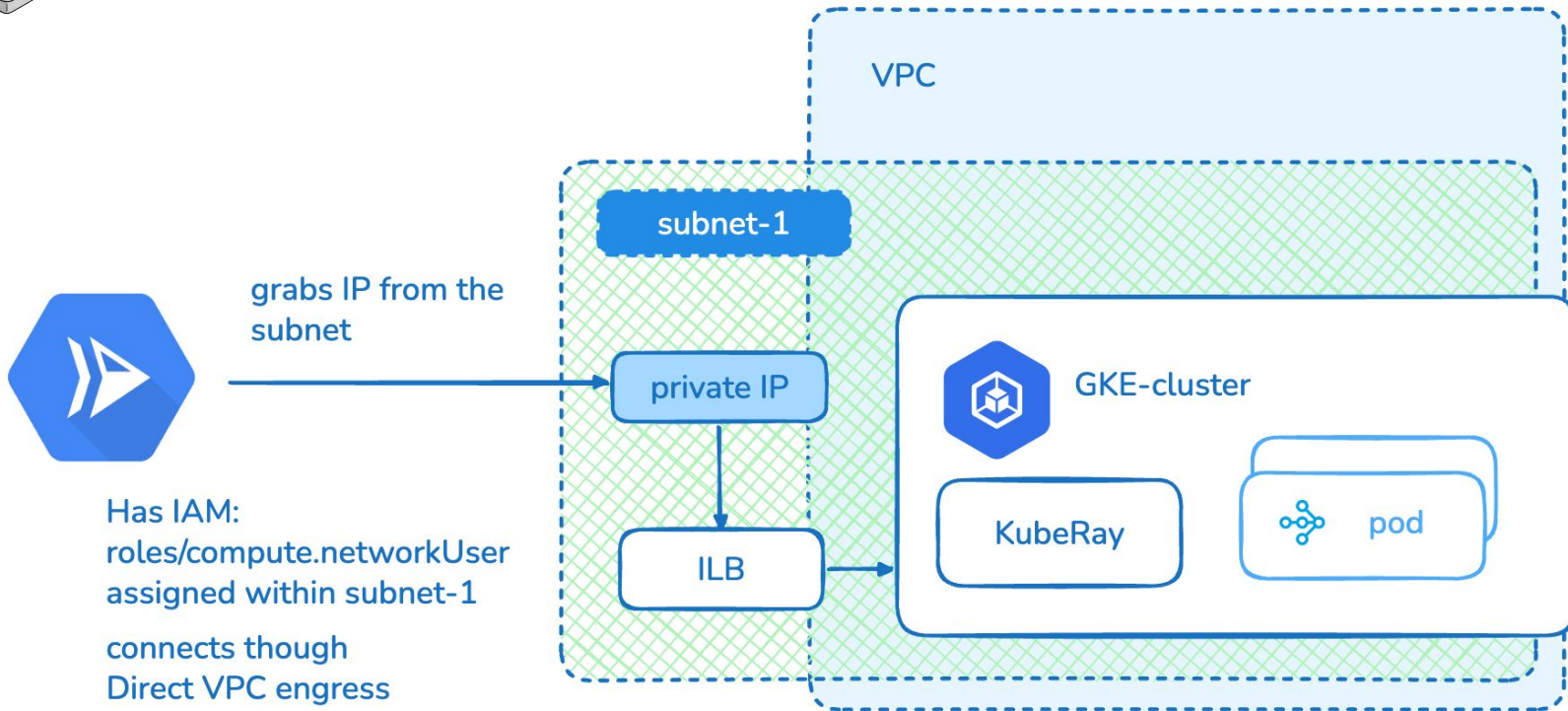


CloudNativeCon

Europe 2026



Inbound Traffic: Direct VPC egress - when it's convenient?



KubeCon

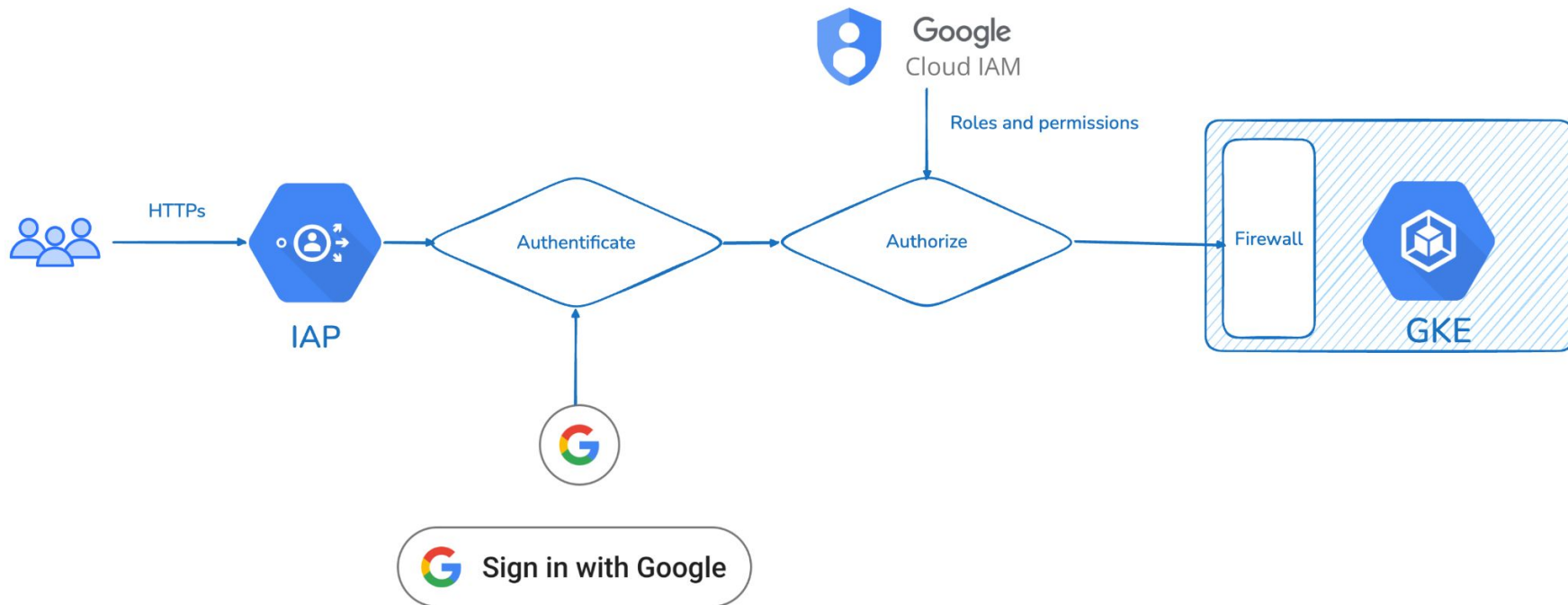


CloudNativeCon

Europe 2026



Inbound traffic: Identity Aware Proxy and when should we use it



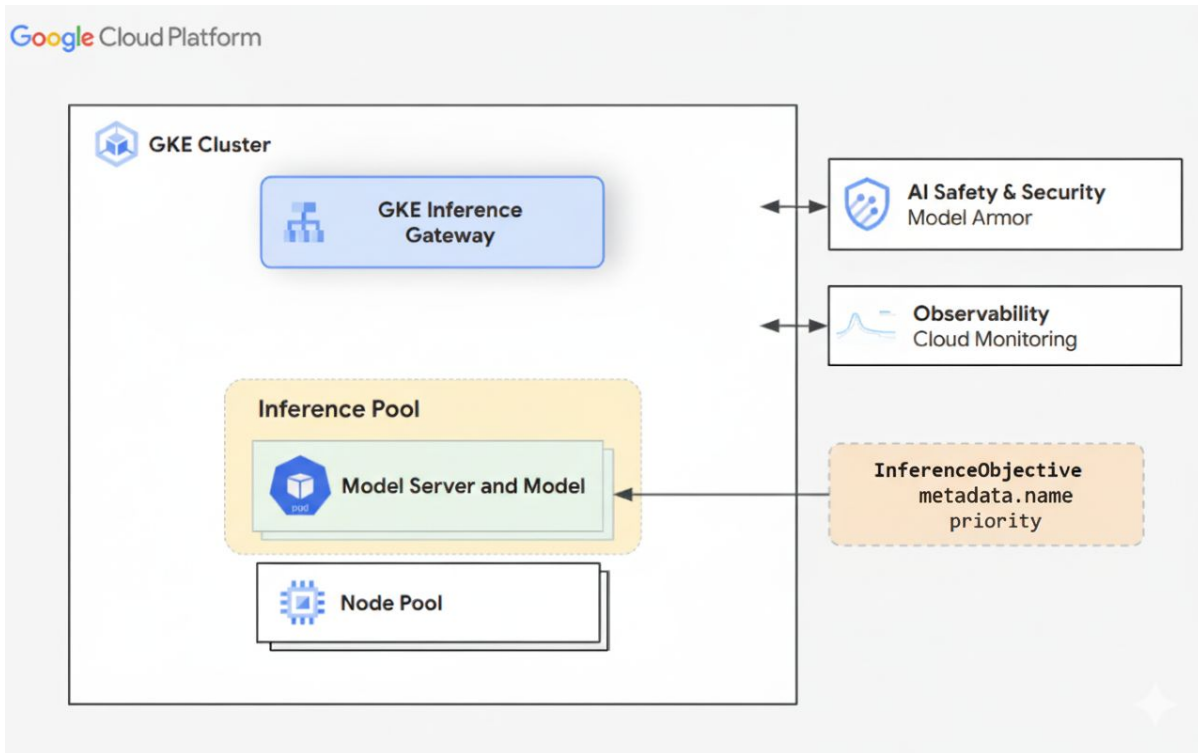


Next step? AI-aware Load Balancer

Inference Gateway talks same language as ML models

The key feature is AI-aware CRDs:

- InferencePool
- InferenceObjective

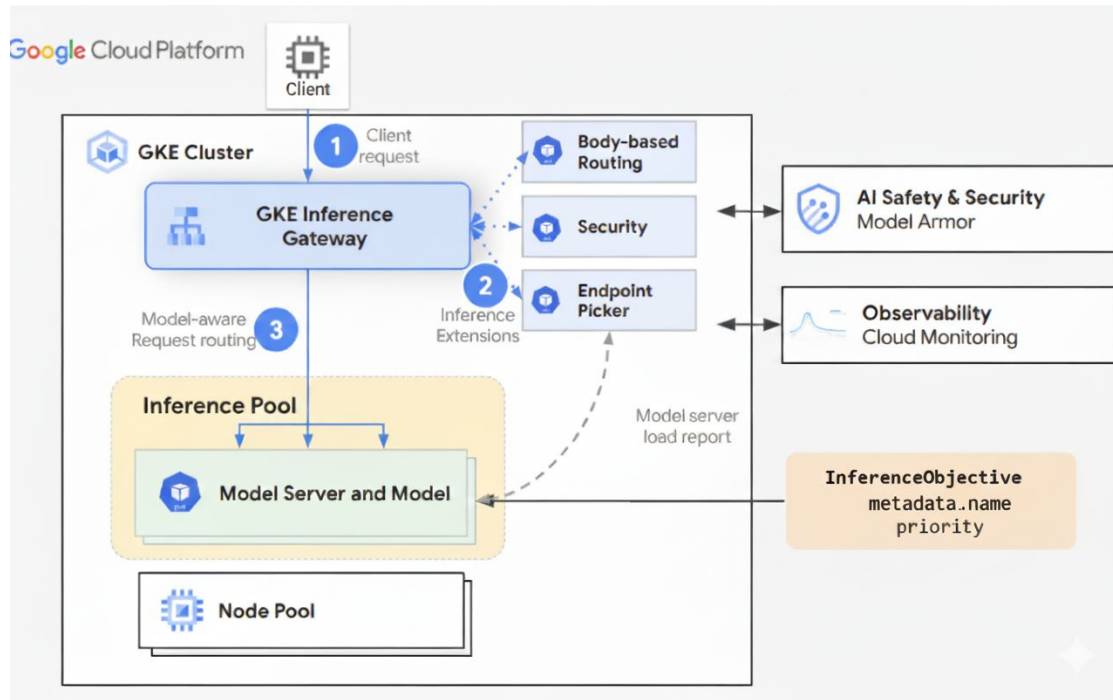




How exactly GKE Inference Gateway “understands” AI needs?

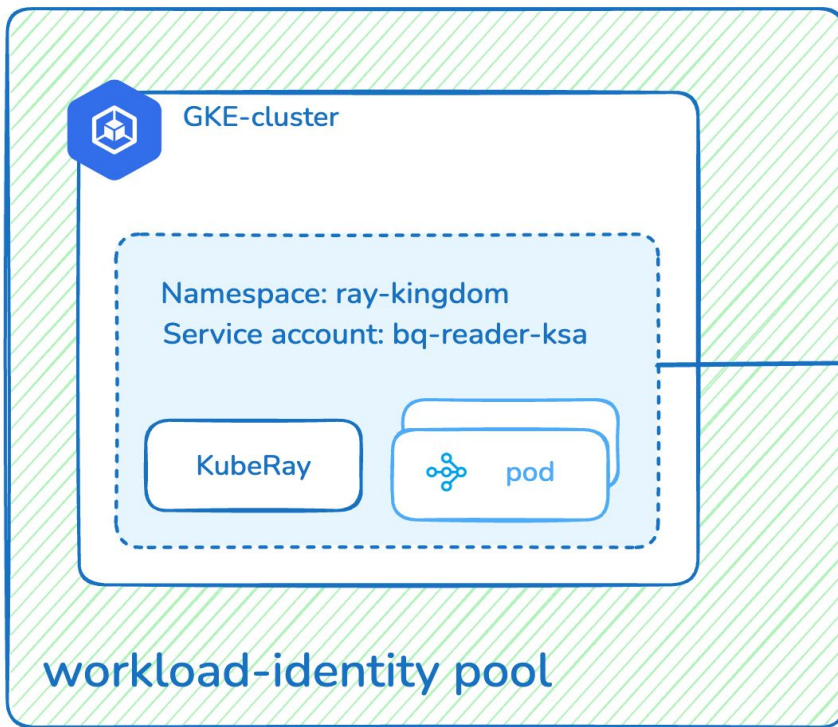
Combination **Body-based Routing, Security (Model Armor)** and **Endpoint Picker** 3 heuristics:

- Prefix Cache Aware Routing
- LoRA Aware Routing
- Load Aware Routing

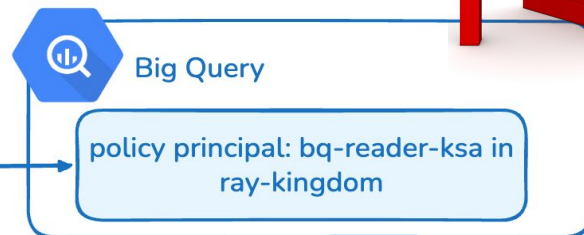




Outbound traffic: Workload identity



Like having private door to GCP APIs



But be aware...



KubeCon



CloudNativeCon

Europe 2026



How to befriend Workload Identity and vCluster



GSA



KSA

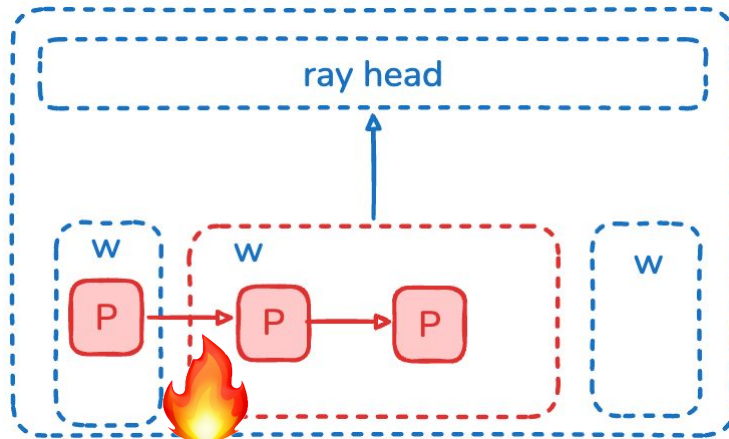


vCluster KSA



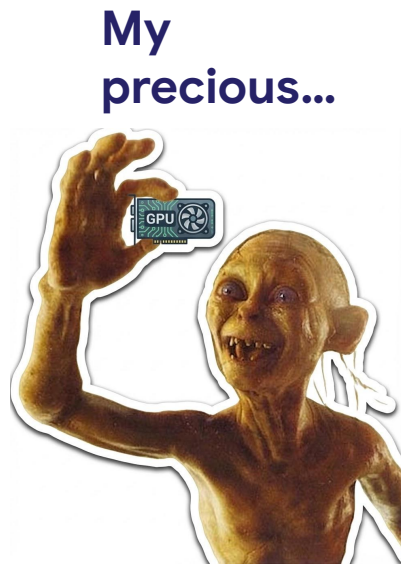


The Béchamel: Zero trust model for extra safety



We cannot block Worker to Worker communication, Ray workers often share weights, etc. Someone, getting the access to our ray cluster and submitting the Ray job on our precious gpus.

How we prevent that?



KubeCon



CloudNativeCon

Europe 2026



Enabling mTLS for Ray Cluster. How we gonna do it?

We could always go with the golden standard



Istio / CSM



That's getting complicated

But of course there are nuances... we need to decide - which one it will be:

Sidecar ? Ambient ? Proxyless gRPC...



or even shockingly configure mTLS with Ray Native way?



KubeCon



CloudNativeCon

Europe 2026



Let's checkout some decision matrix, shall we?

	Ray Native mTLS	Istio Sidecar	Ambient Mesh (ztunnel)
Latency	High (Handshake per request)	Moderate (Double-proxy hop)	Low (Persistent L4 tunnels)
CPU Overhead	High (App-level crypto)	High (Envoy per pod)	Minimal (Shared node proxy)
OpEx (Toil)	High (Manual cert rotation)	Moderate (Sidecar lifecycle)	Low (Managed by CSM)
Security Scope	Ray Internal Only	Full L7 (HTTP/gRPC)	Full L4 + Optional L7



Ray Token Auth (v2.52+) - What's up with it and do we really want to use it?



Now we can secure the Ray APIs with token

Token is either generated on `init()` action, or via separate command on cli

```
apiVersion: ray.io/v1
kind: RayCluster
metadata:
  name: ray-cluster-with-auth
spec:
  # enabling the auth, it will generate and set
  # RAY_AUTH_TOKEN and RAY_AUTH_MODE env vars
  authOptions:
    enabled: true
  headGroupSpec:
    ... # Operator handles the rest
```

Ray Token doesn't support granularity on access -- the token is just passcard, and all services that have it will have the same privileges



Governance and Compliance with **PROWLER**

Most important thing about security is that it should be **consistent**. That's not some switch you may turn on once and then just forget about. We need persistent monitoring and scanning **Prowler** is open-source tool that will solve the issue for us. We may scan for:

Infrastructure

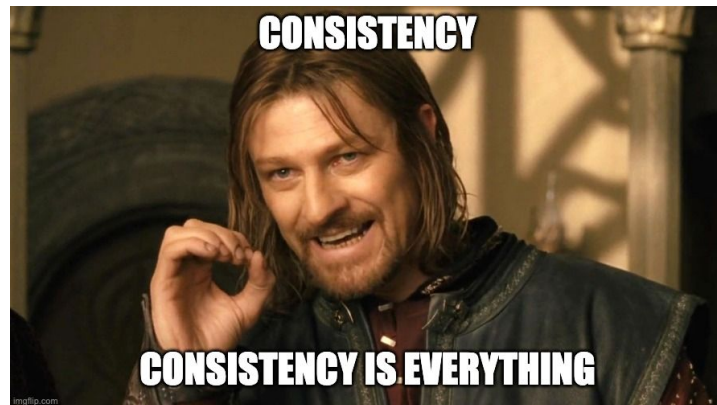
- Public Exposure
- Identity
- Over-privilege
- Logging & Monitoring
- Storage Security

Cluster

- RBAC Hygiene
- Network Policies
- Pod Security
- CIS Benchmarks

Workload

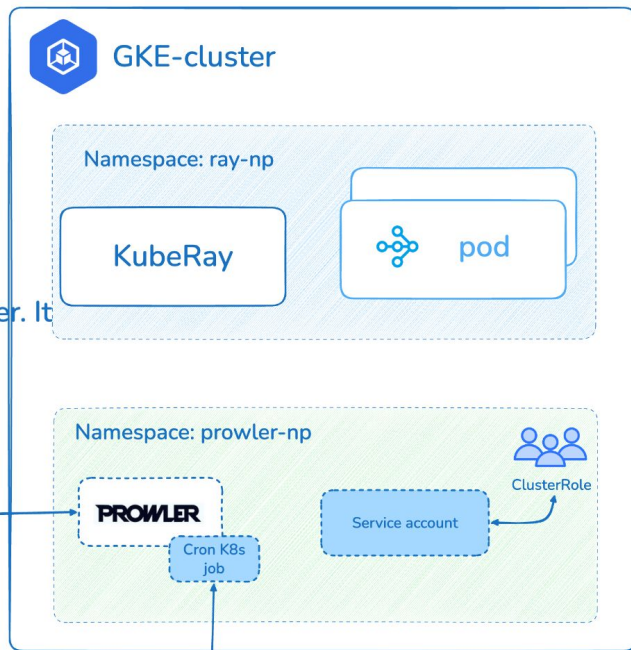
- Exposed APIs
- Secrets Management
- Resource Limits





Install prowler on cluster. It provides the definitions for the main components:

- job
- service account
- role
- role binding



To allow prowler to perform scanning across namespaces - we need ClusterRoleBinding

Will scan for the misconfigurations



My blog

Feedback



Locking Down Ray Serve: How to Secure Ur ML Models?