



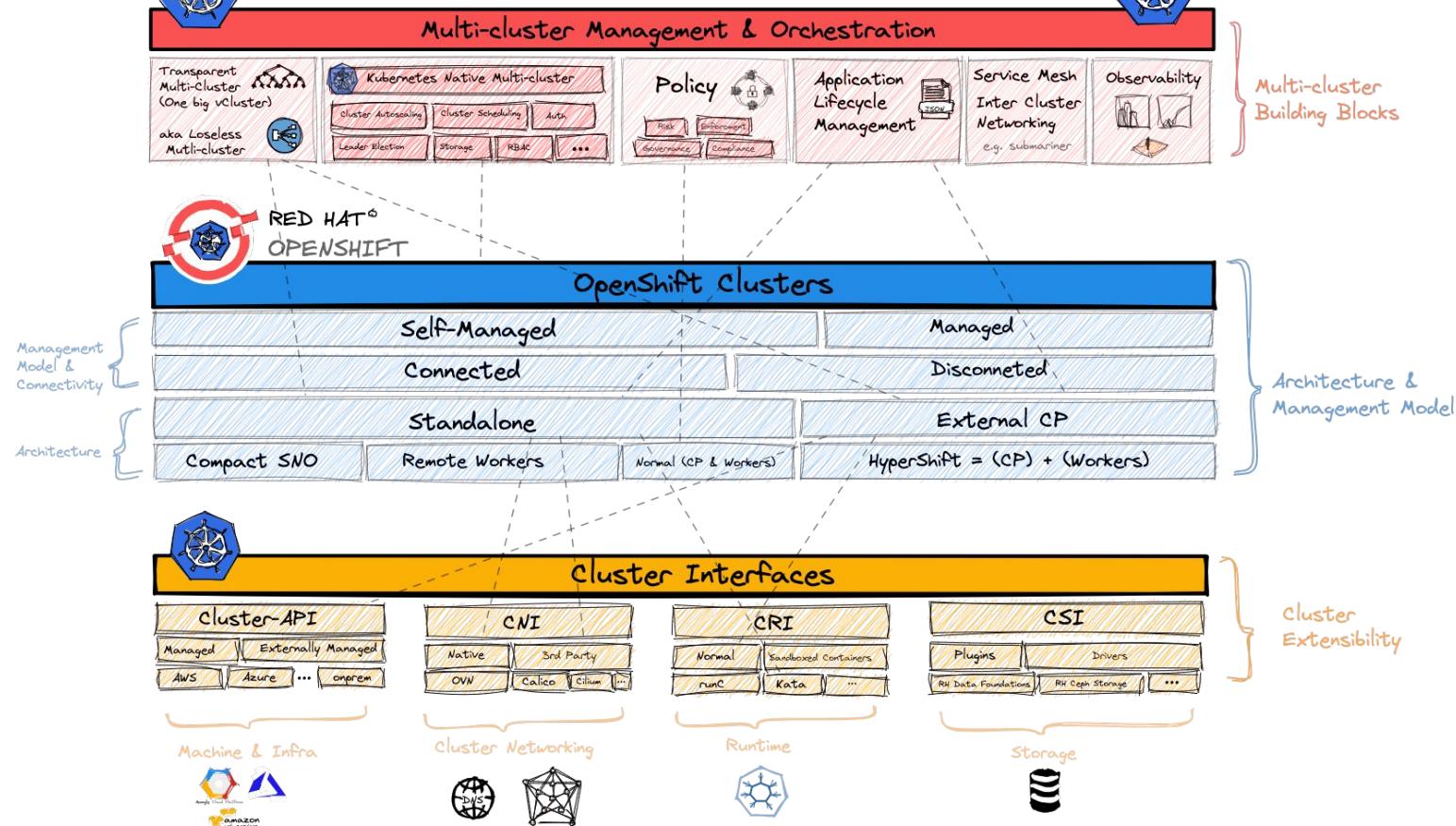
Emerging Multi-Cluster Patterns

A Path to Virtual Dualistic Logically-Centralised Physically-Distributed Clusters

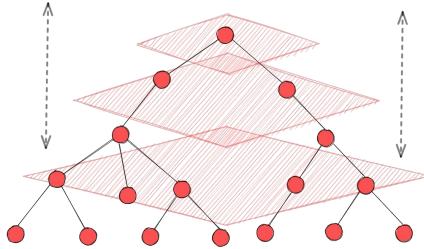
Adel Zaalouk



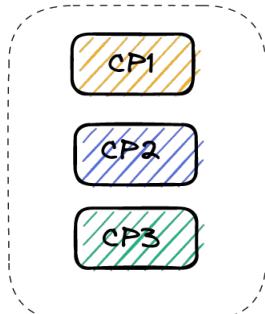
OpenShift Interconnection (OSI) Model



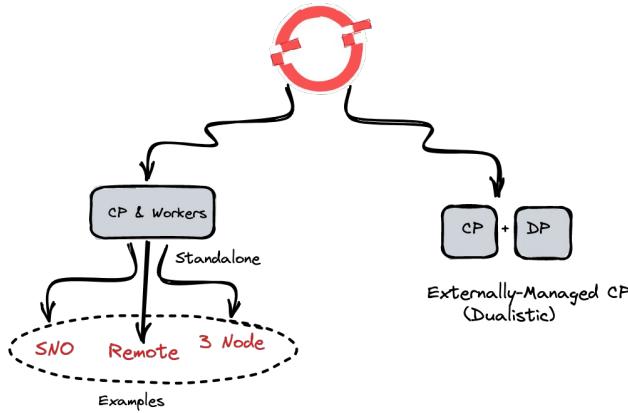
Virtual



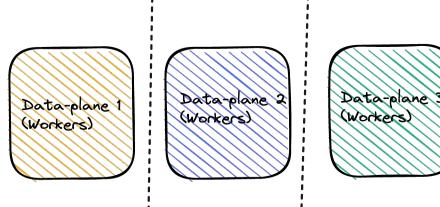
Logically Centralized



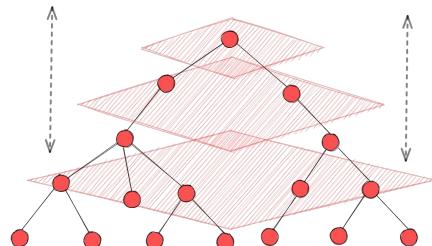
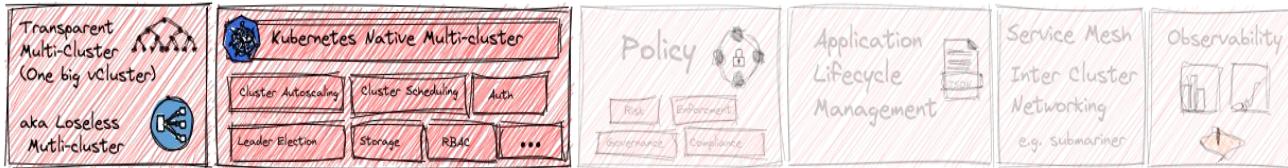
Dualistic



Physically Distributed



Multi-cluster Management & Orchestration



kcp is a minimal Kubernetes API server

Build passing

[kcp-dev/kcp](#): kcp is a prototype of a Kubernetes API server that is not a Kubernetes cluster - a place to create, update, and maintain Kube-like APIs with controllers above or without clusters.

How minimal exactly? `kcp` doesn't know about `Pod`s or `Node`s, let alone `Deployment`s, `Service`s, `LoadBalancer`s, etc.

By default, `kcp` only knows about:

- `Namespace`s
- `ServiceAccount`s and role-based access control types like `Role` and `RoleBinding`
- `Secret`s and `ConfigMap`s, to store configuration data
- `CustomResourceDefinition`s, to define new types
- a handful of other low-level resources like `Lease`s, `Event`s, etc.



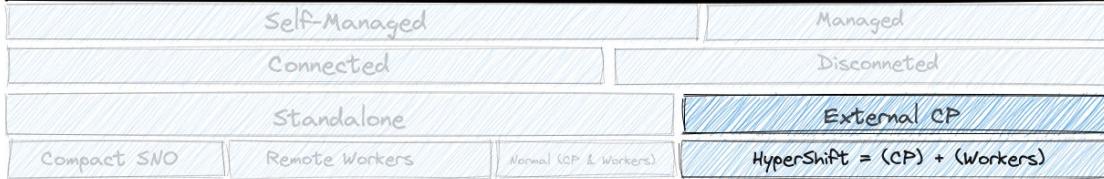
#kcp-prototype



RED HAT® [openshift/hypershift: Hyperscale OpenShift - clusters with hosted control planes](#)

OPENSIFT

OpenShift Clusters



☰ README.md

HyperShift

HyperShift enables [OpenShift](#) administrators to offer managed OpenShift control planes as a service.

How to install the HyperShift CLI

The `hypershift` CLI tool helps you install and work with HyperShift.

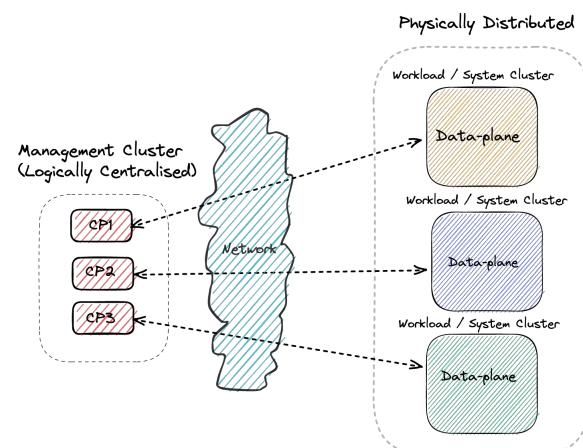
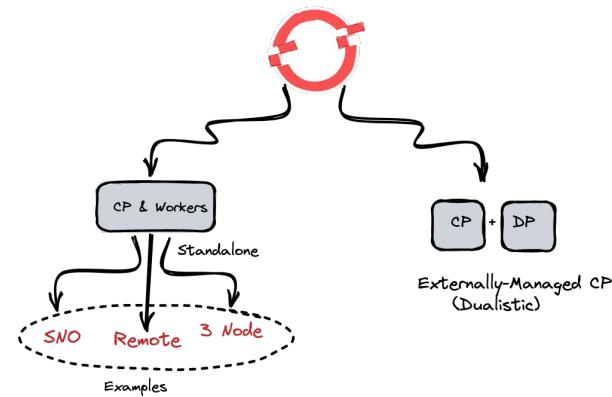
Prerequisites:

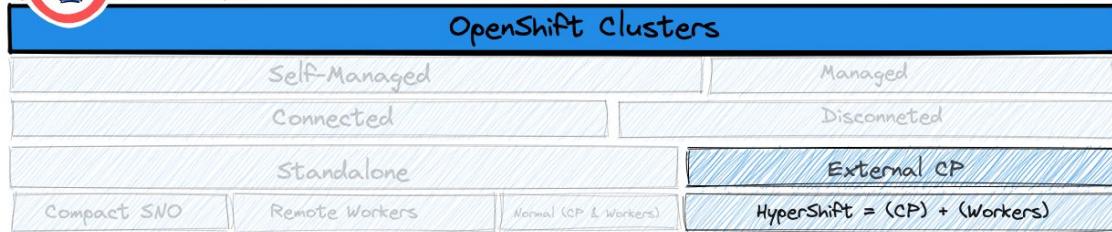
- Go 1.16

Install the `hypershift` CLI using Go:

```
go install github.com/openshift/hypershift@latest
```

The `hypershift` tool will be installed to `$GOBIN/hypershift`.





MGMT and WKLOAD could run on different architectures

Heterogeneous Arch Clusters

Centrally managed control-planes -> centralised multi-cluster management

Easier Multi-cluster Management

Add-ons, layered services can be even more managed (e.g., Service Mesh, ...)

Cross Cluster Delivery Benefits

E.g., Image registry can be shared across tenant reusing the same control-plane cluster

Thinner workers

100s of cluster CPs on the same physical cluster, just a namespace away.
Single Pane of Glass, Happy SREs

Easier Operability

HyperShift Enabler Features

No waiting for machines

Immediate clusters

Multiple CPs per node vs 3 nodes for 1 CP

Cheaper Control-planes

CP as pods, re-use all K8s Goodies
Kubernetes-in-Kubernetes

Isolate tenants, seal away credentials/keys, and prevent users from shooting themselves in the foot

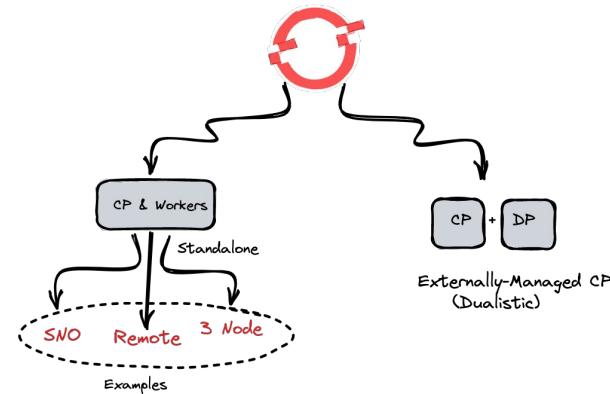
Trust Segmentation & Human Error Reduction

Possible to upgrade MGMT and WKLOAD separately (e.g., CVE, Patches,...)

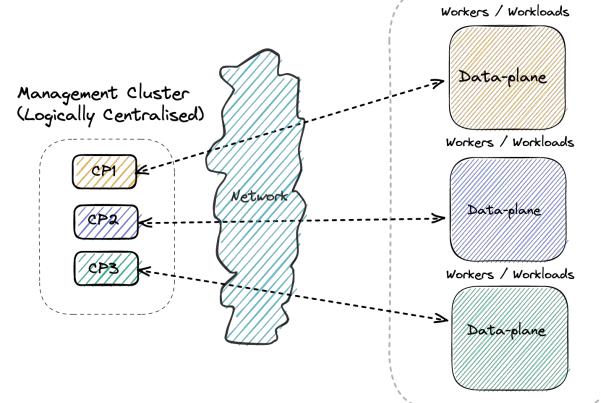
Decoupled Life-cycle Management

MGMT and WKLOAD could run on different providers

Mixed Management & Worker IaaS



Physically Distributed

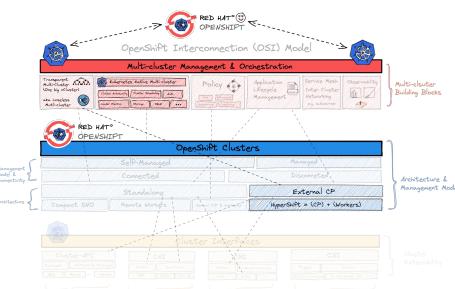
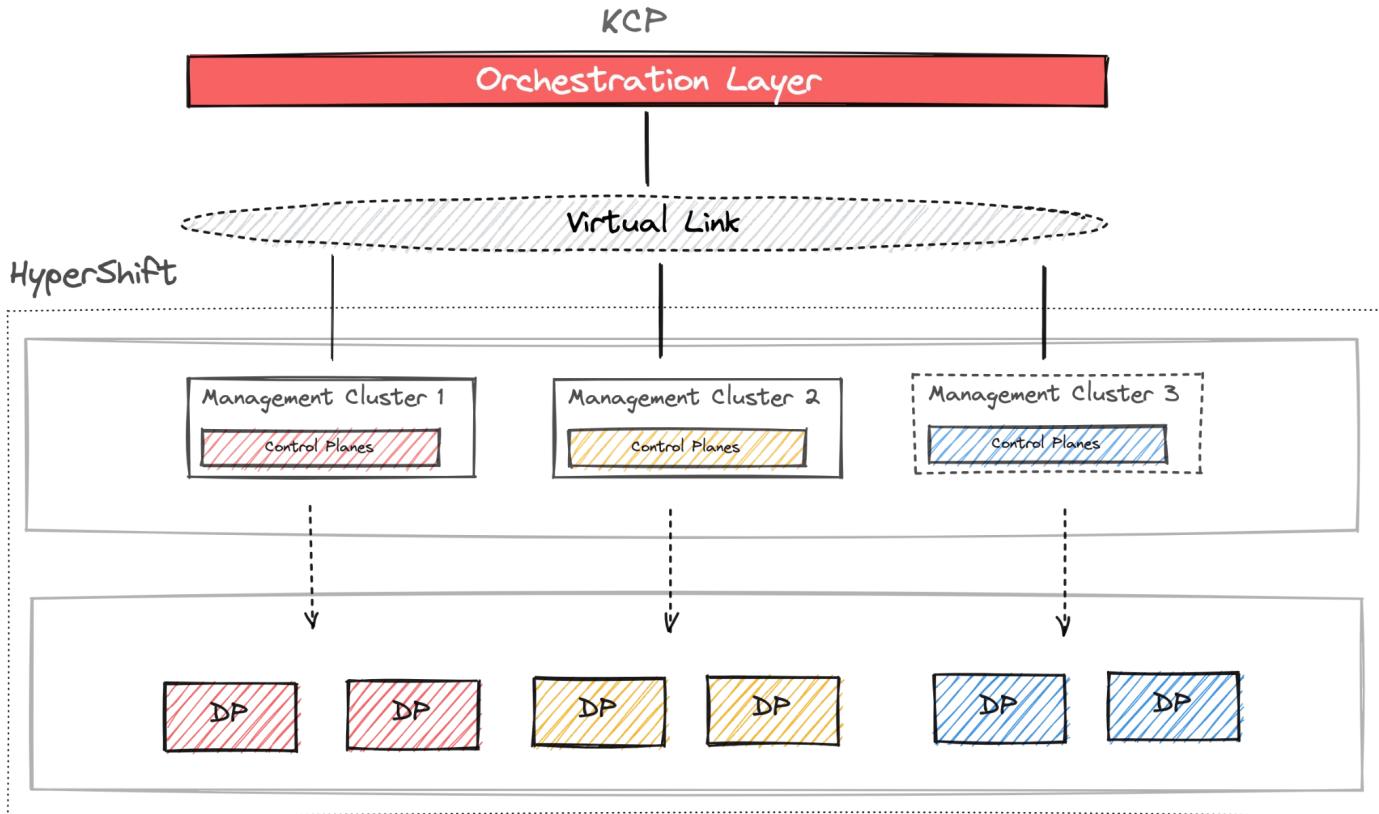


HyperShift

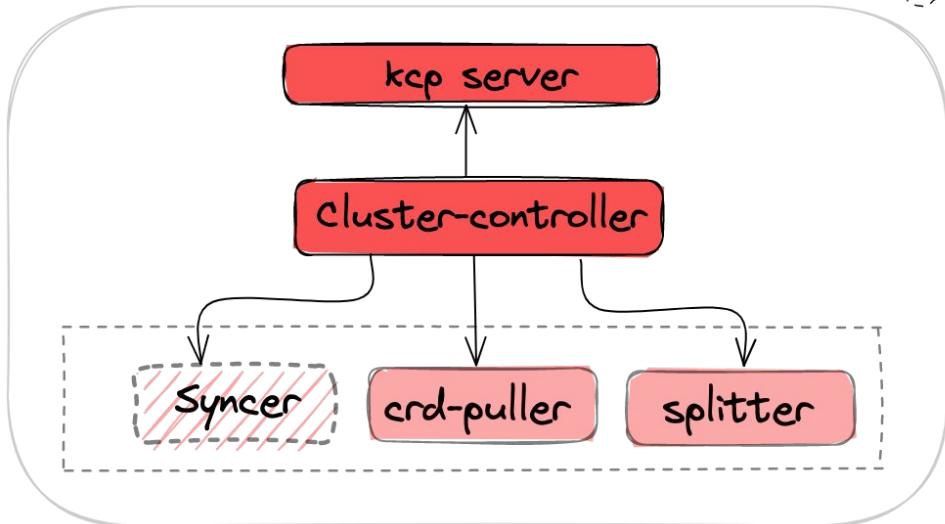


KCP

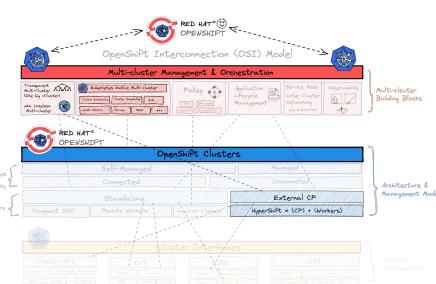
High-Level Architecture



Control Center (KCP)

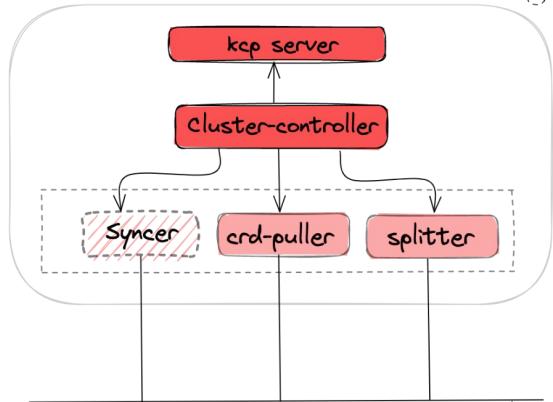


Really Light-weight
only what you need

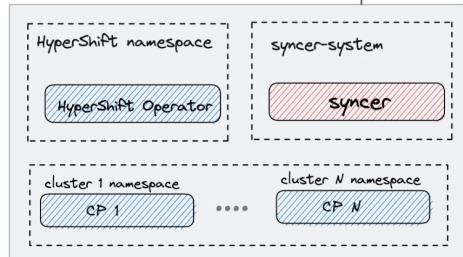


Control Center (KCP)

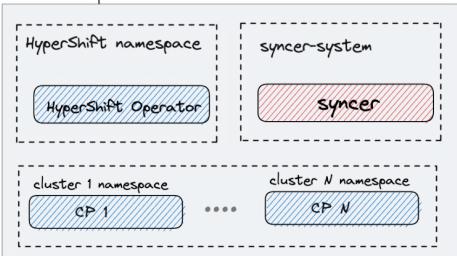
Really Light-weight
only what you need



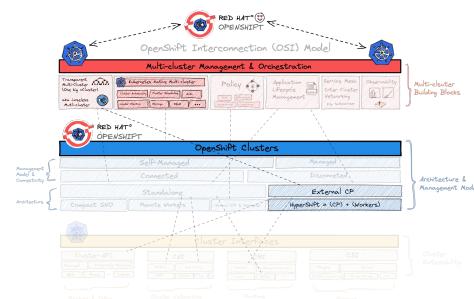
HyperShift



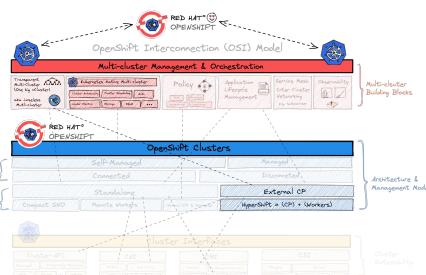
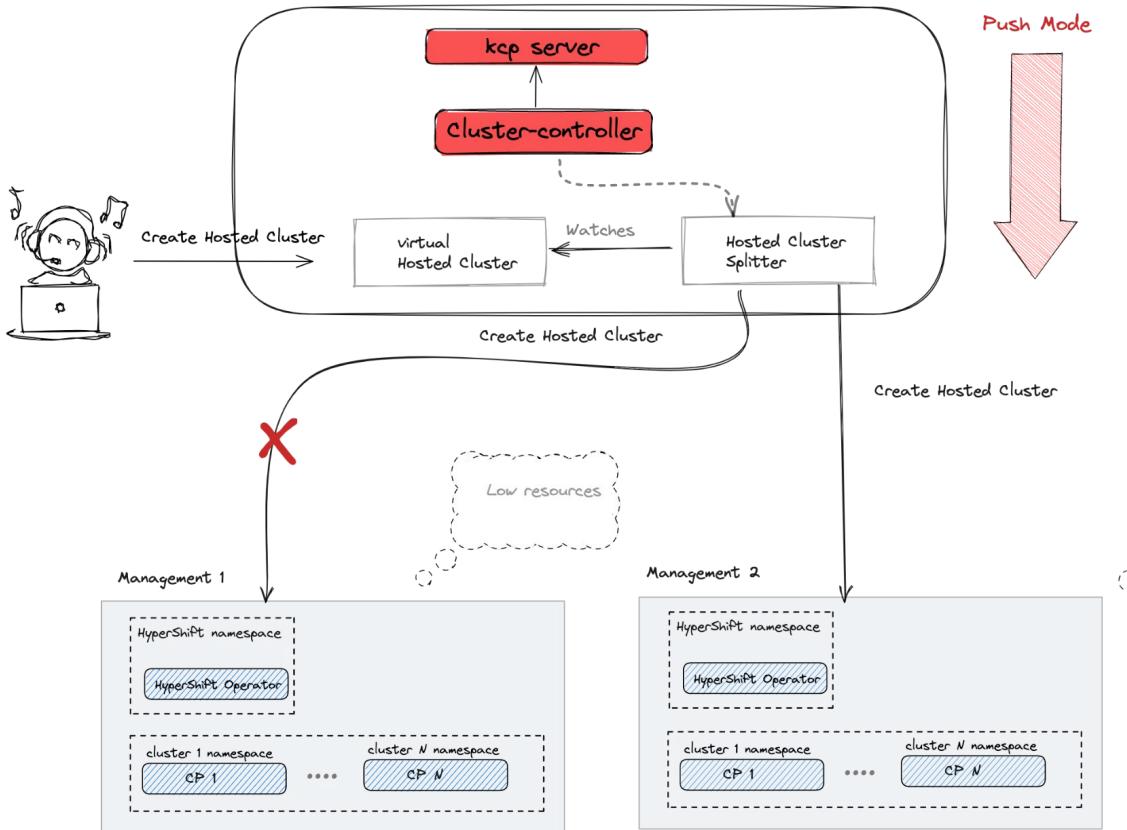
Management Cluster 1



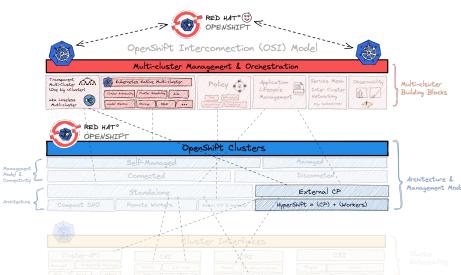
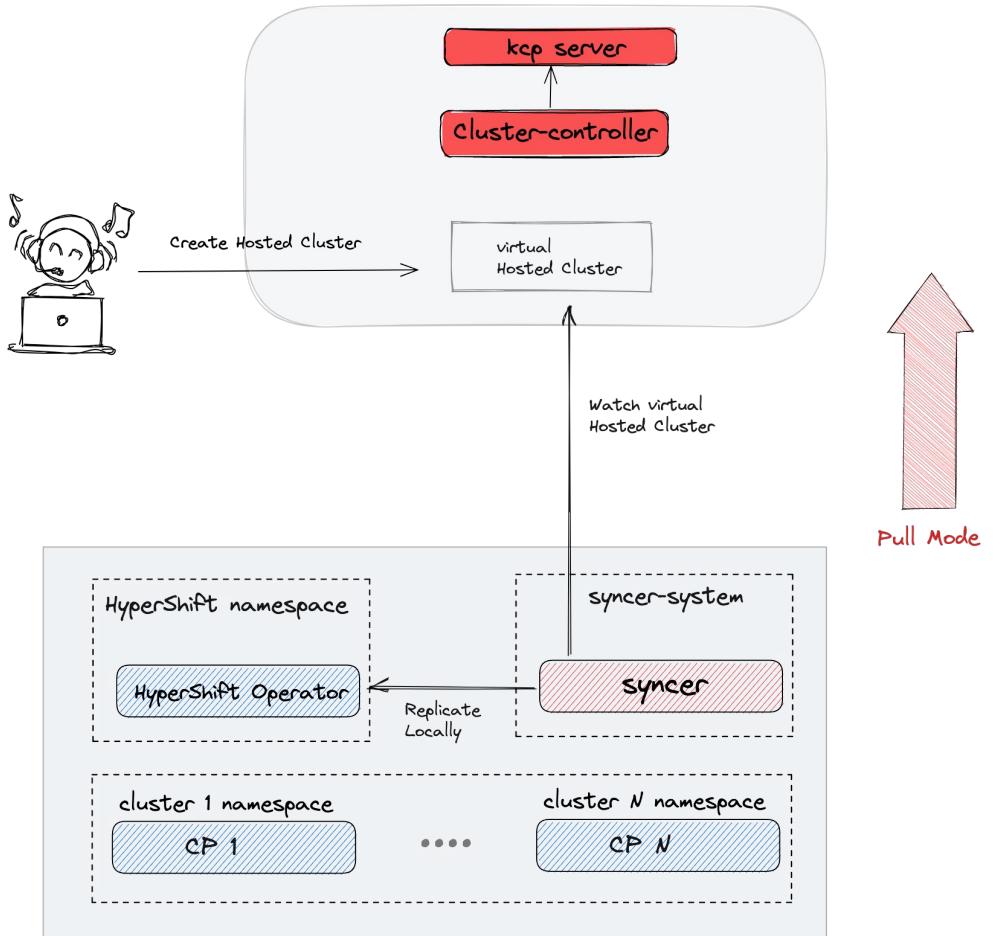
Management Cluster 2



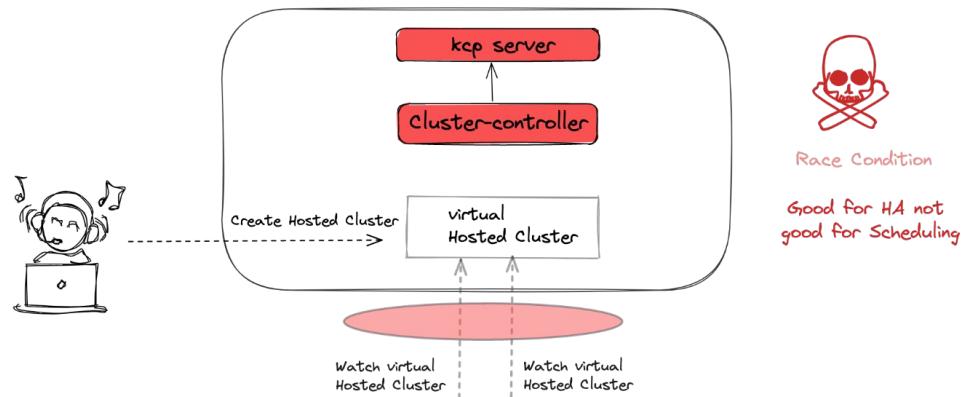
Control Center (KCP)



Control Center (KCP)

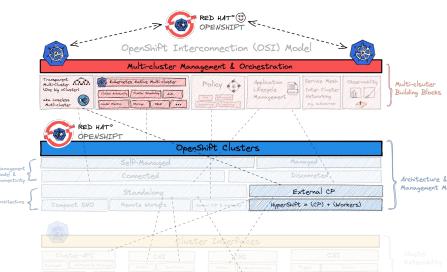


Control Center (KCP)

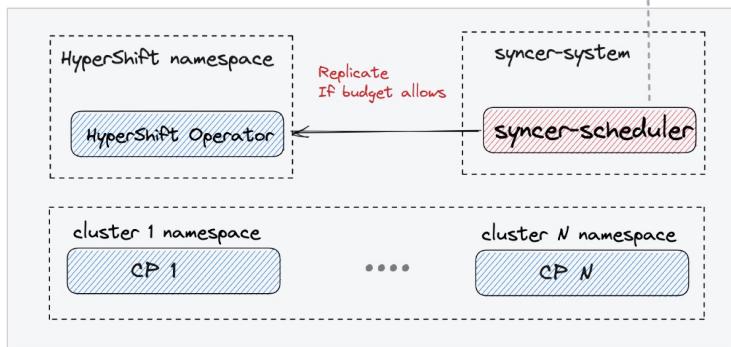


Race Condition

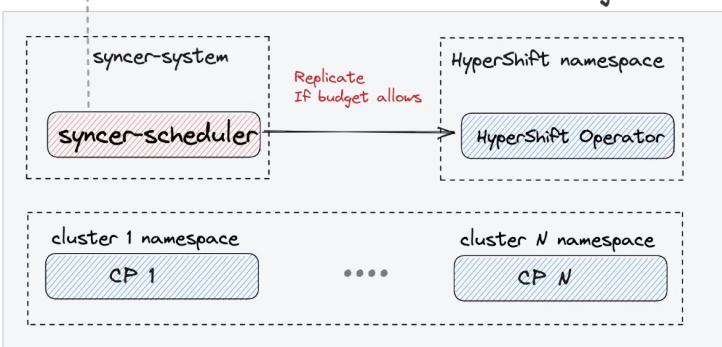
Good for HA not
good for Scheduling



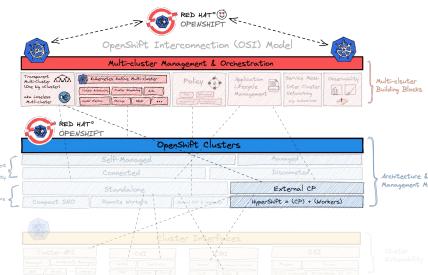
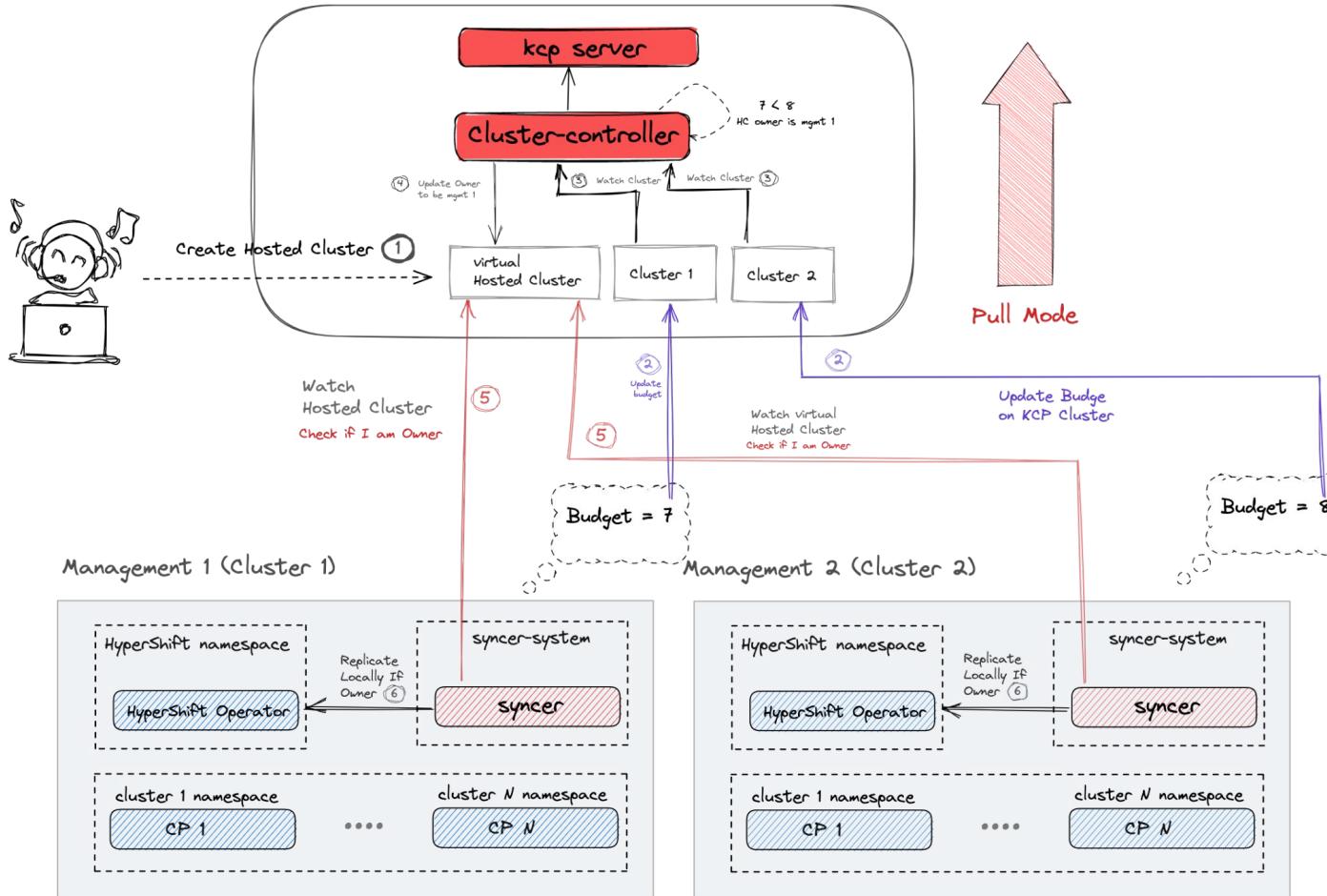
Management 1



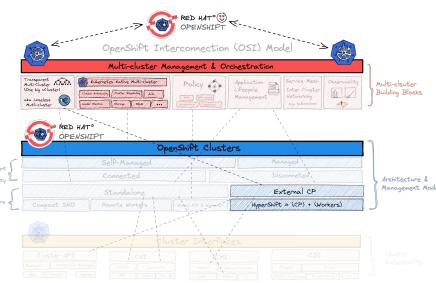
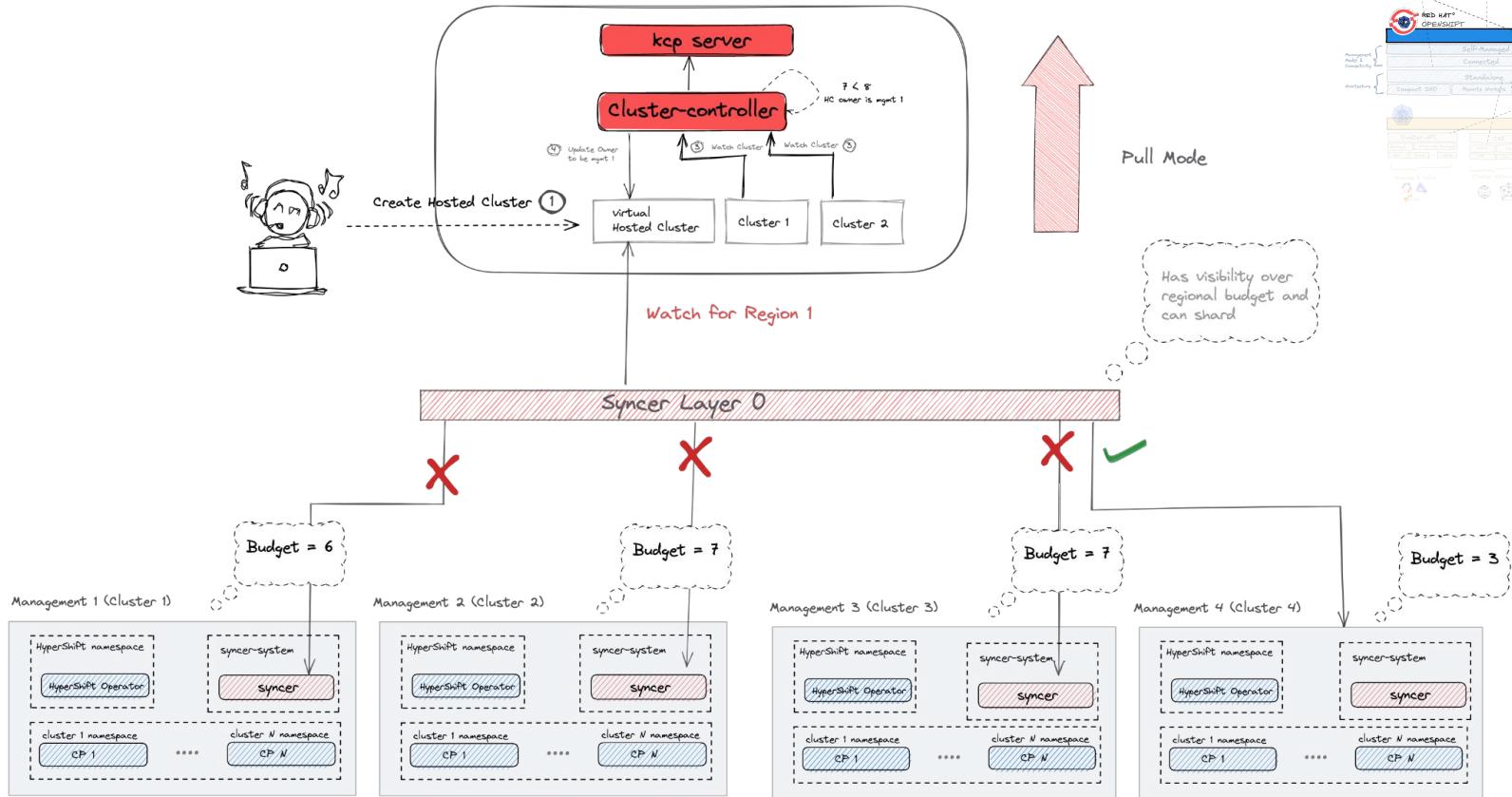
Management 2



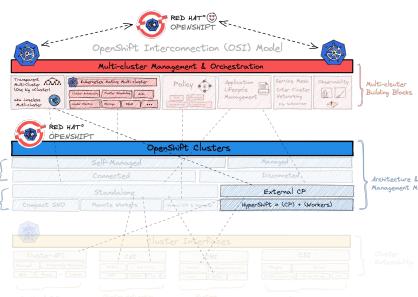
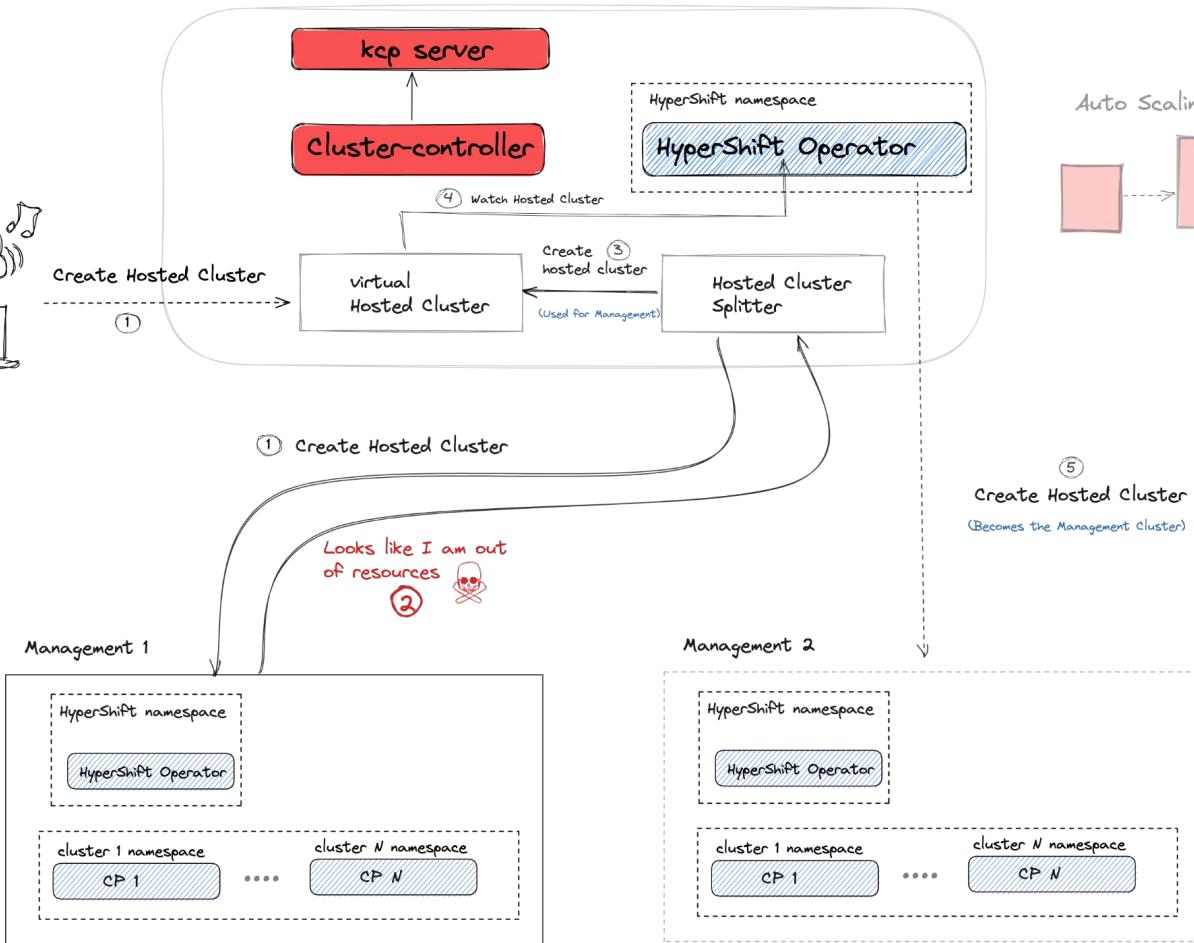
Control Center (KCP)



Control Center (KCP)

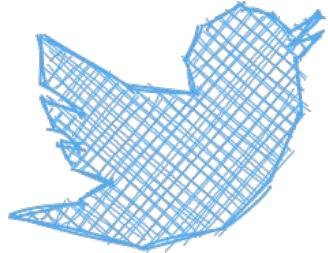


Control Center (KCP)

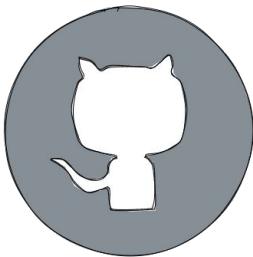


Its DEMO Time!





[Adel Zaalouk \(@ZaNetworker\)](#)



[zanetworker \(Adel Zaalouk\) · GitHub](#)



[Adel Zaalouk – Senior Product Manager - Technical – Red Hat](#)