



KubeCon



CloudNativeCon

India 2026

#KubeCon #CloudNativeCon

Kubernetes Plane **Aerodynamics**: Breaking the Architectural **Sound Barrier**



Cluster Architecture

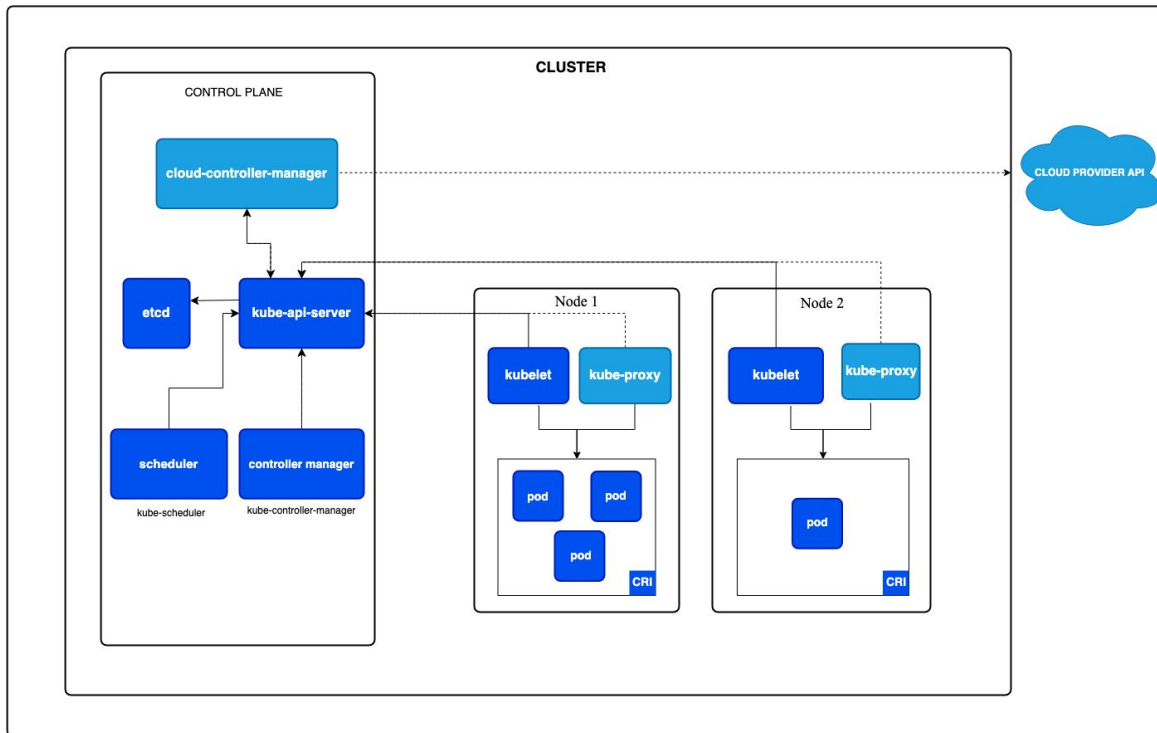


KubeCon



CloudNativeCon

India 2026



From the documentation:

The architectural concepts behind Kubernetes.

A Kubernetes cluster consists of a **control plane** plus a set of **worker machines**, called nodes, that run containerized applications. Every cluster needs at least one worker node in order to run Pods.

...



you must unlearn what you have learned!



Dr. Stefan Schimanski
Principal Engineer



- ex-Redhat, ex-Upbound
- SIG API Machinery Tech Lead
- Co-Creator of CRDs
- ex-Architectural Lead of kcp
- Inventor of kube-bind

- Implemented agents on a physical C64

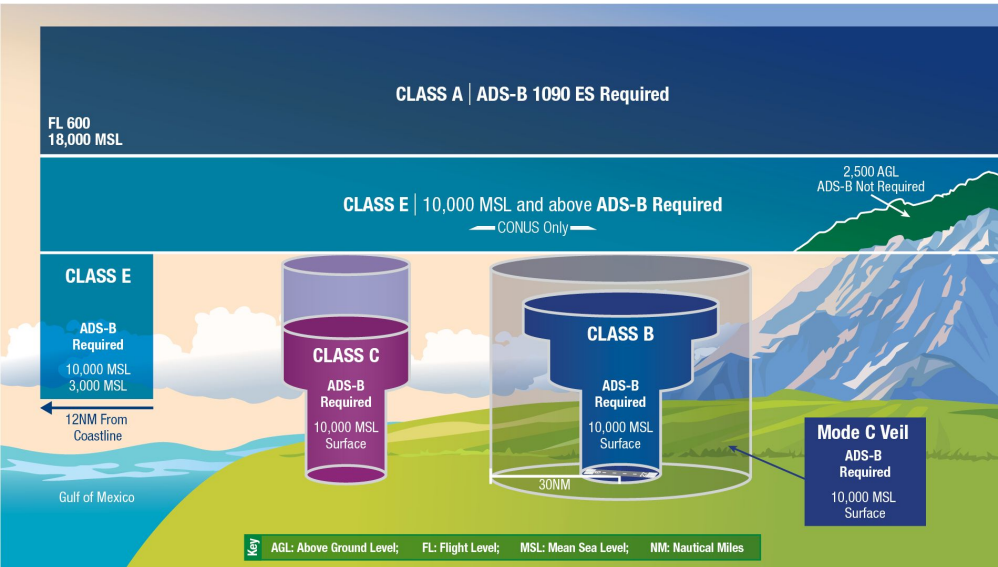


Vasu Chandrasekhara
Independent
NeoNephos Advisor

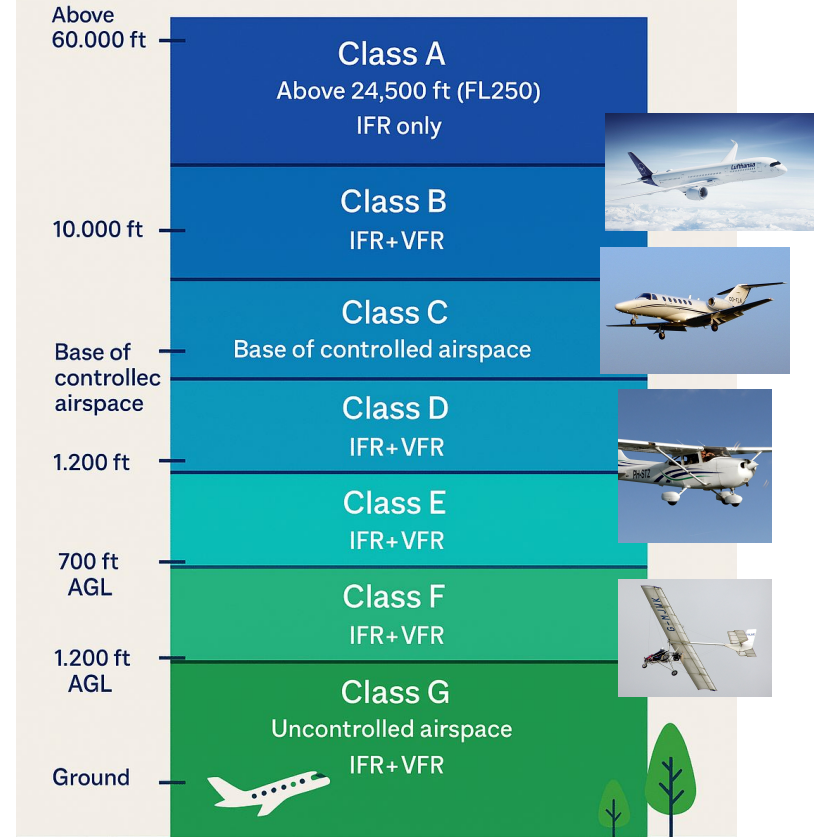


- ex-SAP VP
- Established Kubernetes in the SAP Business Technology Platform
- IPCEI-CIS Project Apeiro CPO that led to NeoNephos Initiative at LF Europe

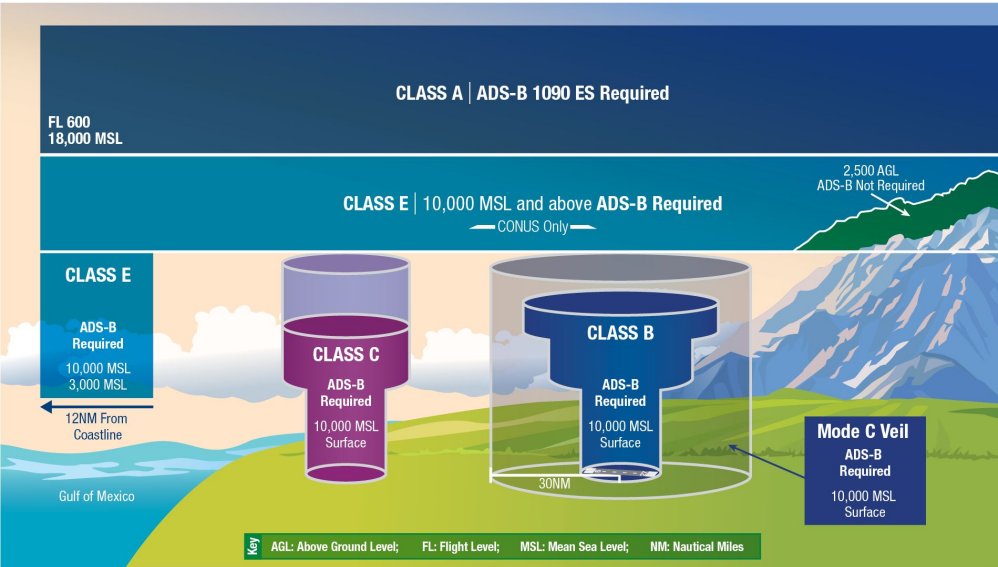
- Accused of having my head in the clouds



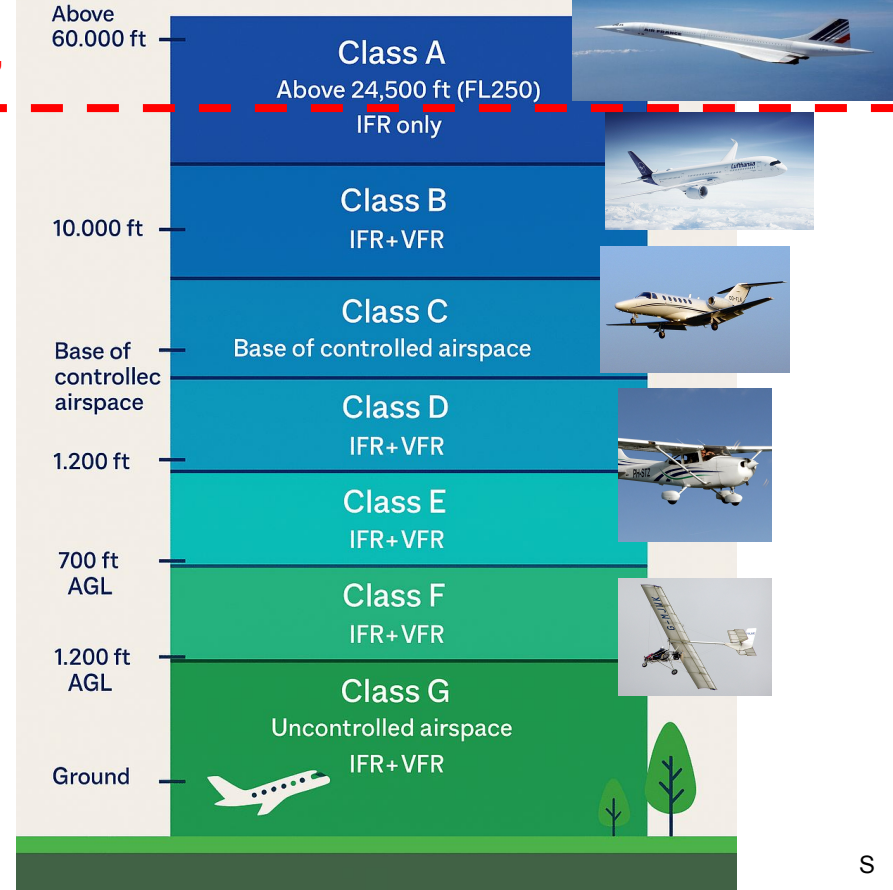
Indian Airspace Classification



If you want to fly fast and far,
you have to fly high. - - -



Indian Airspace Classification





KubeCon



CloudNativeCon

India 2026

First, let's dissect the architecture and establish terminology



Our Planes

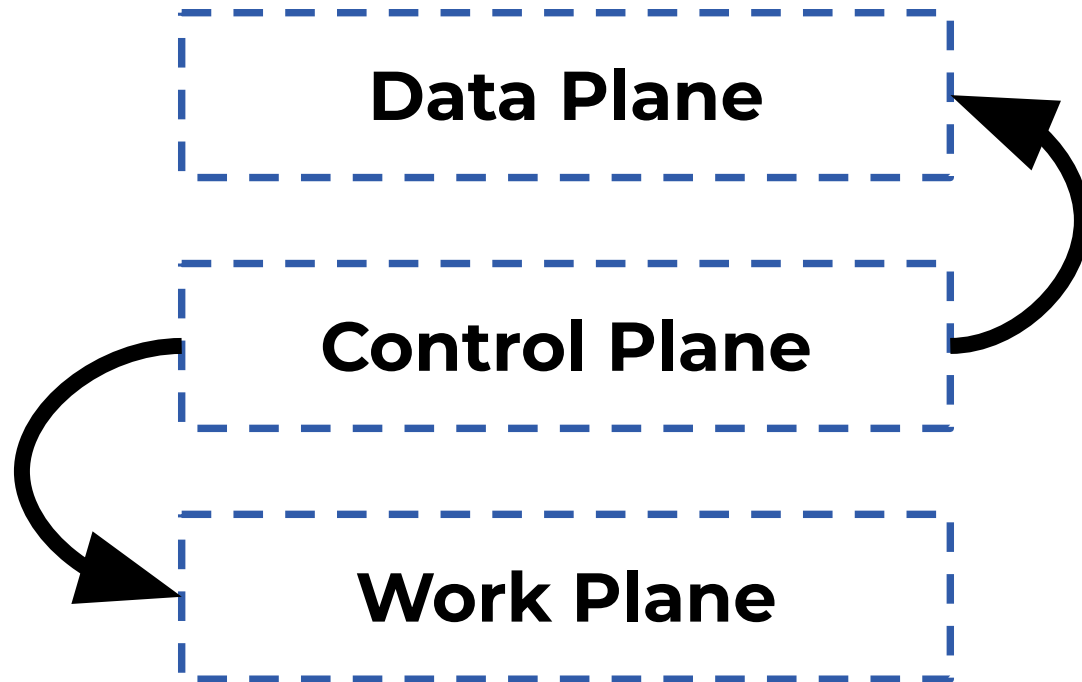


KubeCon



CloudNativeCon

India 2026



Data Plane

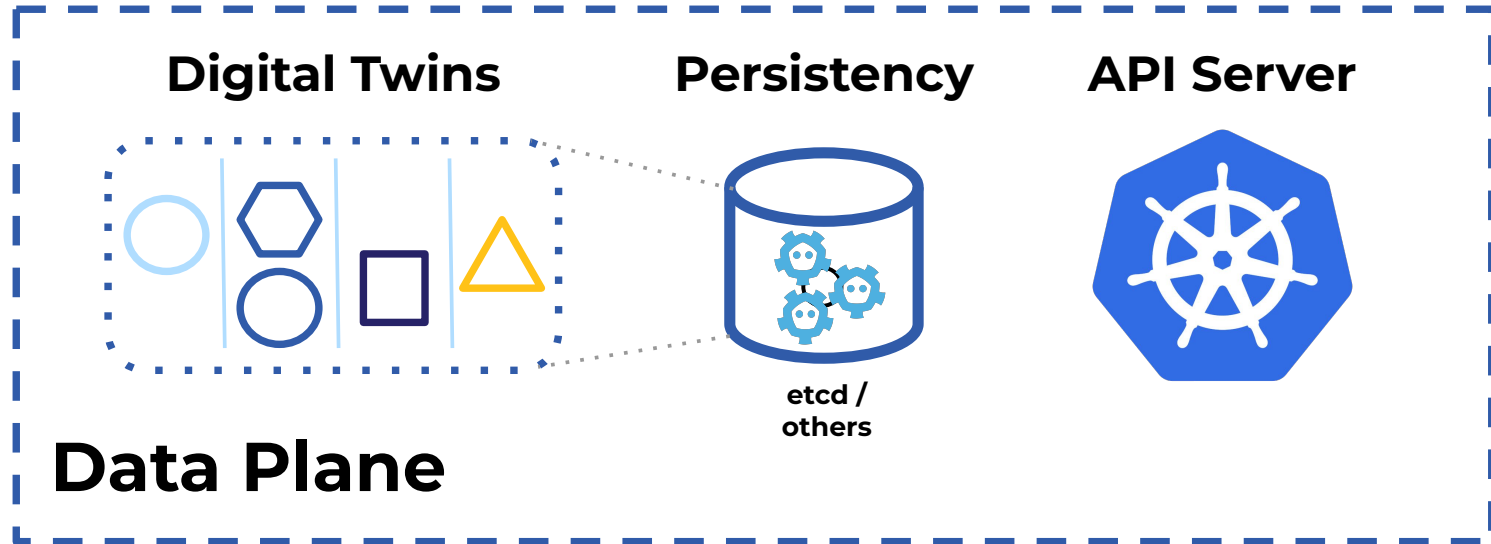


KubeCon



CloudNativeCon

India 2026



Control Plane

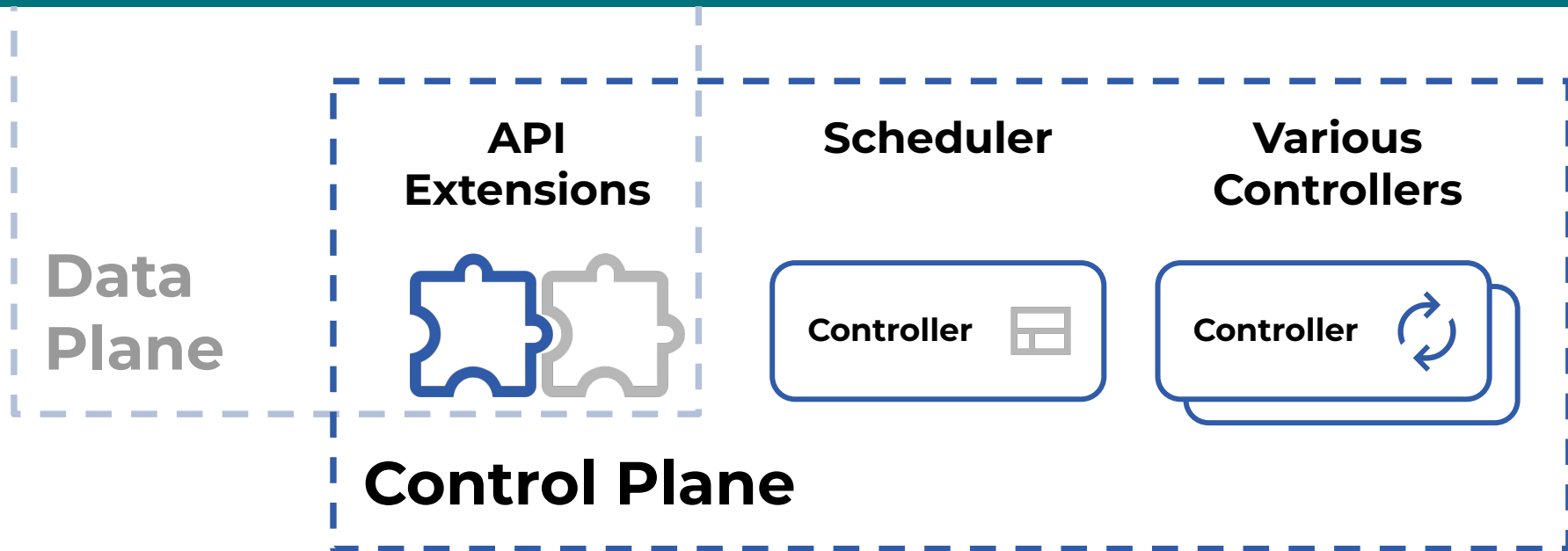


KubeCon



CloudNativeCon

India 2026



The API Extensions (CRD or built-in) semantically *belong* to the controllers (Control Plane). But they are technically *activated* in the API Server (Data Plane).

Work Plane

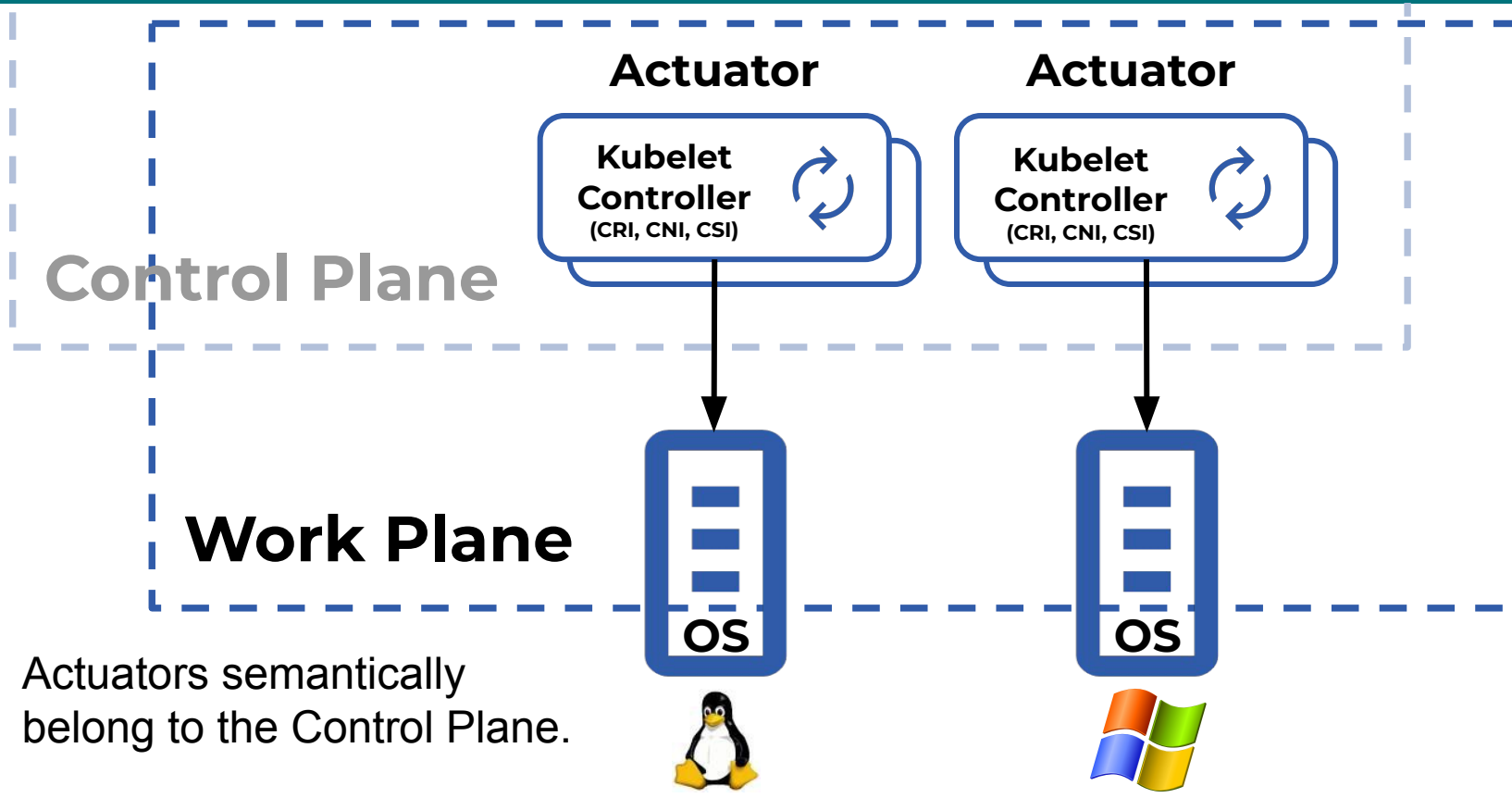


KubeCon



CloudNativeCon

India 2026



Control Plane

Actuator

Kubelet
Controller
(CRI, CNI, CSI)

Actuator

Kubelet
Controller
(CRI, CNI, CSI)

Work Plane

OS

OS

Actuators semantically
belong to the Control Plane.



Putting all together

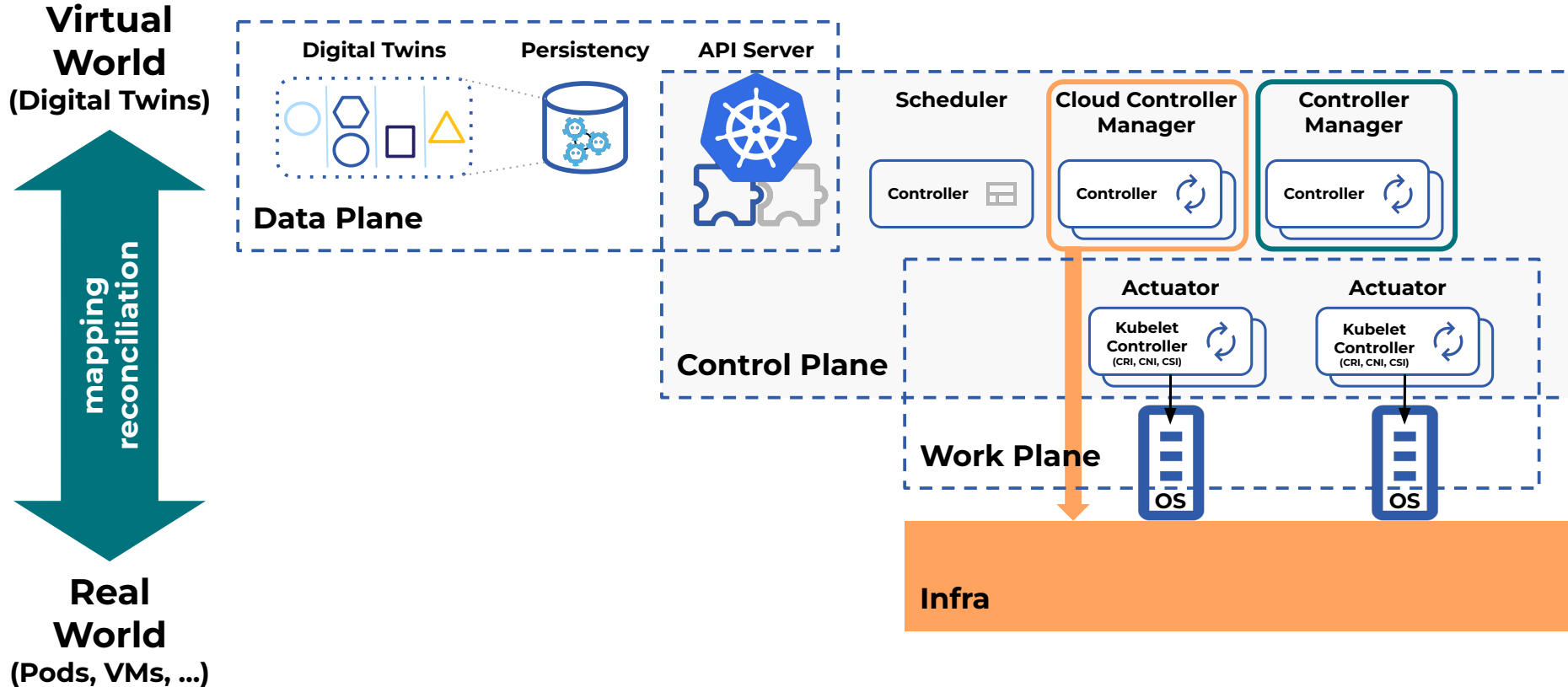


KubeCon



CloudNativeCon

India 2026



Putting all together (2)

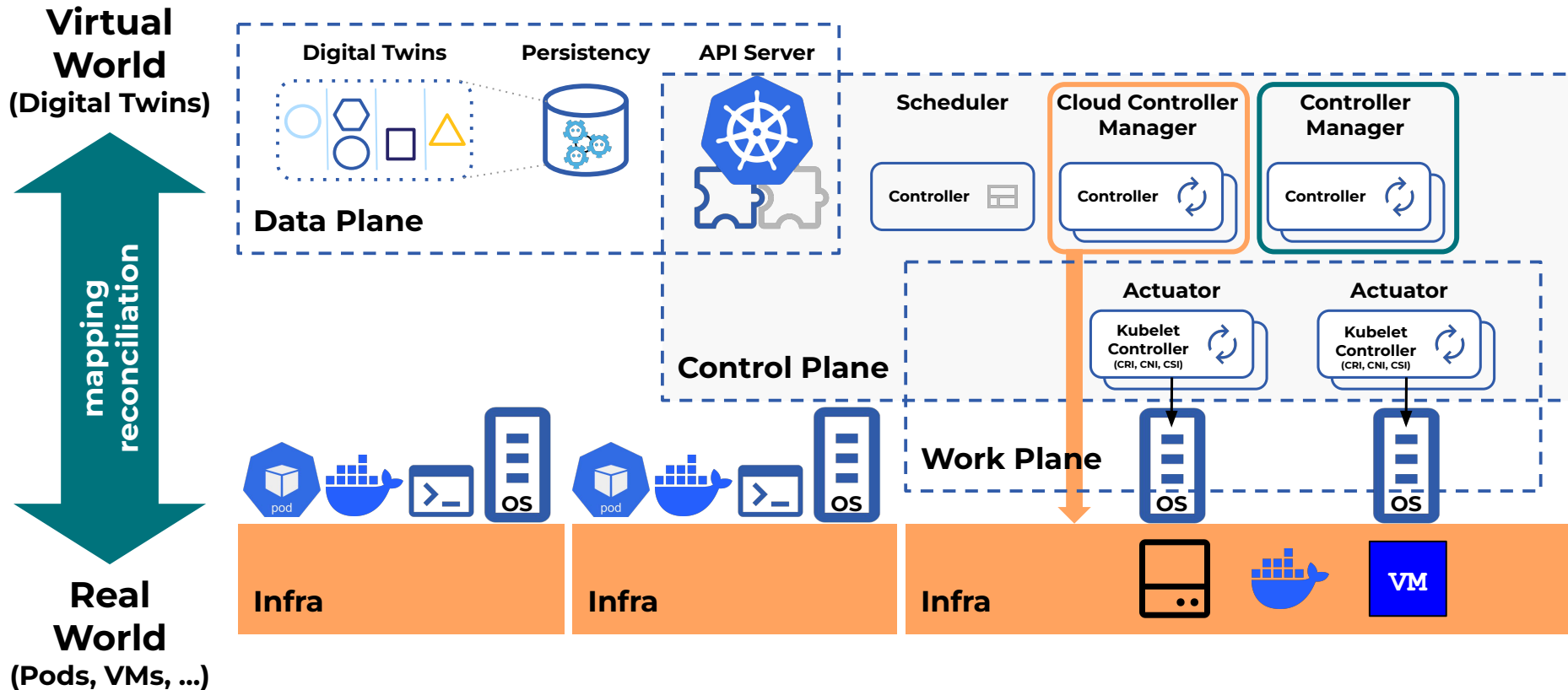


KubeCon



CloudNativeCon

India 2026



Ground and Controlled airspace

Logical vs. Physical view



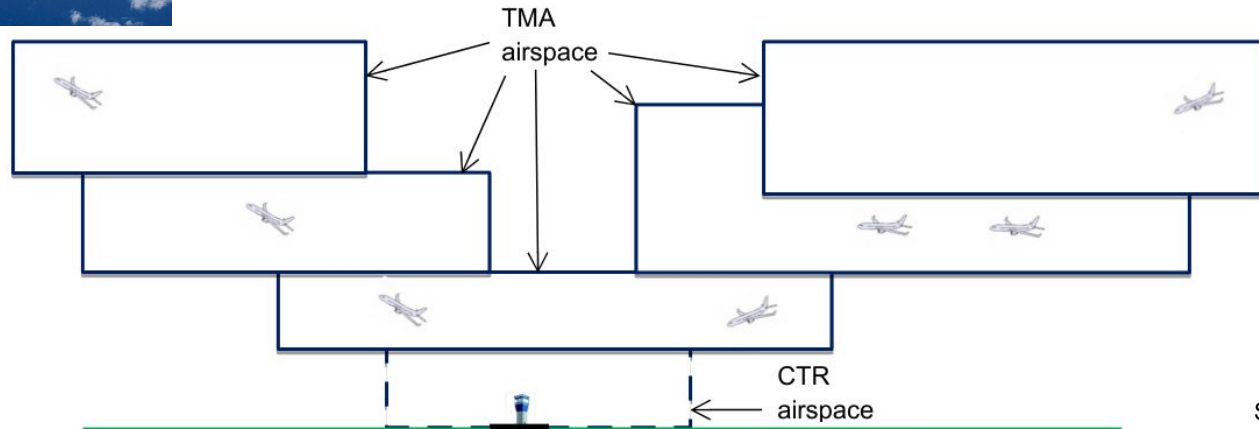
Con



All navigational infrastructure is originally rooted at the ground, but not necessarily at airports (like the VORs)



TWR
eddm



VOR
VHF Omnidirectional Range.

Planes in Space (not a Space Oddity!)

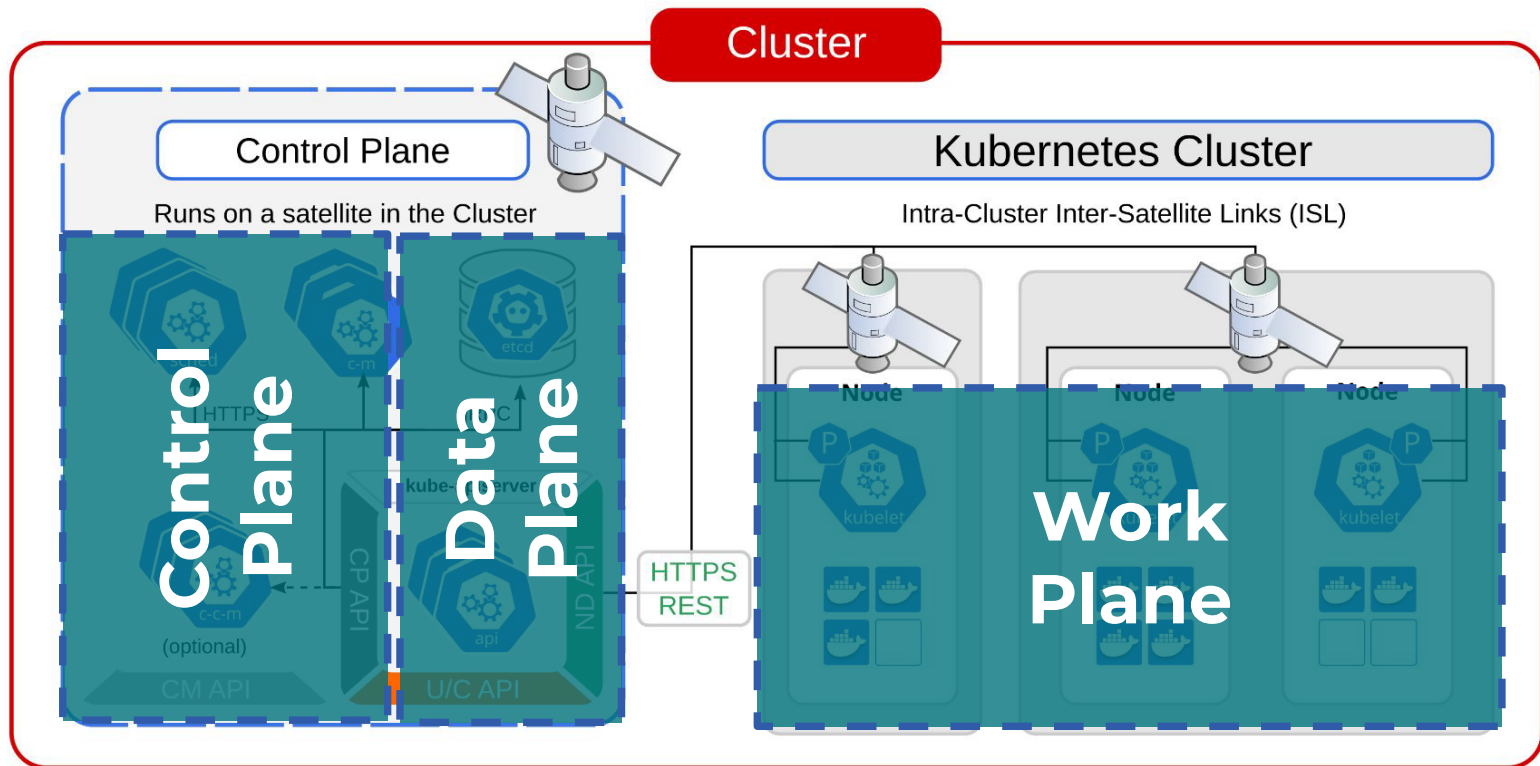


KubeCon



CloudNativeCon

India 2026



Ground Control to Major Tom

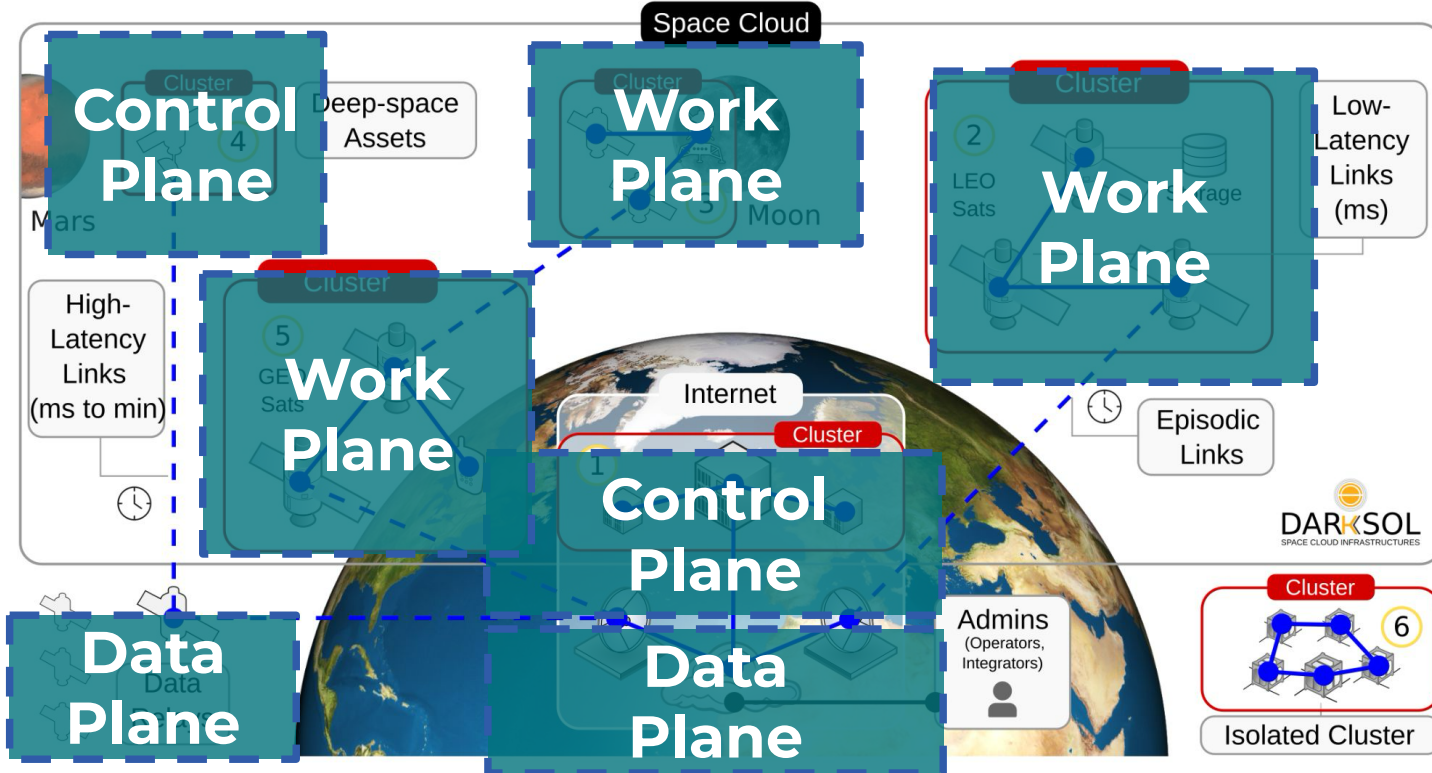


KubeCon



CloudNativeCon

India 2026





KubeCon



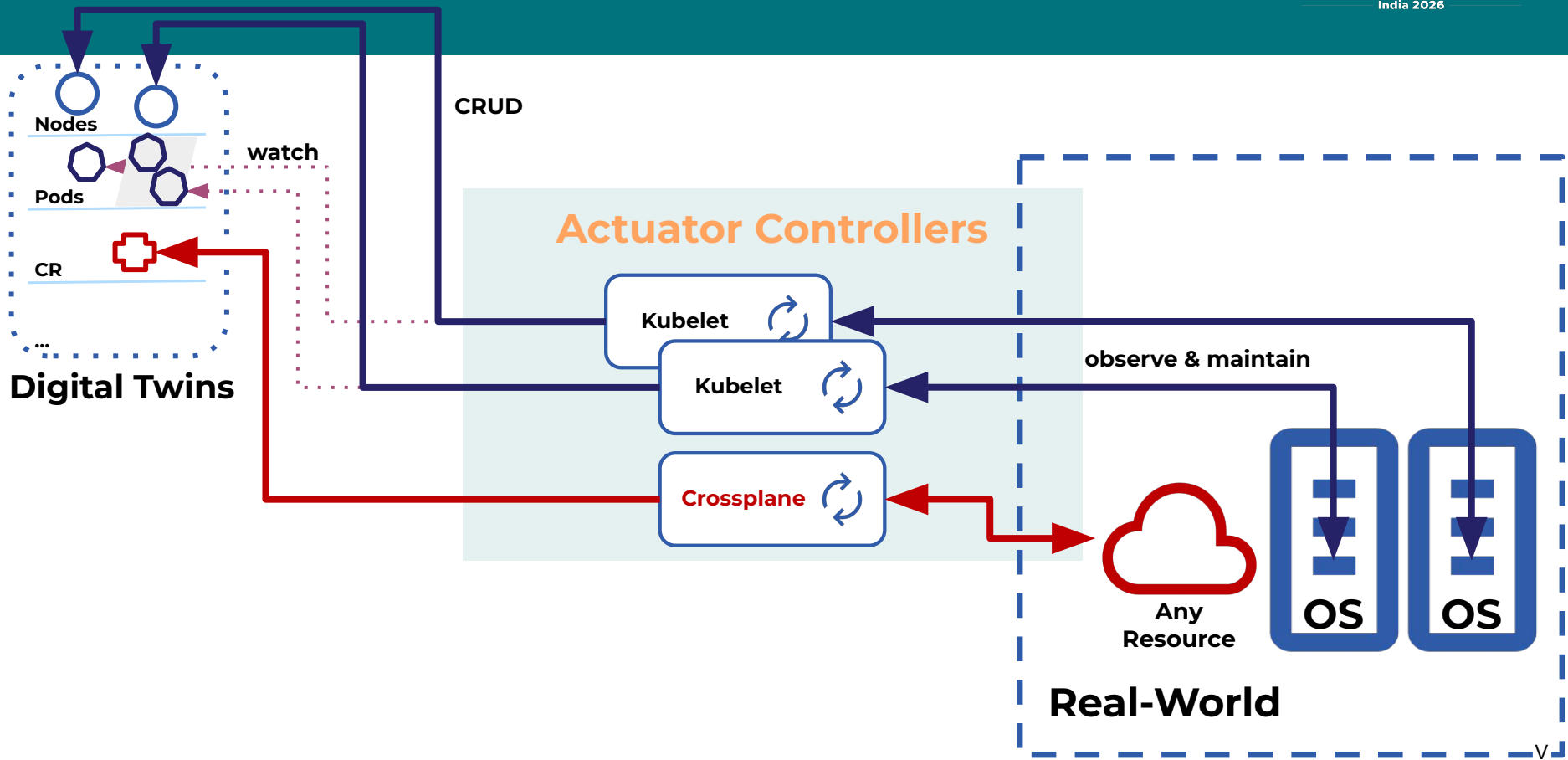
CloudNativeCon

India 2026

Controller Archetypes



Actuators



Attributes

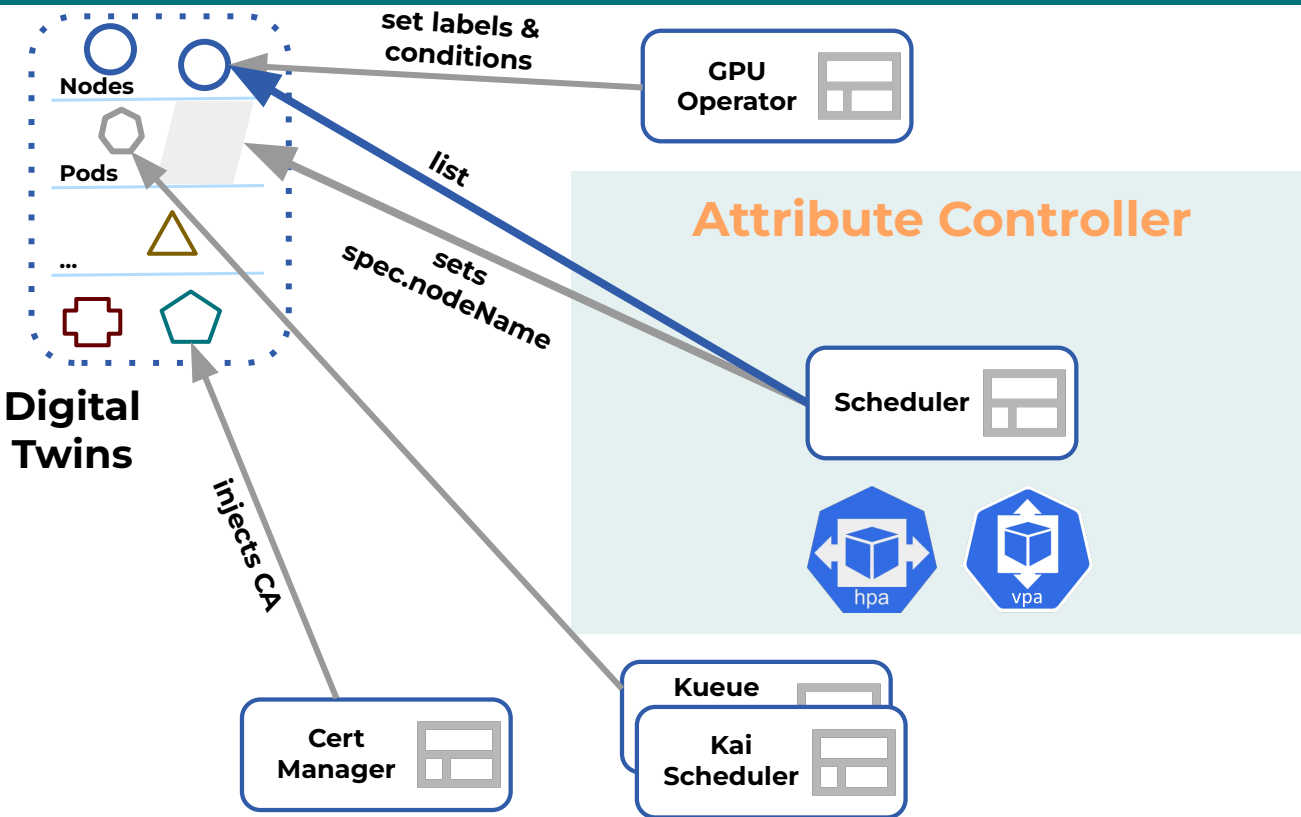


KubeCon



CloudNativeCon

India 2026



Abstraction / Logical Objects

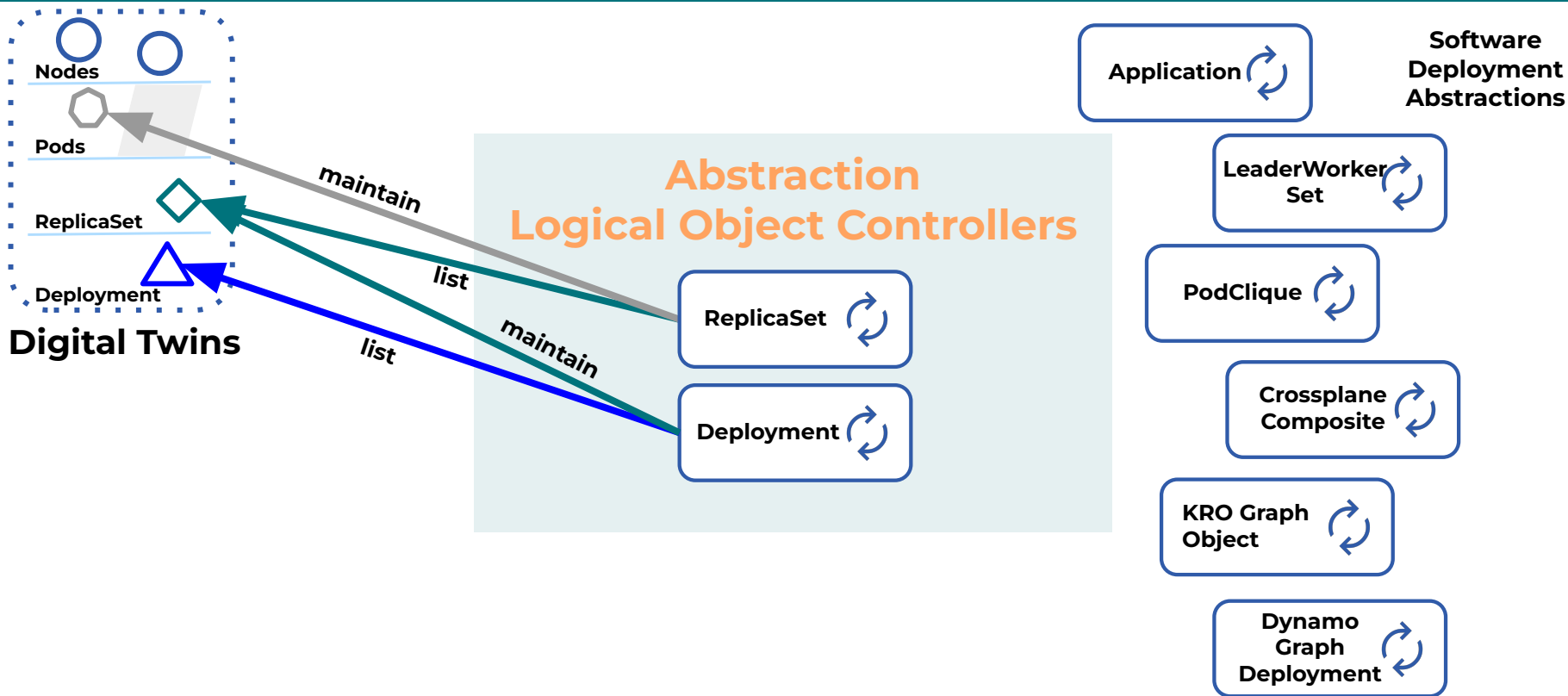


KubeCon



CloudNativeCon

India 2026



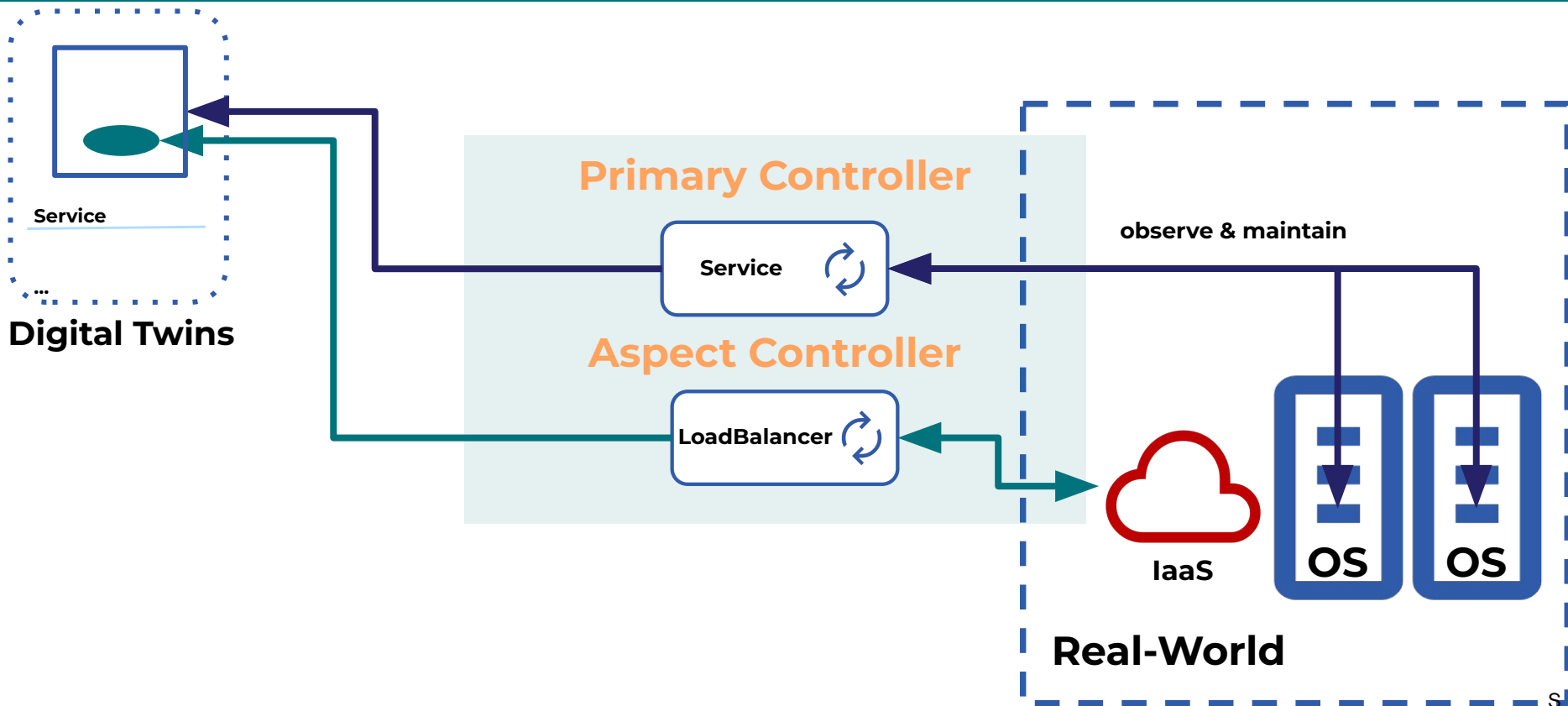
Aspects



KubeCon



CloudNativeCon
India 2026





KubeCon



CloudNativeCon

India 2026

“Between Kubernetes Clusters” with vCluster as an example



vCluster – Work Plane is another Kube

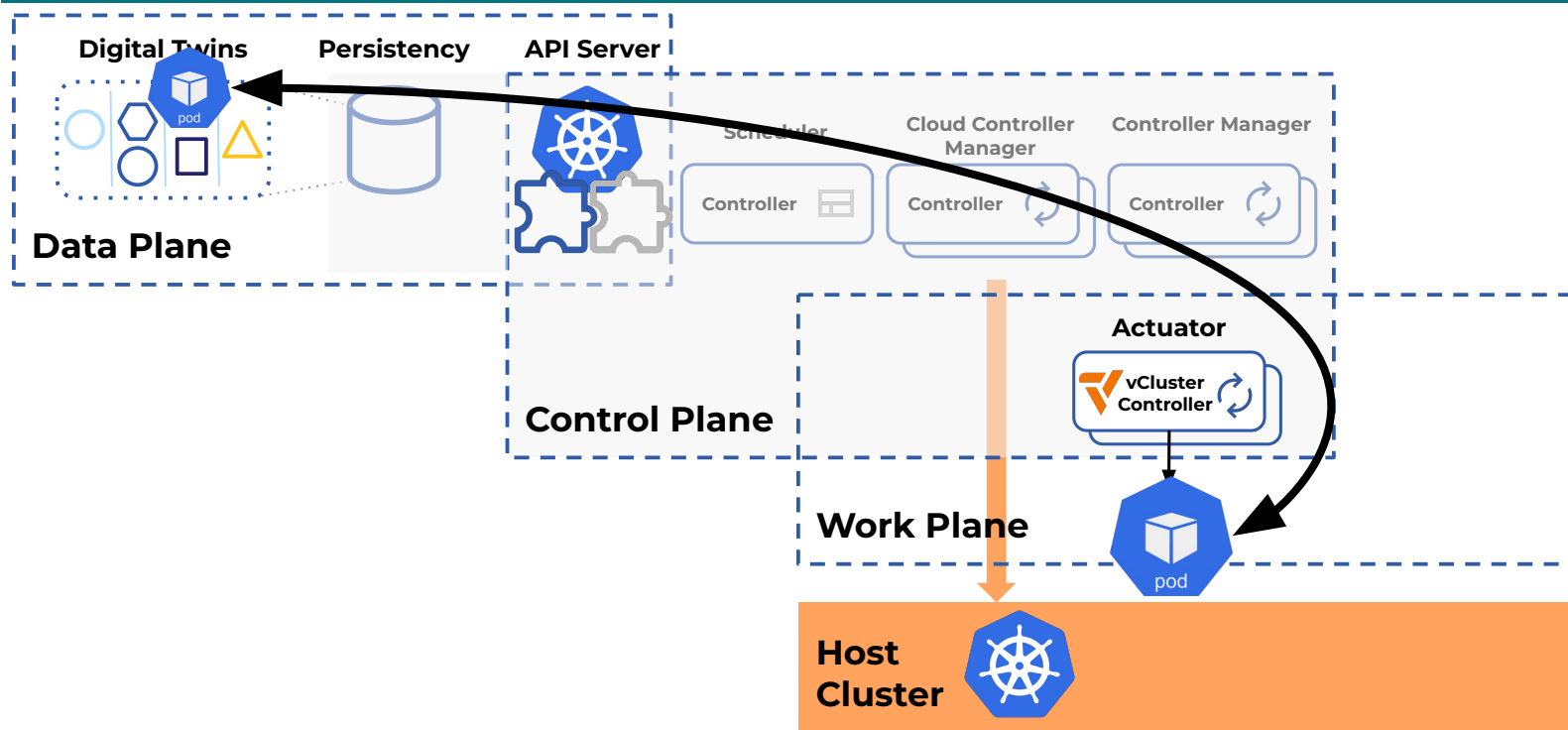


KubeCon

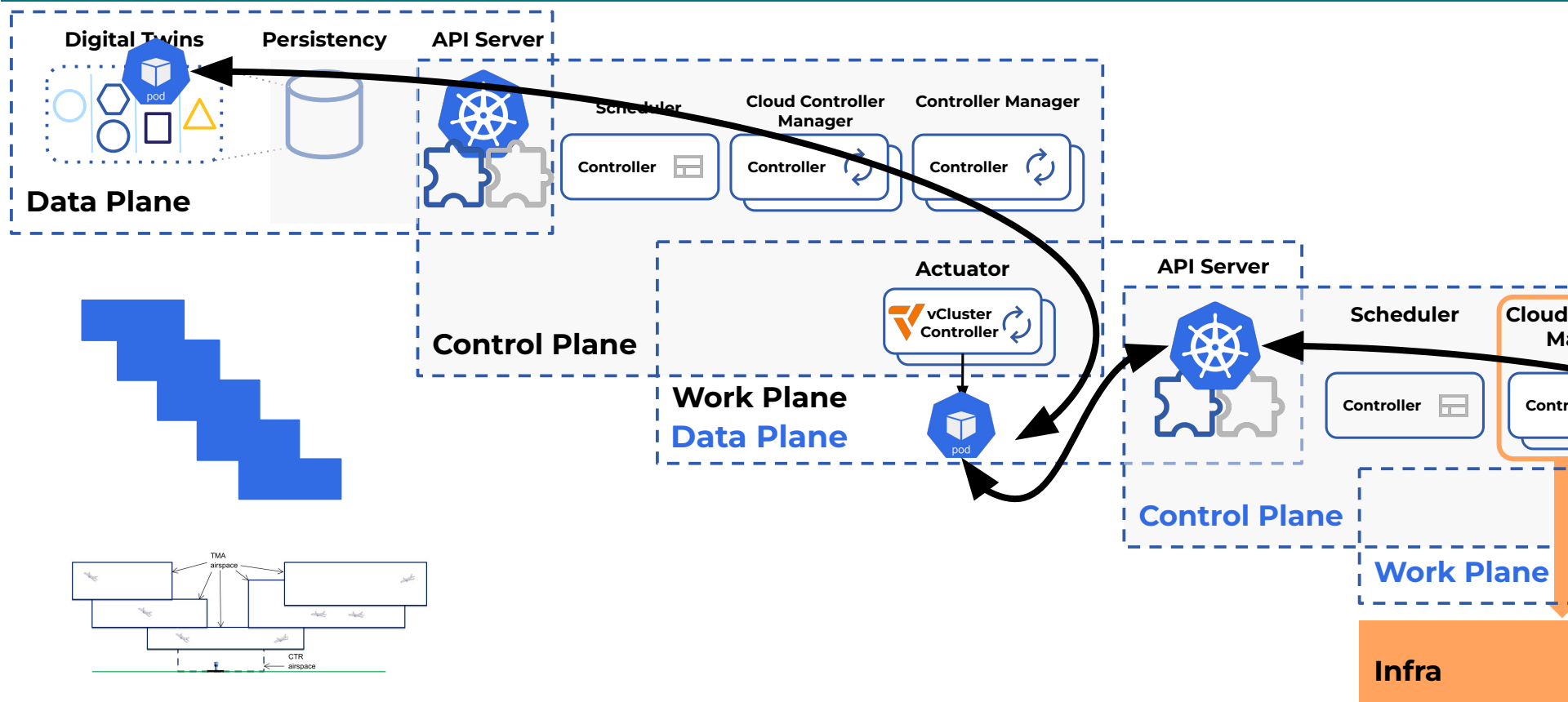


CloudNativeCon

India 2026



vCluster – Work Plane is another Kube's Data Plane



```
vCluster $ kubectl get pods -A
```

```
vCluster $ kubectl get pods -A
```

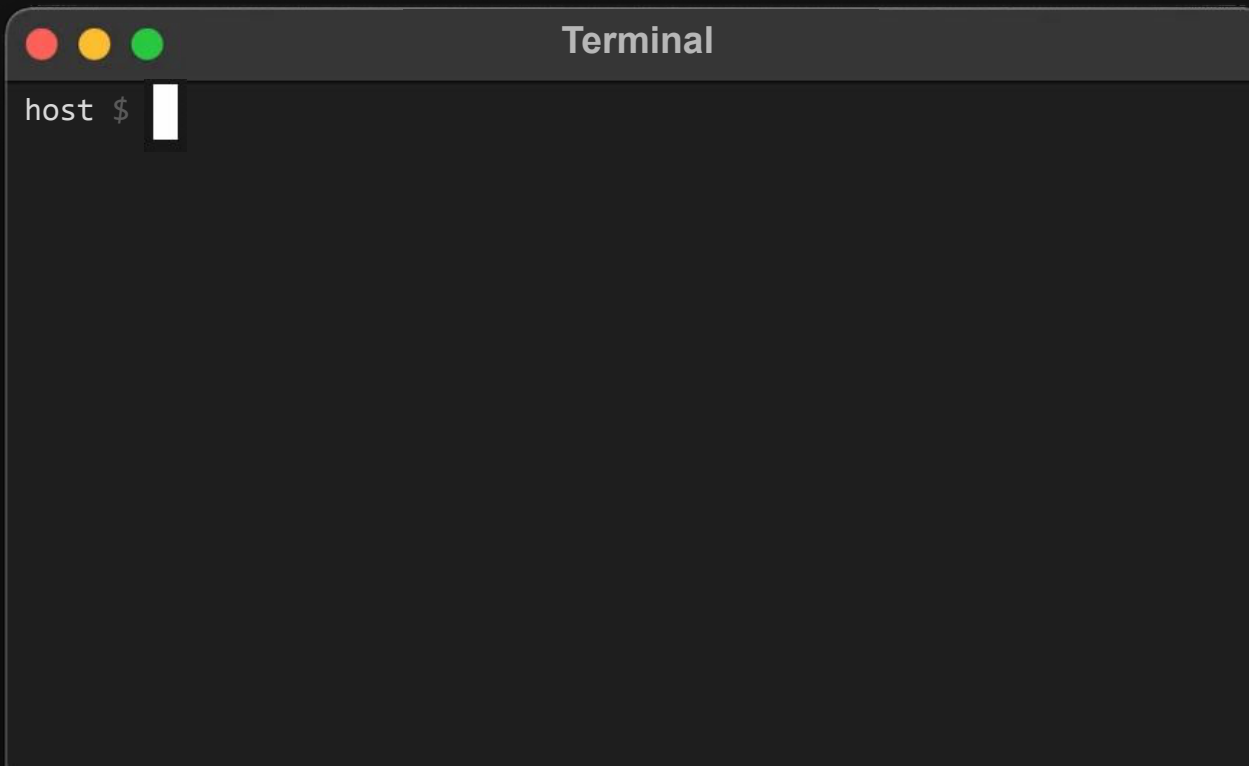
| NAMESPACE | NAME | READY | STATUS | RESTARTS | AGE |
|------------------|-------------------------------|--------------|---------------|-----------------|------------|
| cert-manager | cert-manager-48392k2j43-ka4jh | 1/1 | Running | 0 | 32d |
| default | nginx-7b86f8c875-zrjsr | 2/2 | Running | 3 (12m ago) | 8d |
| default | postgres-56876cb887-xx795 | 1/1 | Running | 0 | 52d |

```
vCluster $
```

```
vCluster $ kubectl get pods -A
```

| NAMESPACE | NAME | READY | STATUS | RESTARTS | AGE |
|------------------|-------------------------------|--------------|---------------|-----------------|------------|
| cert-manager | cert-manager-48392k2j43-ka4jh | 1/1 | Running | 0 | 32d |
| default | nginx-7b86f8c875-zrjsr | 2/2 | Running | 3 (12m ago) | 8d |
| default | postgres-56876cb887-xx795 | 1/1 | Running | 0 | 52d |

```
vCluster $
```



```
vCluster $ kubectl get pods -A
```

| NAMESPACE | NAME | READY | STATUS | RESTARTS | AGE |
|------------------|-------------------------------|--------------|---------------|-----------------|------------|
| cert-manager | cert-manager-48392k2j43-ka4jh | 1/1 | Running | 0 | 32d |
| default | nginx-7b86f8c875-zrjsr | 2/2 | Running | 3 (12m ago) | 8d |
| default | postgres-56876cb887-xx795 | 1/1 | Running | 0 | 52d |

```
vCluster $
```



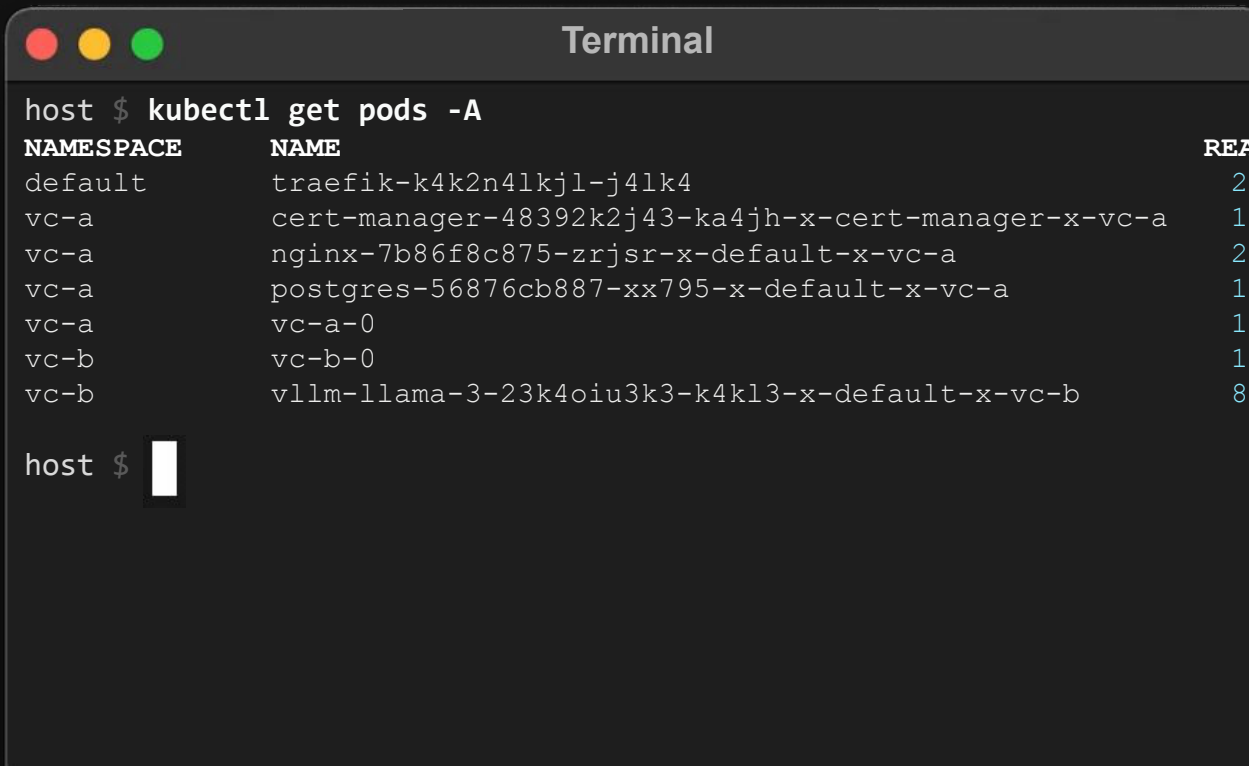
Terminal

```
host $ kubectl get pods -A
```

```
vCluster $ kubectl get pods -A
```

| NAMESPACE | NAME | READY | STATUS | RESTARTS | AGE |
|------------------|-------------------------------|--------------|---------------|-----------------|------------|
| cert-manager | cert-manager-48392k2j43-ka4jh | 1/1 | Running | 0 | 32d |
| default | nginx-7b86f8c875-zrjsr | 2/2 | Running | 3 (12m ago) | 8d |
| default | postgres-56876cb887-xx795 | 1/1 | Running | 0 | 52d |

```
vCluster $
```



A terminal window titled "Terminal" with three colored window control buttons (red, yellow, green) in the top-left corner. The terminal shows the command `host $ kubectl get pods -A` and its output. The output is a table with columns **NAMESPACE**, **NAME**, and **READY**. The rows are: `default traefik-k4k2n4lkjl-j4lk4 2`, `vc-a cert-manager-48392k2j43-ka4jh-x-cert-manager-x-vc-a 1`, `vc-a nginx-7b86f8c875-zrjsr-x-default-x-vc-a 2`, `vc-a postgres-56876cb887-xx795-x-default-x-vc-a 1`, `vc-a vc-a-0 1`, `vc-b vc-b-0 1`, and `vc-b vllm-llama-3-23k4oiu3k3-k4kl3-x-default-x-vc-b 8`. Below the output, the prompt `host $` is followed by a white cursor bar.

```
Terminal  
host $ kubectl get pods -A  
NAMESPACE      NAME                                                    READY  
default         traefik-k4k2n4lkjl-j4lk4                               2  
vc-a            cert-manager-48392k2j43-ka4jh-x-cert-manager-x-vc-a   1  
vc-a            nginx-7b86f8c875-zrjsr-x-default-x-vc-a               2  
vc-a            postgres-56876cb887-xx795-x-default-x-vc-a             1  
vc-a            vc-a-0                                                  1  
vc-b            vc-b-0                                                  1  
vc-b            vllm-llama-3-23k4oiu3k3-k4kl3-x-default-x-vc-b       8  
  
host $
```

```
vCluster $ kubectl get pods -A
```

| NAMESPACE | NAME | READY | STATUS | RESTARTS | AGE |
|------------------|-------------------------------|--------------|---------------|-----------------|------------|
| cert-manager | cert-manager-48392k2j43-ka4jh | 1/1 | Running | 0 | 32d |
| default | nginx-7b86f8c875-zrjsr | 2/2 | Running | 3 (12m ago) | 8d |
| default | postgres-56876cb887-xx795 | 1/1 | Running | 0 | 52d |

```
vCluster $
```

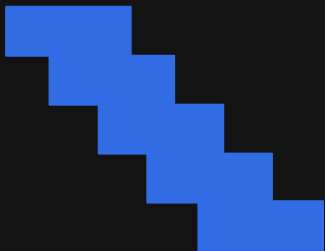


Terminal

```
host $ kubectl get pods -A
```

| NAMESPACE | NAME | READY |
|------------------|---|--------------|
| default | traefik-k4k2n4lkjl-j4lk4 | 2 |
| vc-a | cert-manager-48392k2j43-ka4jh-x-cert-manager-x-vc-a | 1 |
| vc-a | nginx-7b86f8c875-zrjsr-x-default-x-vc-a | 2 |
| vc-a | postgres-56876cb887-xx795-x-default-x-vc-a | 1 |
| vc-a | vc-a-0 | 1 |
| vc-b | vc-b-0 | 1 |
| vc-b | vllm-llama-3-23k4oiu3k3-k4kl3-x-default-x-vc-b | 8 |

```
host $
```





KubeCon



CloudNativeCon

India 2026

Multi-Cluster Federation



Naive / Direct Implementation



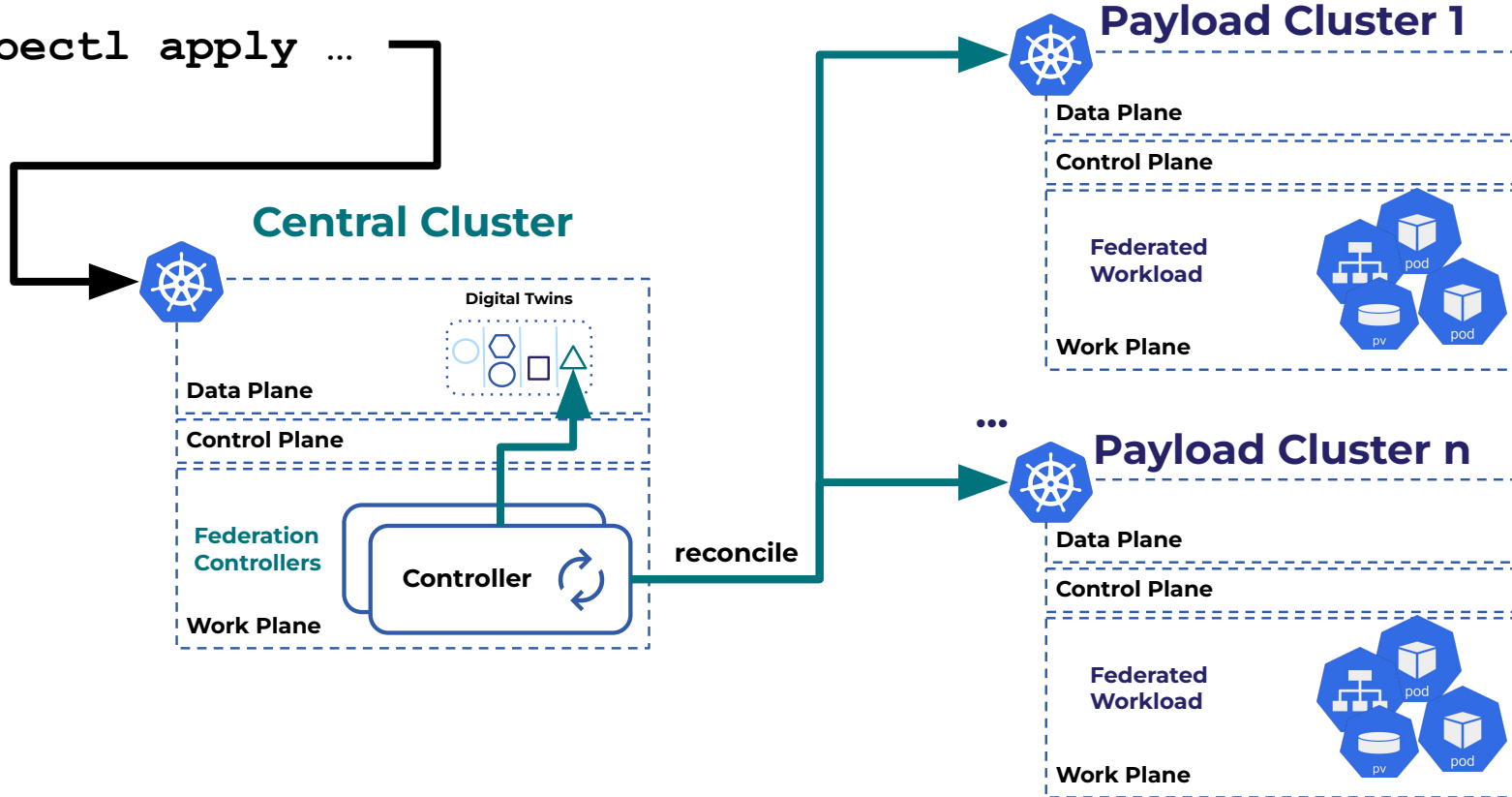
KubeCon



CloudNativeCon

India 2026

kubectl apply ...





Remember the Kubelet Pattern, Anyone?

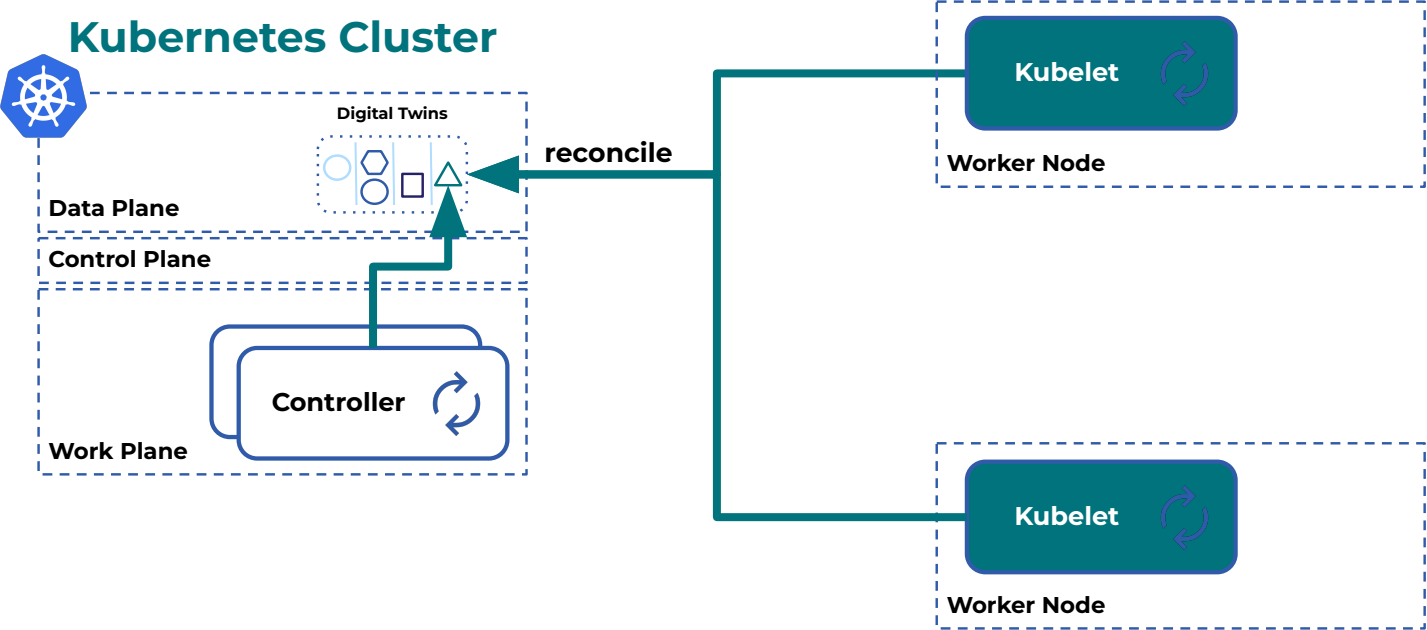


KubeCon

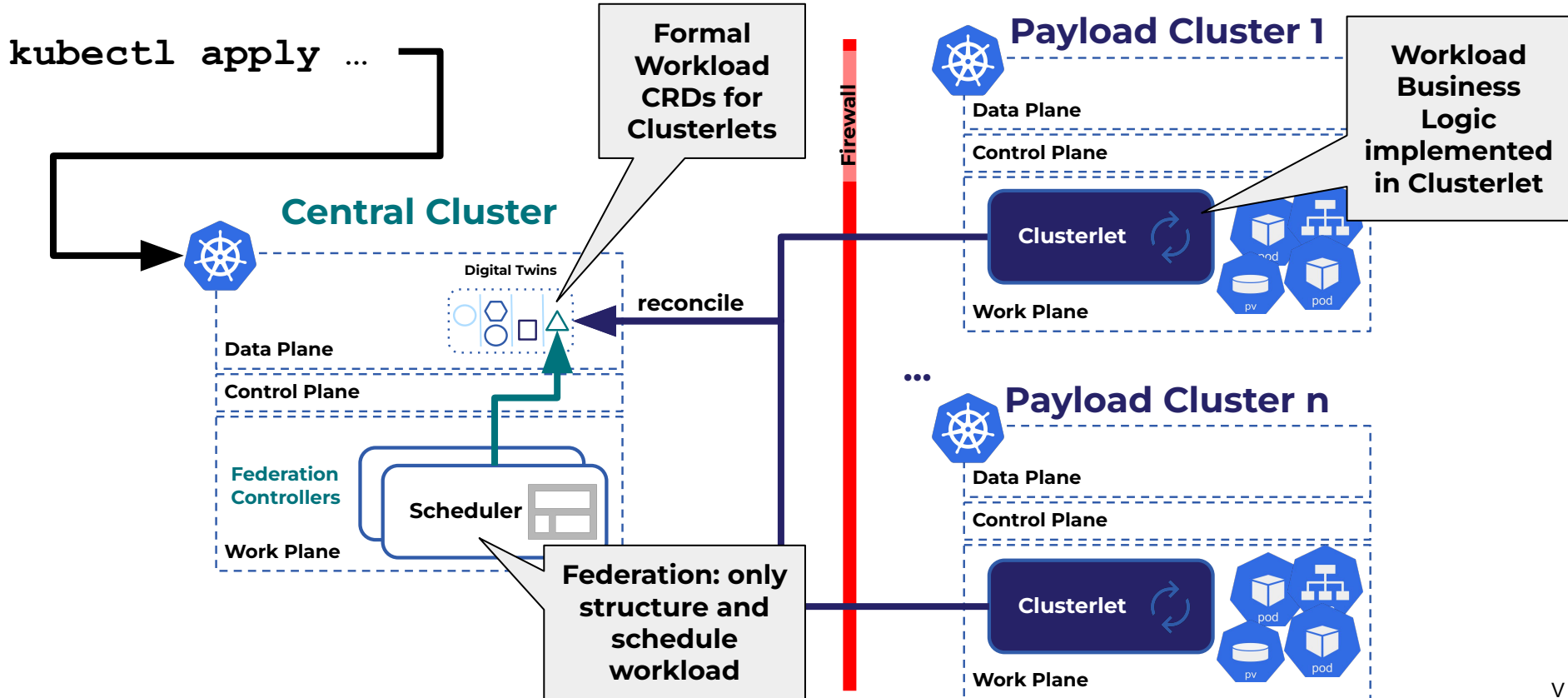


CloudNativeCon

India 2026



Cloud Native Implementation



Hosted Control Planes (HCP)

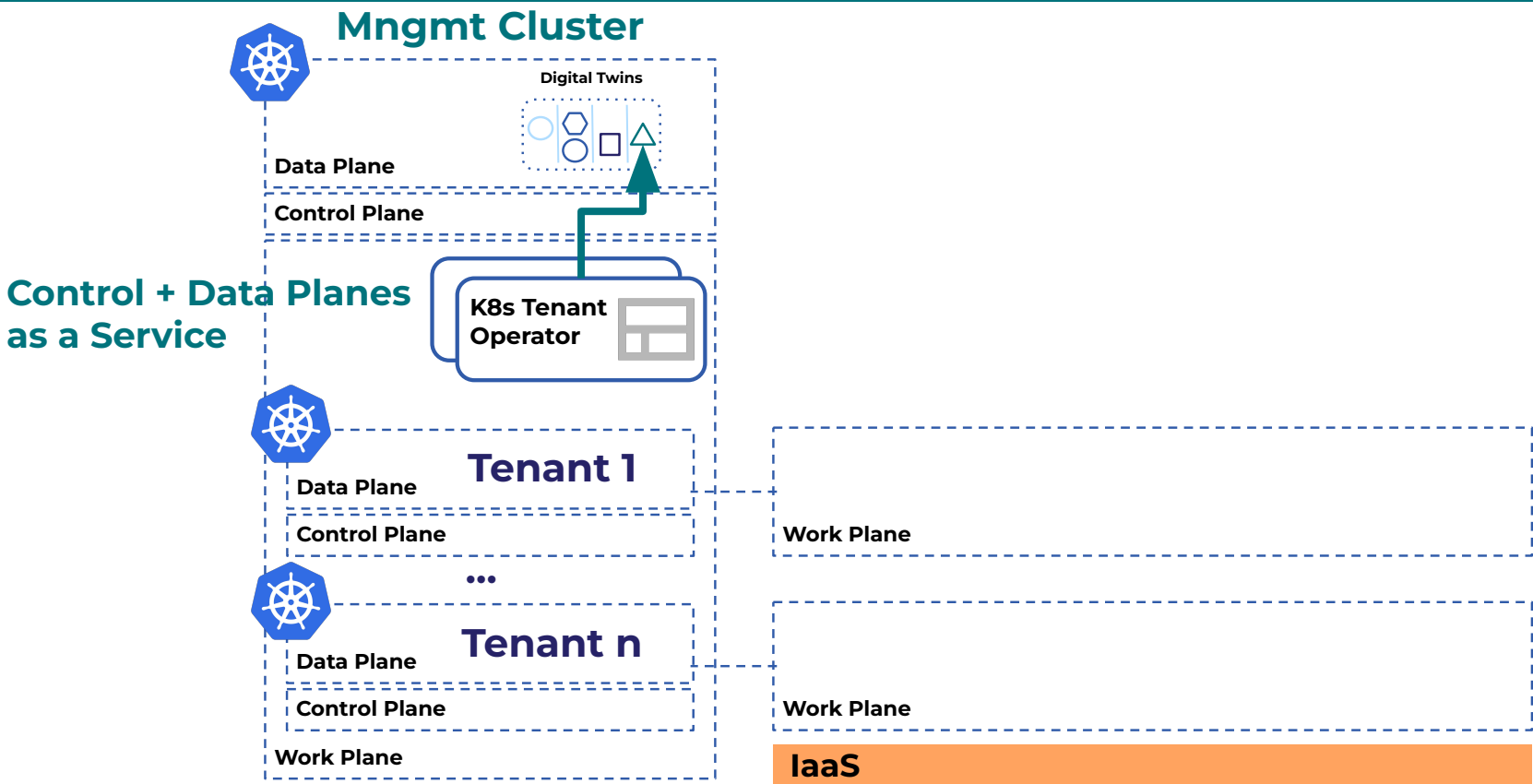


KubeCon



CloudNativeCon

India 2026



Hosted Control Planes (HCP): NeoNephos Gardener for large scale

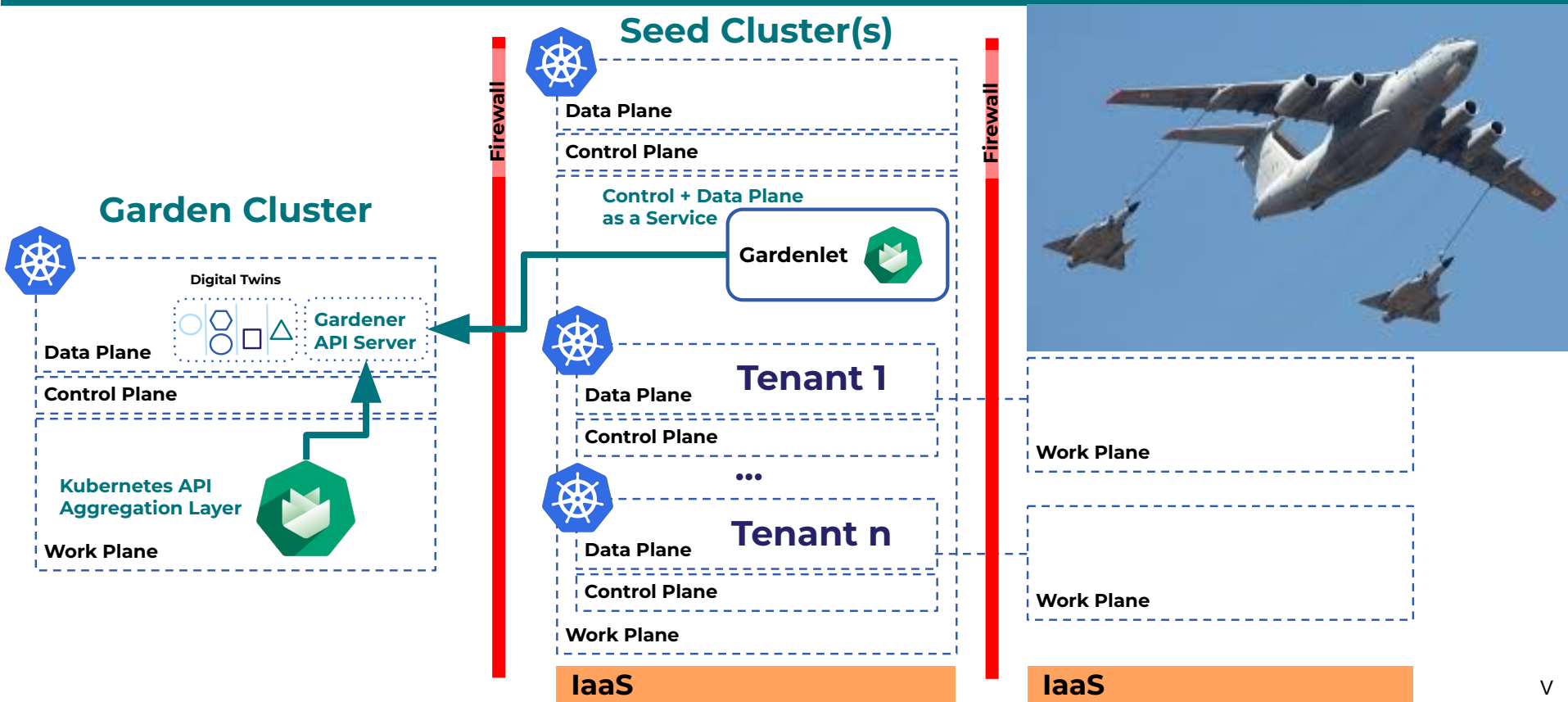


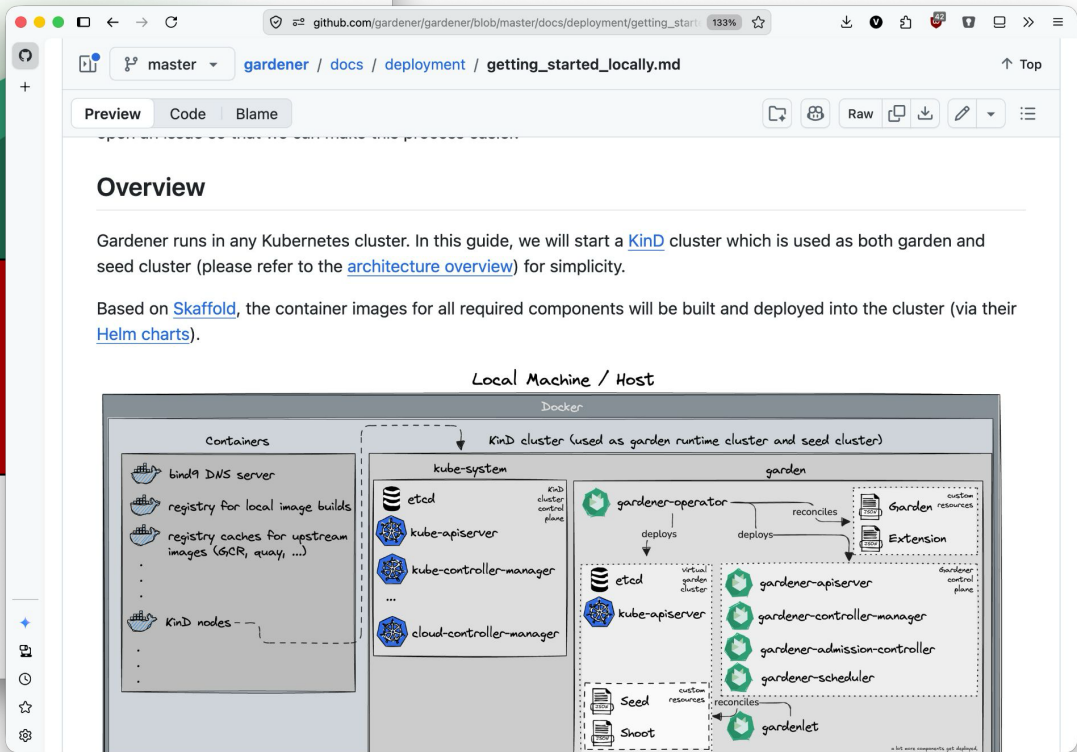
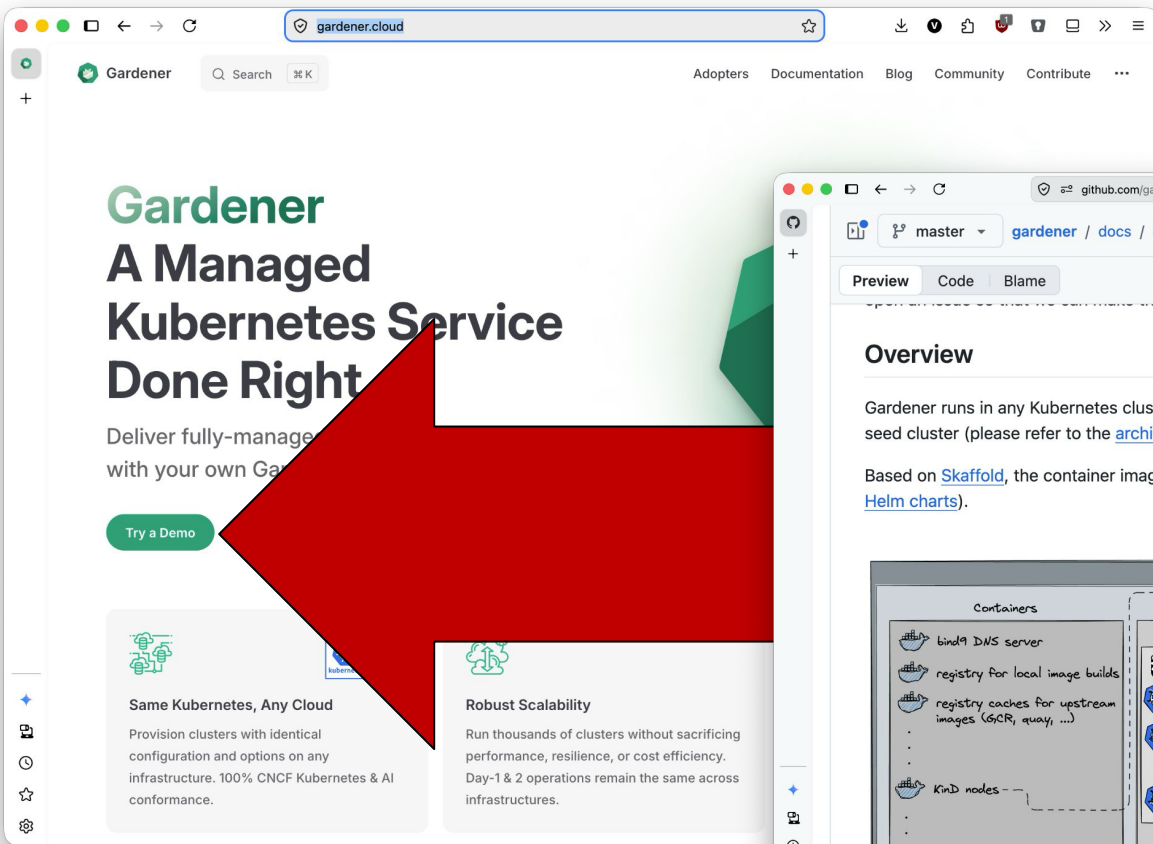
KubeCon



CloudNativeCon

India 2026





<https://gardener.cloud/>
https://github.com/gardener/gardener/blob/master/docs/deployment/getting_started_locally.md

```
$ less example/provider-local/shoot.yaml
apiVersion: core.gardener.cloud/v1beta1
kind: Shoot
metadata:
  name: local
  namespace: garden-local
  annotations:
    shoot.gardener.cloud/cloud-config-execution-max-delay-seconds: "0"
    authentication.gardener.cloud/issuer: "managed"
spec:
  cloudProfile:
    name: local
  secretBindingName: local # dummy, doesn't contain any credentials
  region: local
  networking:
    type: calico
    nodes: 10.10.0.0/16
  provider:
    type: local
  workers:
  - name: local
    machine:
      type: local
    cri:
      name: containerd
      minimum: 1
```

Garden
Cluster

```
$ kubectl apply -f example/provider-local/shoot.yaml
shoot.core.gardener.cloud/local created
```

```
$ kubectl get shoot local --watch # specify `-o yaml` for detailed output
```

| NAME | CLOUDPROFILE | PROVIDER | REGION | K8S VERSION | HIBERNATION | LAST OPERATION | STATUS |
|-------|--------------|----------|--------|-------------|-------------|--------------------------|---------|
| local | local | local | local | 1.32.0 | Awake | Create Processing (13%) | healthy |
| 100s | | | | | | | |
| ... | | | | | | | |
| local | local | local | local | 1.32.0 | Awake | Create Processing (96%) | healthy |
| 9m32s | | | | | | | |
| local | local | local | local | 1.32.0 | Awake | Create Processing (97%) | healthy |
| 9m32s | | | | | | | |
| local | local | local | local | 1.32.0 | Awake | Create Processing (98%) | healthy |
| 9m32s | | | | | | | |
| local | local | local | local | 1.32.0 | Awake | Create Processing (100%) | healthy |
| 9m32s | | | | | | | |

```
$ █
```

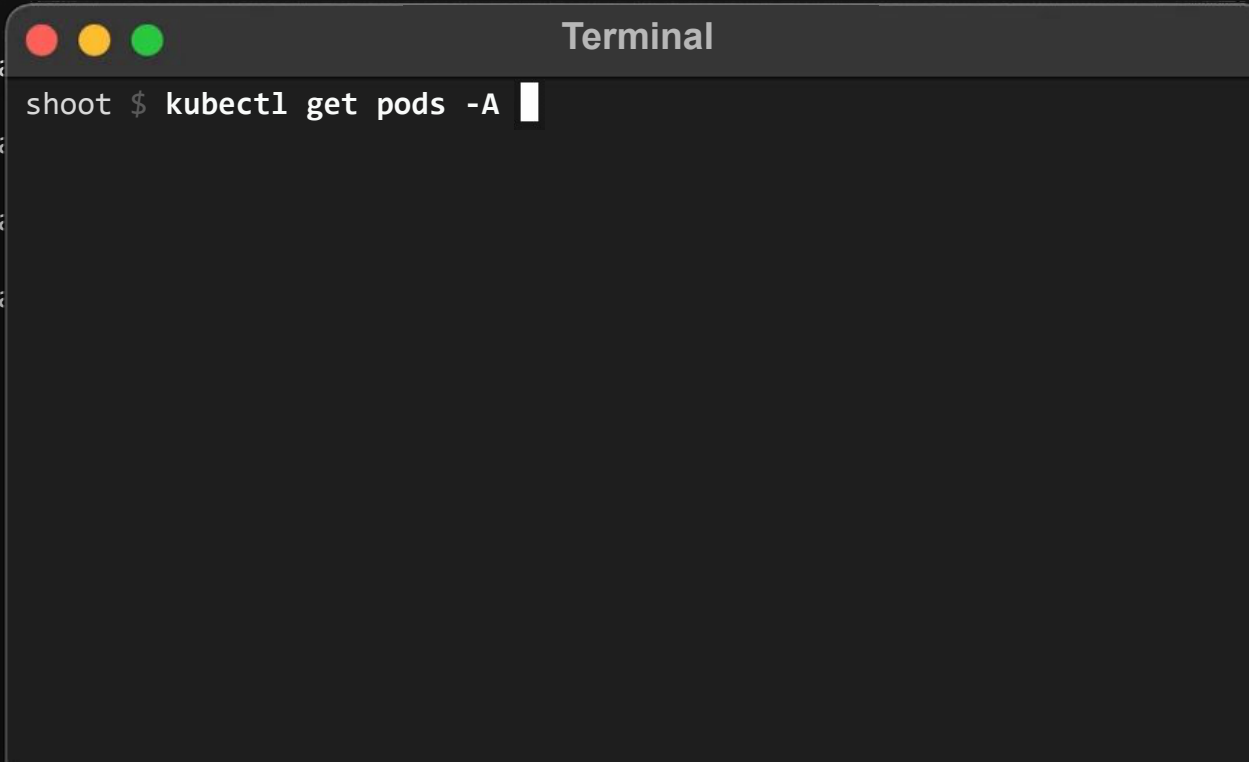
Garden
Cluster

```
$ kubectl apply -f example/provider-local/shoot.yaml
shoot.core.gardener.cloud/local created
```

```
$ kubectl get shoot local --watch # specify `-o yaml` for detailed output
```

| NAME | CLOUDPROFILE | PROVIDER | REGION | K8S VERSION | HIBERNATION | LAST OPERATION | STATUS |
|-------|--------------|----------|--------|-------------|-------------|-------------------------|---------|
| local | local | local | local | 1.32.0 | Awake | Create Processing (13%) | healthy |
| ... | | | | | | | |
| local | local | local | local | | | | |
| 9m32s | | | | | | | |
| local | local | local | local | | | | |
| 9m32s | | | | | | | |
| local | local | local | local | | | | |
| 9m32s | | | | | | | |
| local | local | local | local | | | | |
| 9m32s | | | | | | | |

```
$
```



```
$ kubectl apply -f example/provider-local/shoot.yaml
shoot.core.gardener.cloud/local created
```

```
$ kubectl get shoot local --watch # specify `-o yaml` for detailed output
```

| NAME | CLOUDPROFILE | PROVIDER | REGION | K8S VERSION | HIBERNATION | LAST OPERATION | STATUS |
|-------|--------------|----------|--------|-------------|-------------|-------------------------|---------|
| local | local | local | local | 1.32.0 | Awake | Create Processing (13%) | healthy |

100s

...

```
local local local local
```

9m32s

```
local local local local
```

9m32s

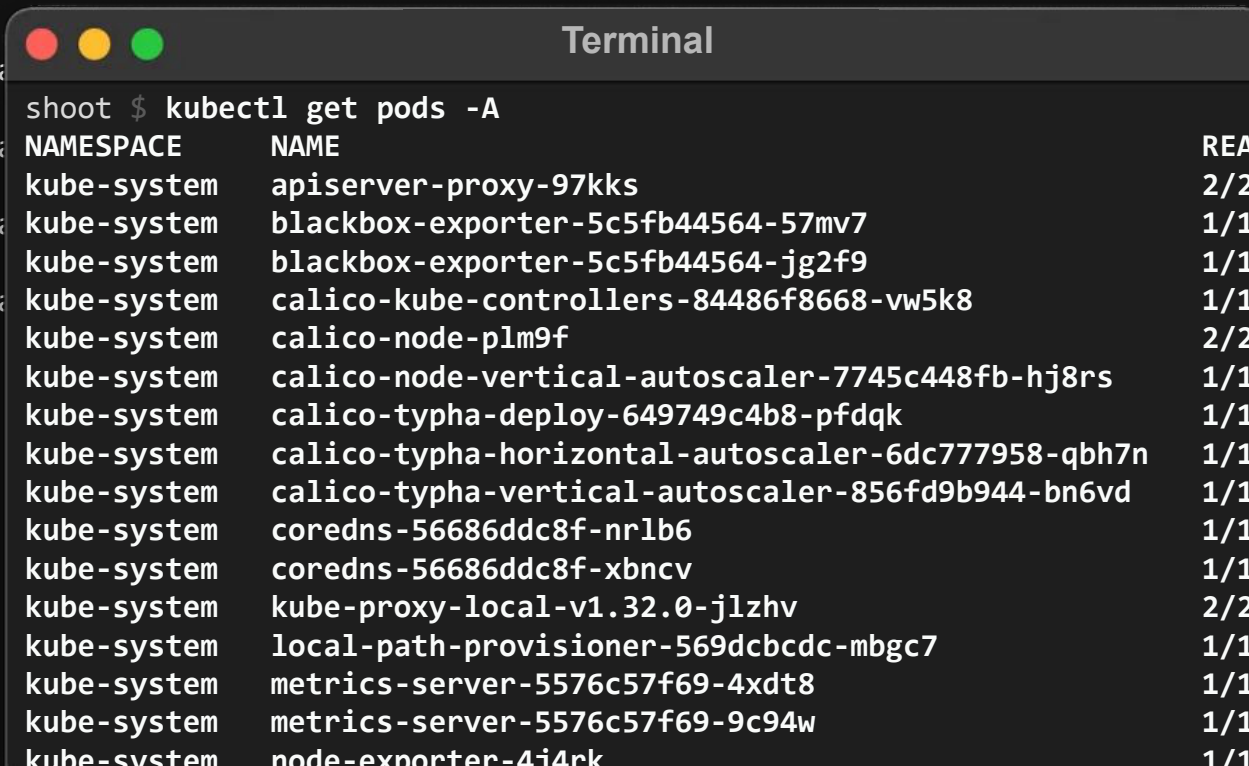
```
local local local local
```

9m32s

```
local local local local
```

9m32s

```
$
```



Terminal

```
shoot $ kubectl get pods -A
NAMESPACE      NAME
kube-system    apiserver-proxy-97kks
kube-system    blackbox-exporter-5c5fb44564-57mv7
kube-system    blackbox-exporter-5c5fb44564-jg2f9
kube-system    calico-kube-controllers-84486f8668-vw5k8
kube-system    calico-node-plm9f
kube-system    calico-node-vertical-autoscaler-7745c448fb-hj8rs
kube-system    calico-typha-deploy-649749c4b8-pfdqk
kube-system    calico-typha-horizontal-autoscaler-6dc777958-qbh7n
kube-system    calico-typha-vertical-autoscaler-856fd9b944-bn6vd
kube-system    coredns-56686ddc8f-nrlb6
kube-system    coredns-56686ddc8f-xbncv
kube-system    kube-proxy-local-v1.32.0-jlzhv
kube-system    local-path-provisioner-569dcbcdc-mbgc7
kube-system    metrics-server-5576c57f69-4xdt8
kube-system    metrics-server-5576c57f69-9c94w
kube-system    node-exporter-4i4rk
```


demo.gardener.cloud/terminal
🏠 ⭐ 📄 🗑️ 🔄 📱 📺 📶 ☰

Gardener Demo

```

-- 0 garden - bash
[~/src/github.com/gardener/gardener (*|garden:garden-local)] ((v1.120.1))
$ export KUBECONFIG=~/.kube/garden.config
[~/src/github.com/gardener/gardener (*|garden:garden-local)] ((v1.120.1))
$ k get seeds
NAME      STATUS  LAST OPERATION      PROVIDER  REGION  AGE  VERSION  K8S VERSION
local    Ready  Reconcile Succeeded (100%)  local    local   17h  v1.120.1  v1.32.0
managedseed  Ready  Reconcile Succeeded (100%)  local    local   17h  v1.120.1  v1.32.0
[~/src/github.com/gardener/gardener (*|garden:garden-local)] ((v1.120.1))
$

-- 1 runtime#seeds - bash
142 shoot--garden--managedseed  1/1  Running  0  5h14m
144 shoot--garden--managedseed  1/1  Running  0  4h49m
144 shoot--garden--managedseed  1/1  Running  0  135m
145 shoot--garden--managedseed  2/2  Running  0  17h
146 shoot--garden--managedseed  2/2  Running  0  17h
147 virtual-garden-istio-ingress  1/1  Running  0  17h
148 virtual-garden-istio-ingress  1/1  Running  0  17h
149 virtual-garden-istio-ingress  1/1  Running  0  17h
150 virtual-garden-istio-ingress  1/1  Running  0  17h
151 virtual-garden-istio-ingress  1/1  Running  0  17h
152 virtual-garden-istio-ingress  1/1  Running  0  17h
[~/src/github.com/gardener/gardener (*|runtime:garden)] ((v1.120.1))
$

-- 2 managed seed - bash
49 kube-system                vpn-shoot-1                Running  0  5h20m
50 shoot--local--local-ms     blackbox-exporter          Running  0  5h13m
51 shoot--local--local-ms     blackbox-exporter          Running  0  4h40m
52 shoot--local--local-ms     cluster-autoscaler         Running  0  12h
53 shoot--local--local-ms     etcd-events-0              Running  0  17h
54 shoot--local--local-ms     etcd-main-0                Running  0  17h
55 shoot--local--local-ms     gardener-resource-manager-5c785566 Running  0  6h15m
56 shoot--local--local-ms     gardener-resource-manager-5c785566 Running  0  6h16m
57 shoot--local--local-ms     kube-apiserver-7dc5ff7c85-hx48n Running  0  7h53m
58 shoot--local--local-ms     kube-apiserver-7dc5ff7c85-mfgrb Running  0  7h54m
59 shoot--local--local-ms     kube-controller-manager-5868b49d9 Running  0  12h
60 shoot--local--local-ms     kube-scheduler-56556db5c8-vckpz Running  0  12h
61 shoot--local--local-ms     kube-state-metrics-65bb79cdb7-fmf6 Running  0  7h51m
62 shoot--local--local-ms     machine-controller-manager-857dd8 Running  0  12h

-- 3 shoot - bash
8 kube-system    calico-typha-deploy-649749c4b8-zp9vq  1/1  Running  0
9 kube-system    calico-typha-horizontal-autoscaler-6dc777958-dzn2q  1/1  Running  0
10 kube-system   calico-typha-vertical-autoscaler-856fd9b944-98hms  1/1  Running  0
11 kube-system   coredns-685fd8ddb-9n9qf  1/1  Running  0
12 kube-system   coredns-685fd8ddb-mmjhl  1/1  Running  0
13 kube-system   kube-proxy-local-v1.32.0-v16xz  2/2  Running  0
14 kube-system   local-path-provisioner-569dcbcdc-6mftb  1/1  Running  2 (17h ago) 5h12m
15 kube-system   metrics-server-5f69f7b9c6-dmmpw  1/1  Running  0 10h
16 kube-system   metrics-server-5f69f7b9c6-lrqpt  1/1  Running  0 10h
17 kube-system   node-exporter-h6vfw  1/1  Running  0 5h13m
18 kube-system   node-problem-detector-jrrpc  1/1  Running  0 92m
19 kube-system   vpn-shoot-687b4d7dc5-6wwzq  1/1  Running  0 16h

[~/src/github.com/gardener/gardener (*|garden:N/A)] ((v1.120.1))
$
gardener 0:demo 1:garden 2:seeds 3:shoots 4:runtime#seed pods 5:managed seed pods 6:layers* 7:gardenctl 8:k9s

```

Tip

Need more terminals? Check out our [TMUX GUIDE](#) below.

1/21: What is Gardener?

Gardener is an open-source project contributed by SAP and others. It provides a Kubernetes-based platform to manage fleets of Kubernetes clusters across multiple cloud providers in a uniform way. This means you can manage and operate clusters on Azure, AWS, GCP, and more using the same set of APIs and concepts.

<https://gardener.cloud>

Focus on the Most Important: What is a Shoot and How Can It Be Created?

A "Shoot" in Gardener terminology is a Kubernetes cluster that is managed by Gardener. It's the primary resource that Gardener introduces on top of standard Kubernetes resources. The `Shoot` resource contains the specification for the desired state of the Kubernetes cluster. This includes the Kubernetes version, infrastructure type, and machine types. You can find more details about the `Shoot` resource in the [Gardener documentation](#).

Let's Dive In: Creating Your First Shoot

TMUX GUIDE
PREVIOUS
NEXT

DISCONNECT Updated 14 minutes ago Terminates in 4 hours 8bkz



KubeCon



CloudNativeCon

India 2026

Retrospective on Clusterlet



An Architectural Observation!

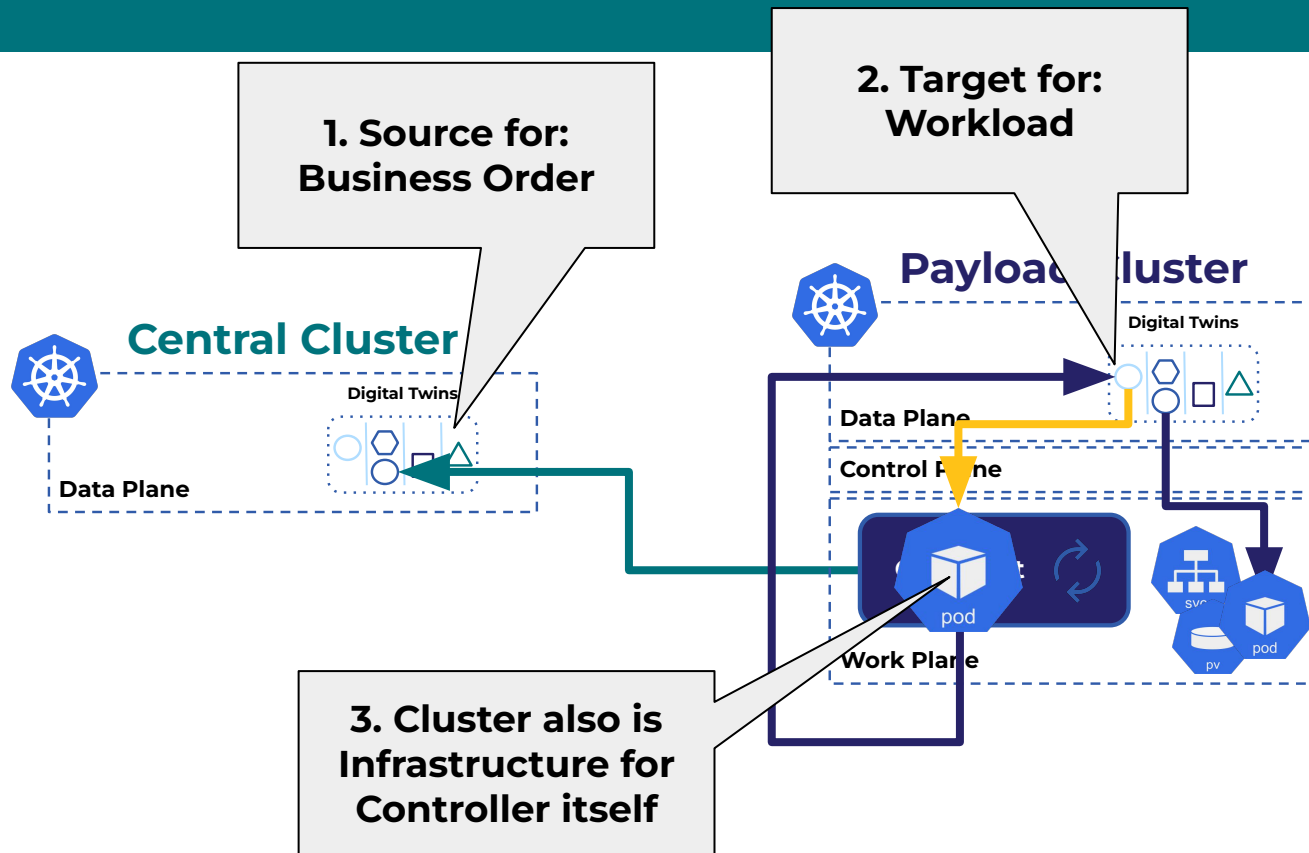


KubeCon



CloudNativeCon

India 2026



Generalization of our Observation

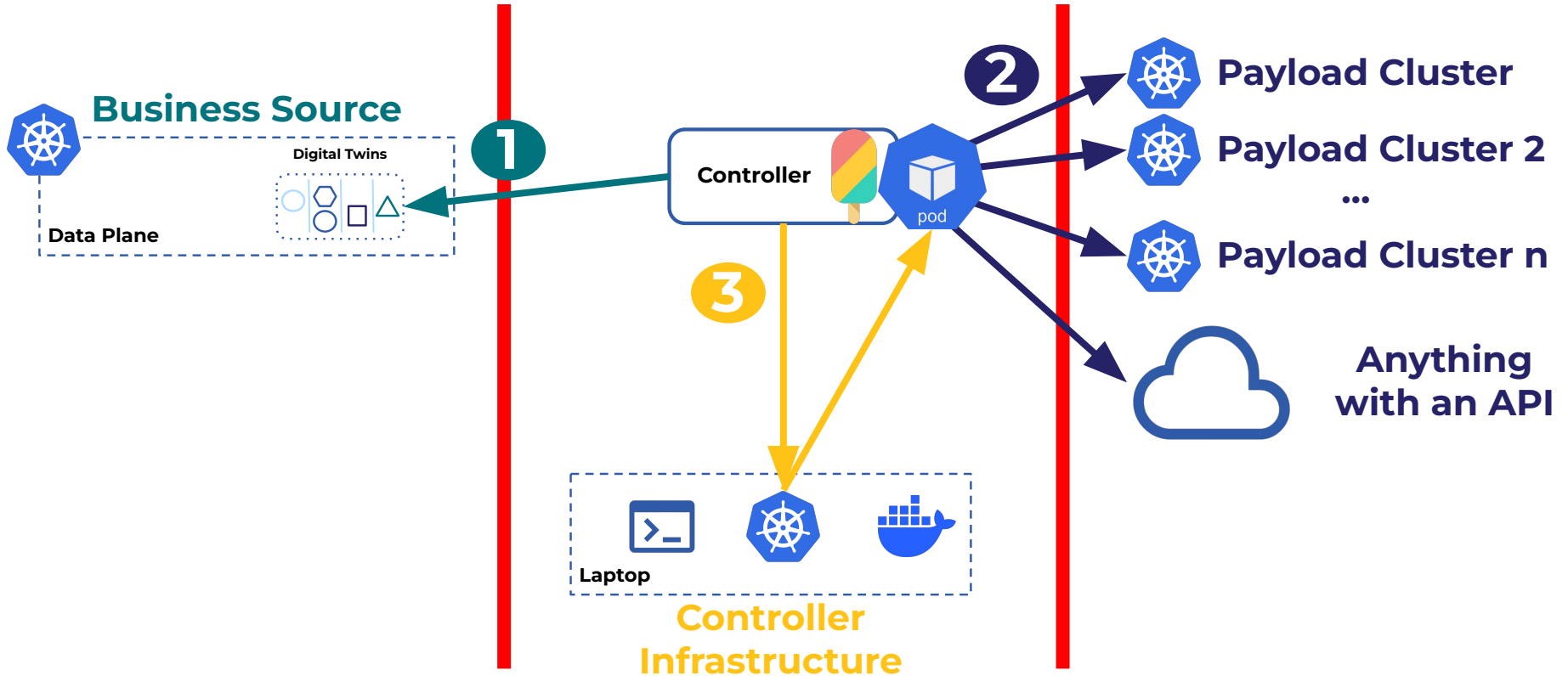


KubeCon



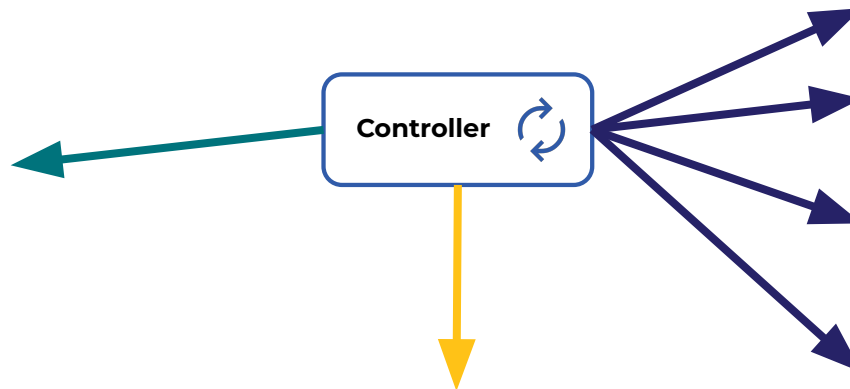
CloudNativeCon

India 2026





Multi-Cluster-Aware Reconcilers



Run reconcilers that listen to some cluster(s) and operate other clusters





KubeCon



CloudNativeCon

India 2026

Breaking the Architectural Sound Barrier



Separation of Concerns



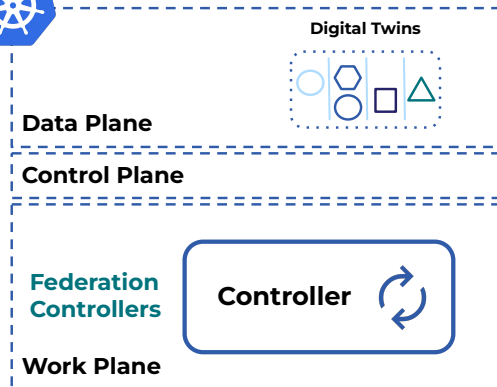
KubeCon



CloudNativeCon

India 2026

Central Cluster



Controller Cluster



Separation of Concerns

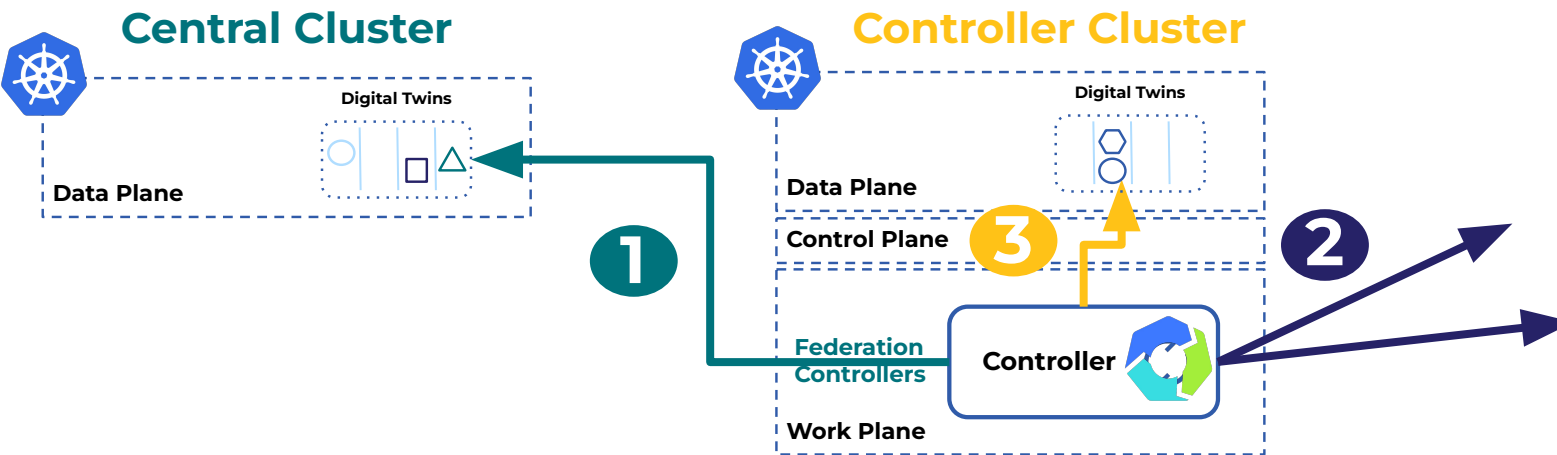


KubeCon

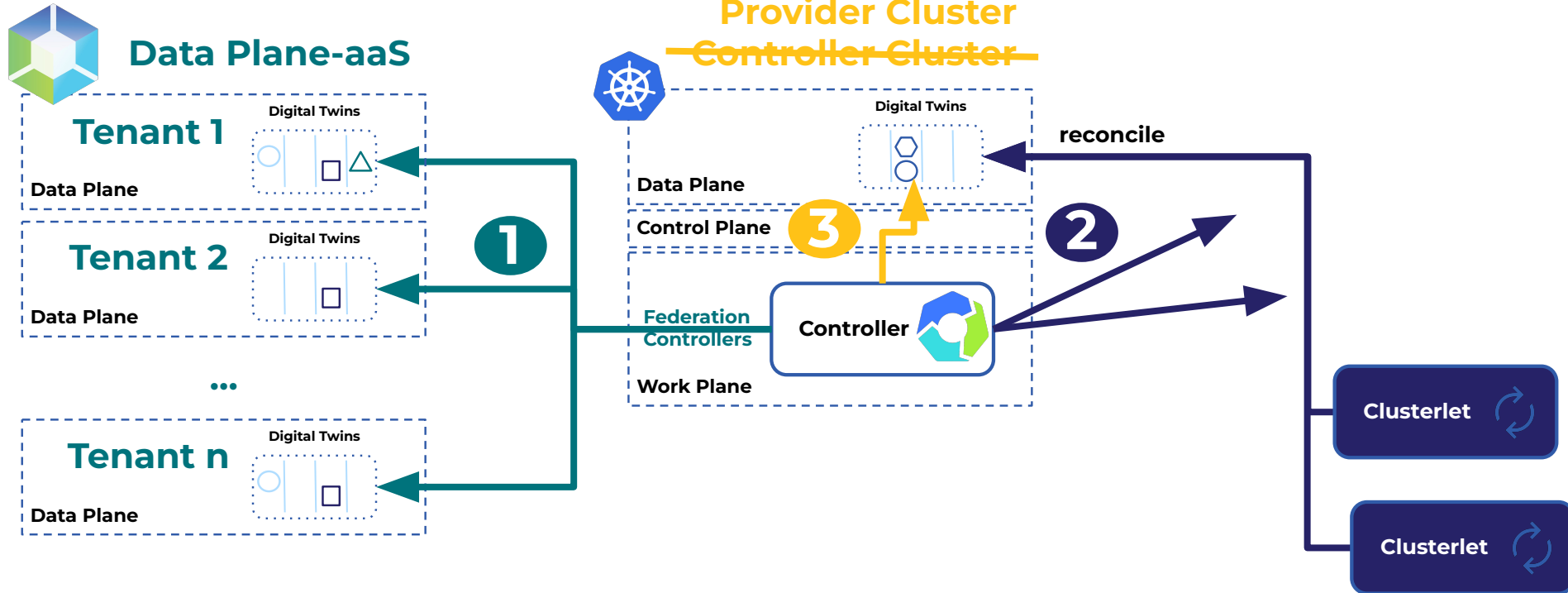


CloudNativeCon

India 2026



Pure Data Planes with kcp



Platform Engineering

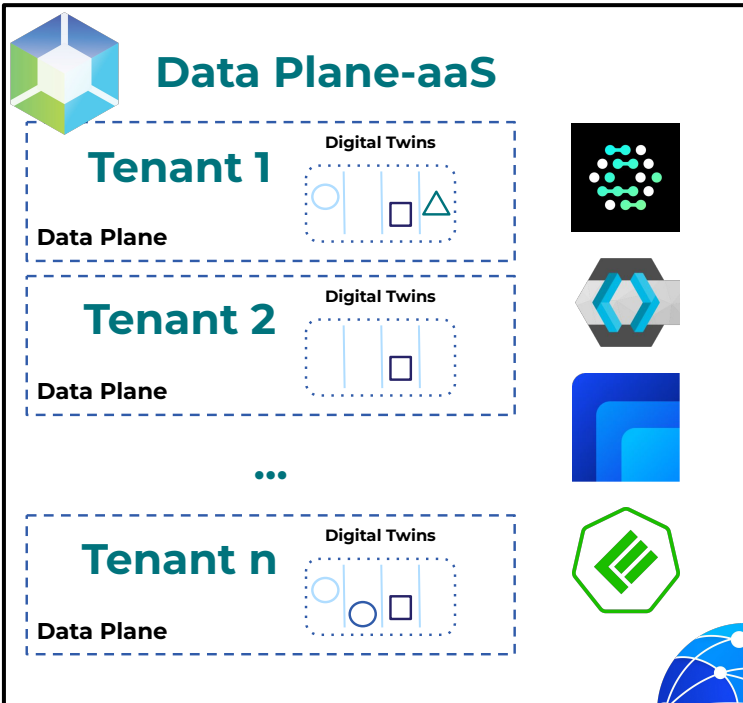


KubeCon



CloudNativeCon

India 2026



OpenFGA for reBAC

KeyCloack for OIDC

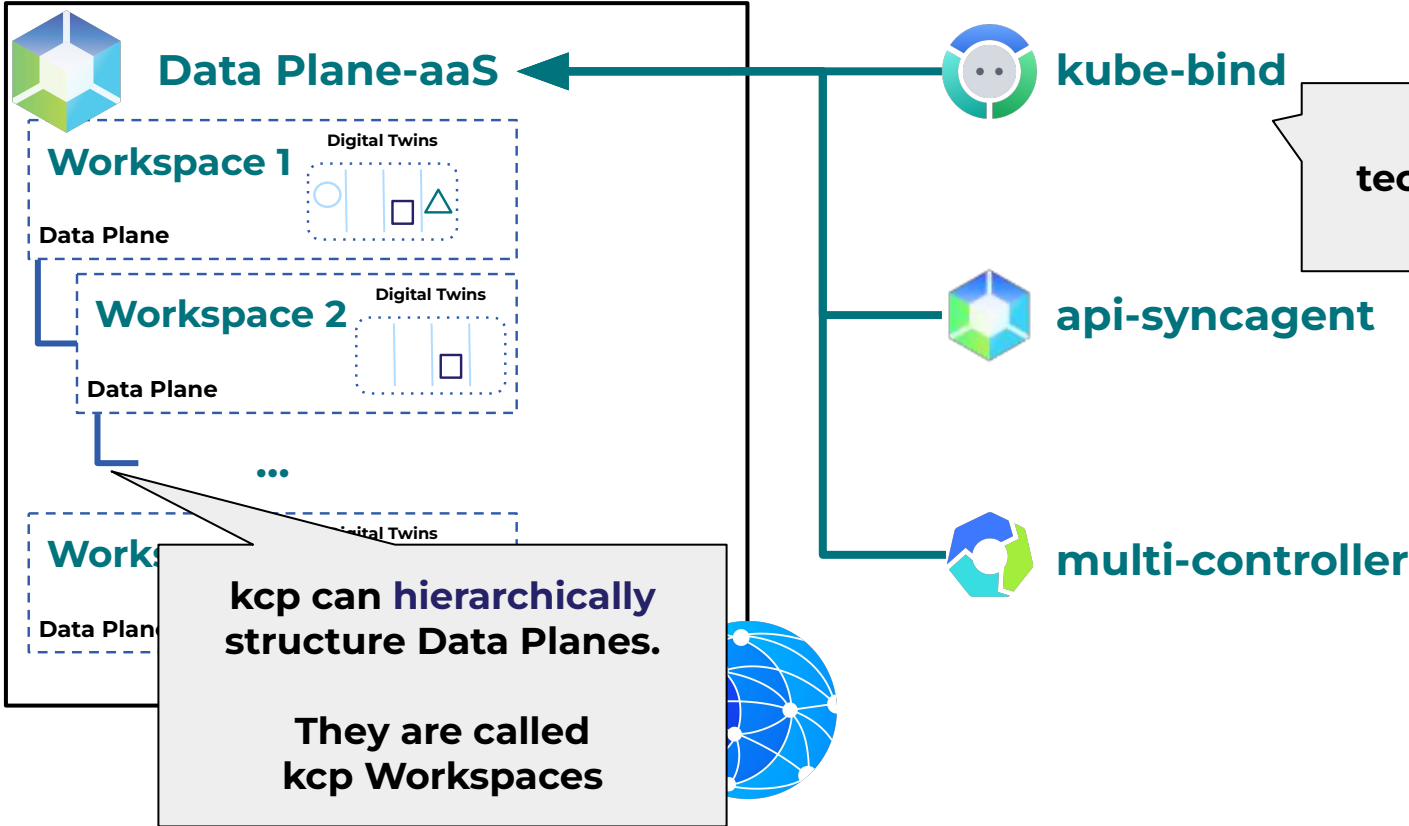
OpenMFP for Micro-Frontend Platform

Luigi for Micro-Frontend Framework

...

Platform-Mesh

Platform Engineering



Platform Engineering with Providers

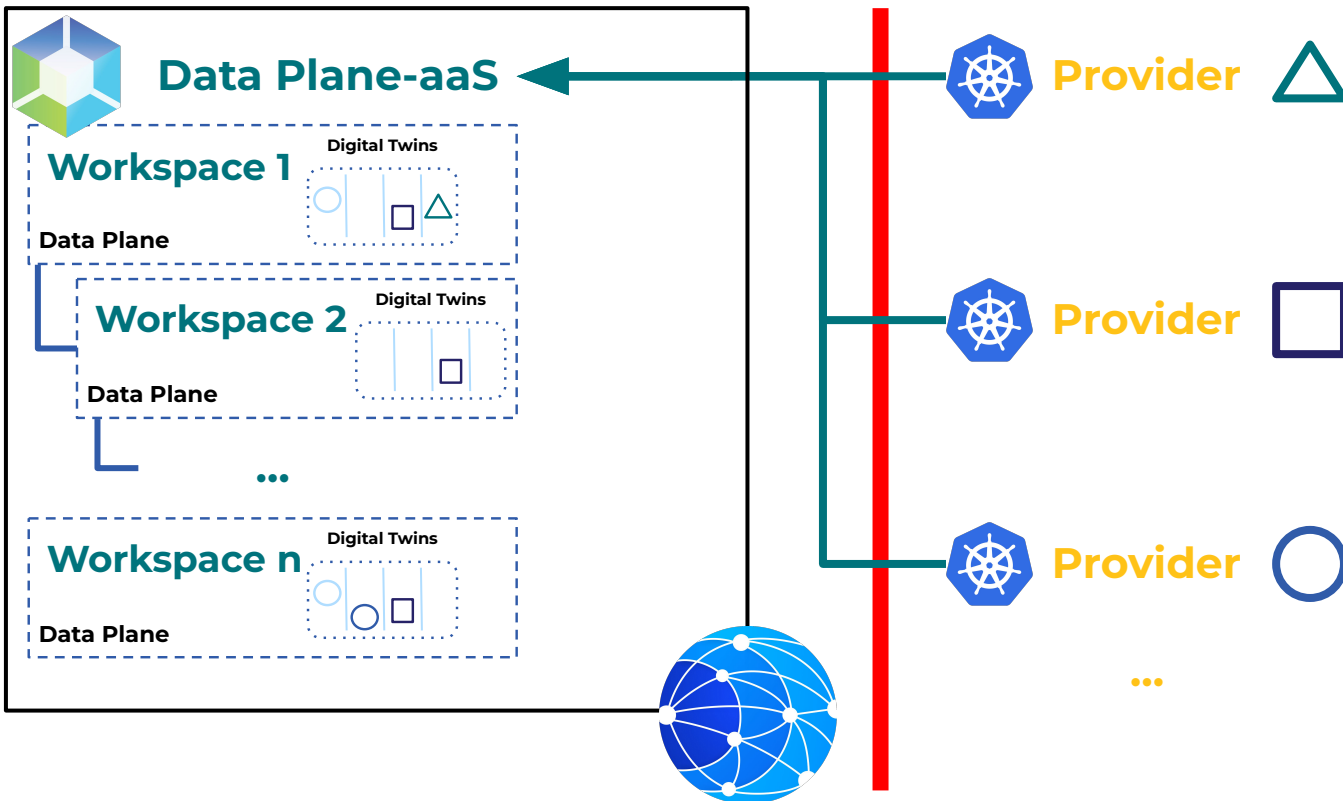


KubeCon

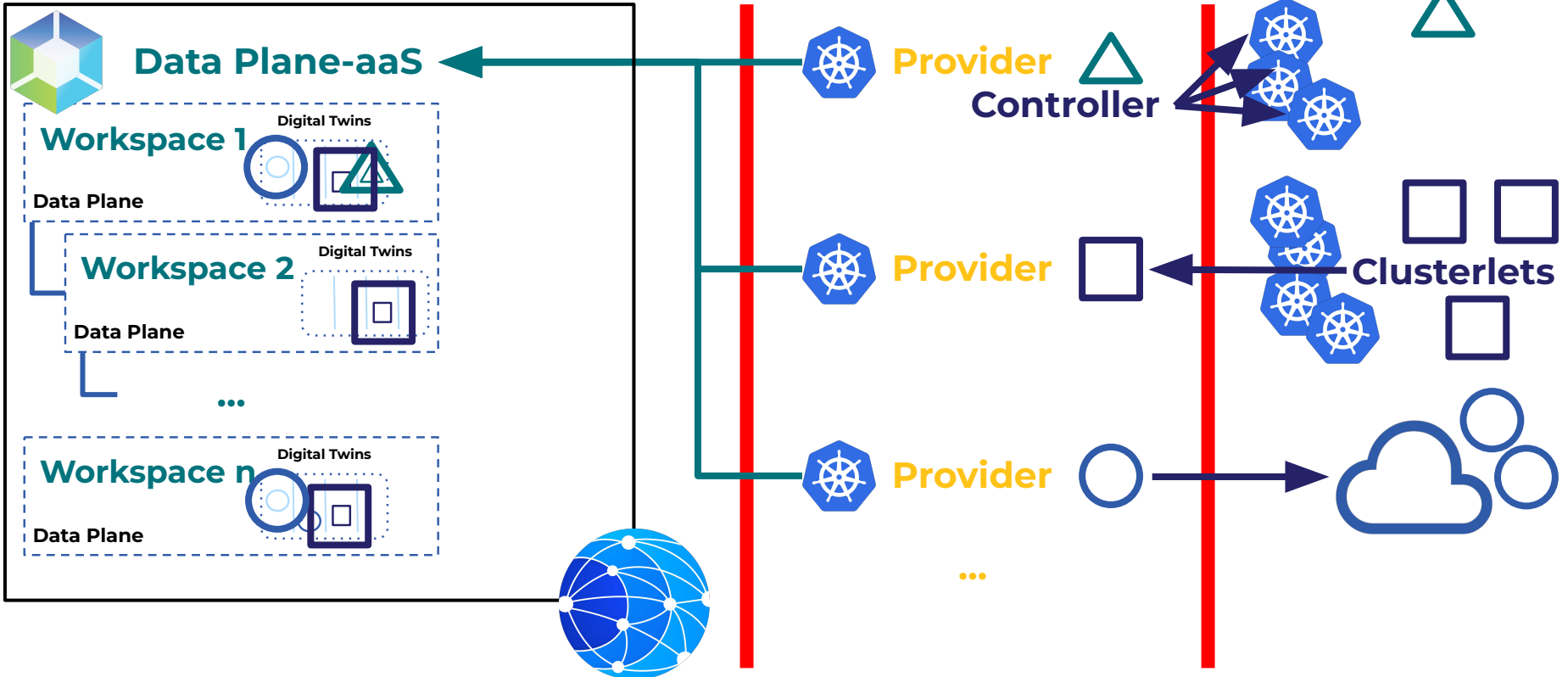


CloudNativeCon

India 2026



Platform Engineering with Providers



Ecosystem Building / Engineering



KubeCon



CloudNativeCon

India 2026

Business Contract

Technical Contract

Data Plane-as-a-Service

Workspace 1

Digital Twins

Data Plane

Workspace 2

Digital Twins

Data Plane

...

Workspace n

Digital Twins

Data Plane

**Digital Twin = Order Contract:
Kubernetes Resource Model &
Status Feedback**

Internal



Own Dev & Product Teams

External



External Contract w/ API & SLA

External



External Contract w/ API & SLA

...

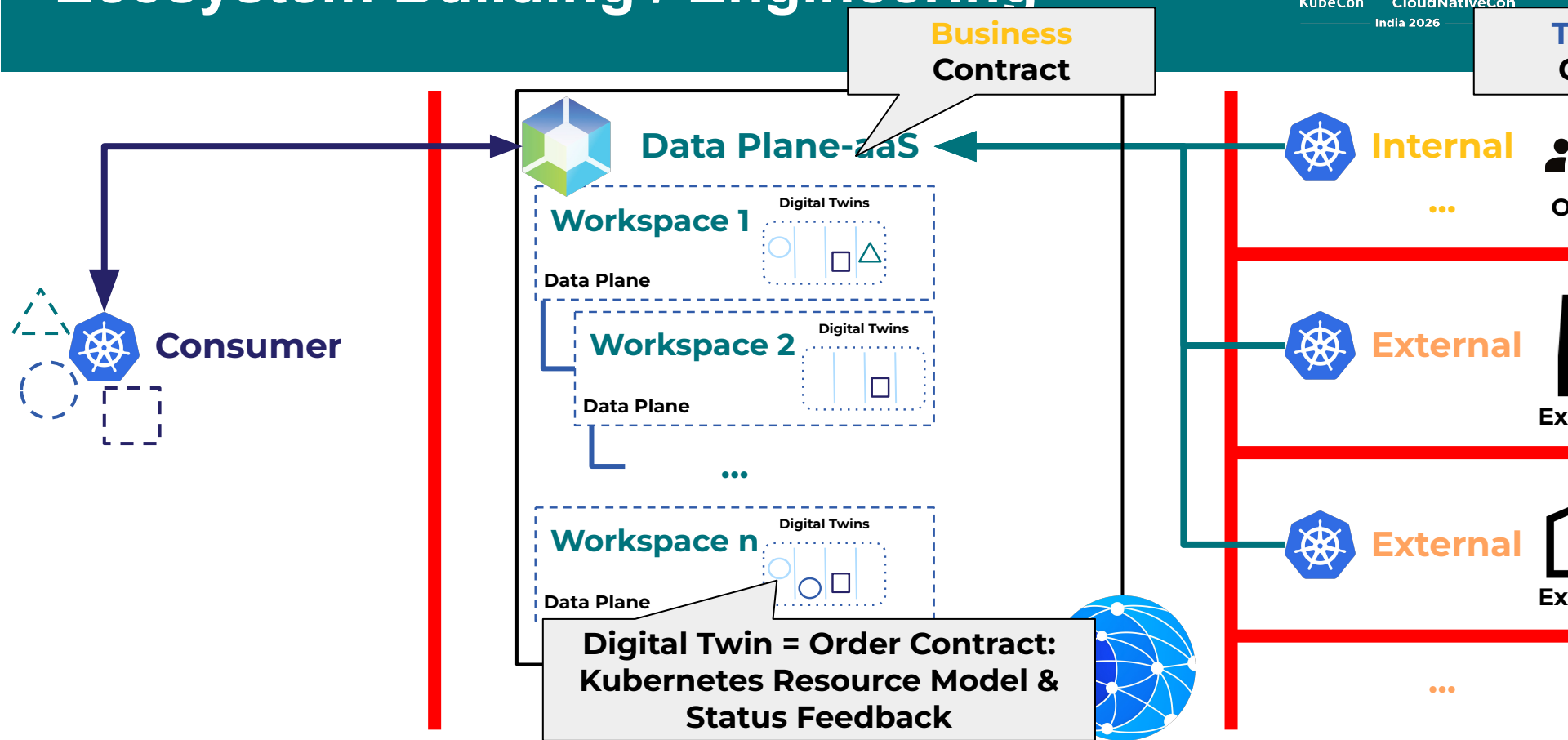
Ecosystem Building / Engineering



KubeCon



CloudNativeCon
India 2026





KubeCon



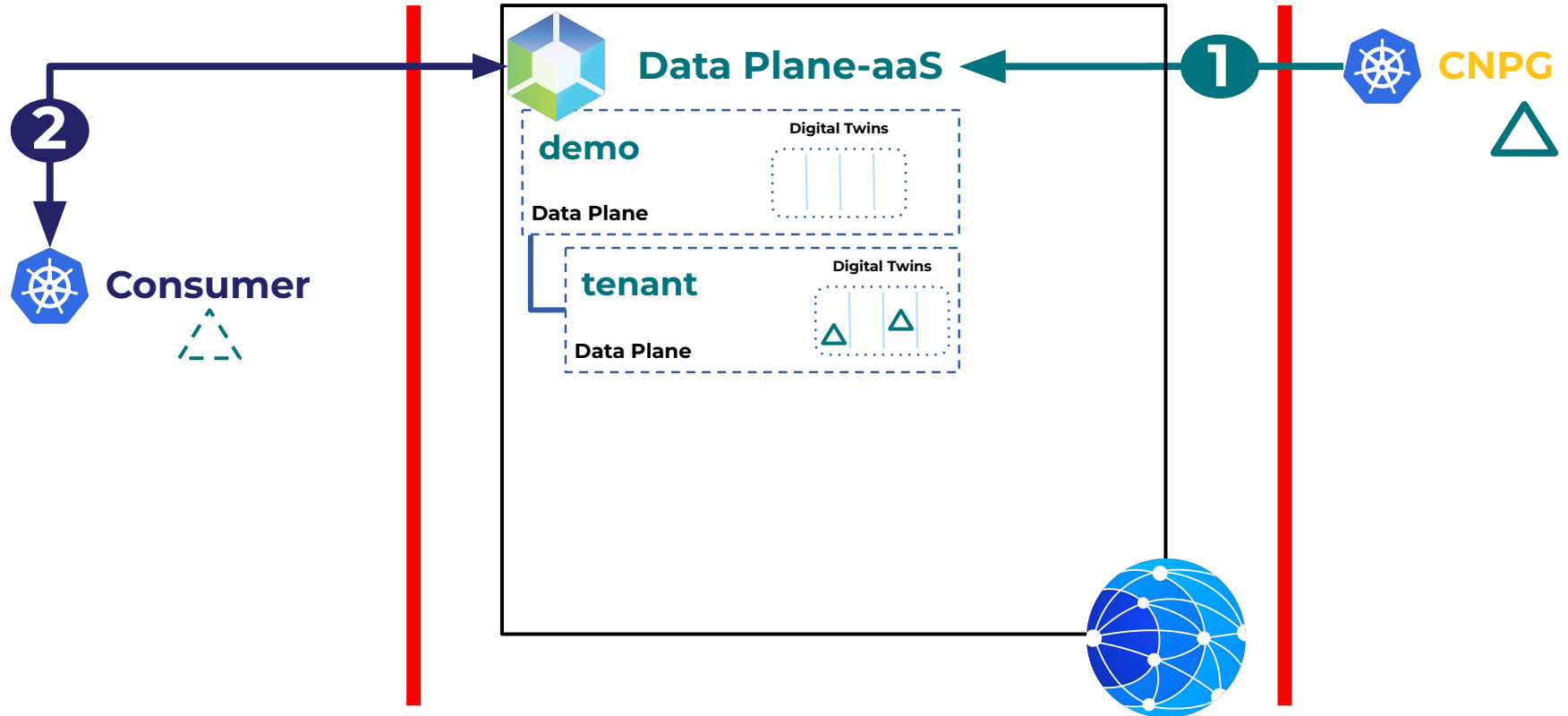
CloudNativeCon

India 2026

Breaking the Architectural Sound Barrier: Demo with Platform-Mesh



Demo



Why is it called Platform-**Mesh**?

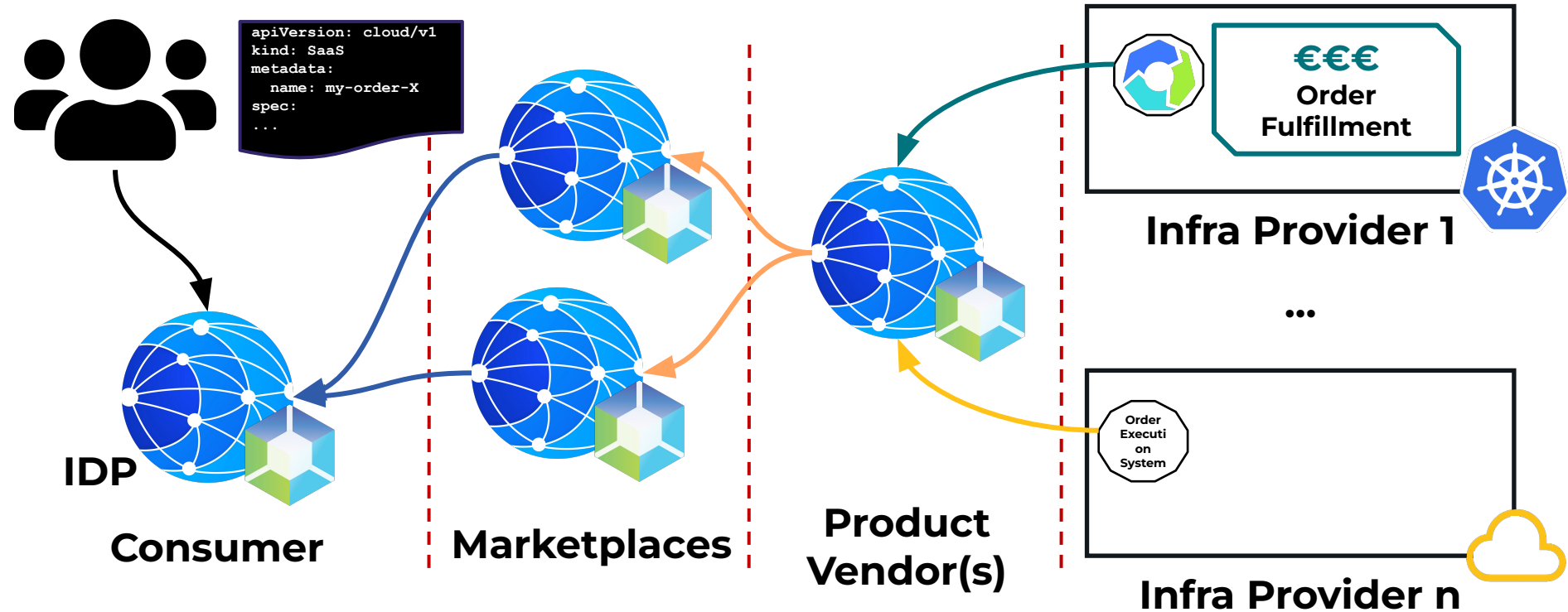


KubeCon



CloudNativeCon

India 2026



Cloud Native KRM = Reduced Cognitive Load

RSVP to **kcpCON** (virtual), Oct 1st



KubeCon



CloudNativeCon

India 2026



Upcoming KCDs ▾

Host a KCD

Events

Community Groups



Log in

KubeCon + CloudNativeCon India 2026 · 18-19 June · Mumbai · REGISTER NOW



kcpCON

OCTOBER 1 | VIRTUAL

#kcpCON

kcpCON 2026

Oct 1, 4:00 – 7:00 PM (GMT+2)

Virtual Project Events (Hosted by CNCF)

Login to
get
tickets

Submit a Talk



KubeCon



CloudNativeCon

India 2026

Thank You!

