



**NET  
MAKERS**

# Creazione di servizi Multi-Cluster e Multi-Cloud con il framework open- source Ligo.io

Fulvio Riso  
Politecnico di Torino

# Problema: proliferazione dei clusters



## Struttura organizzativa

- Dipartimenti multipli all'interno dell'azienda, con cluster distinti per ognuno di essi (on-prem, in cloud, etc)

## Cloud multipli

- Cloud pubblico + server on-prem; esigenze di robustezza e/o no lock-in

## Vincoli geografici

- Rispetto di legislazioni locali; necessità di garantire un certo livello di prestazioni indipendentemente dalla posizione del client

## Scalabilità

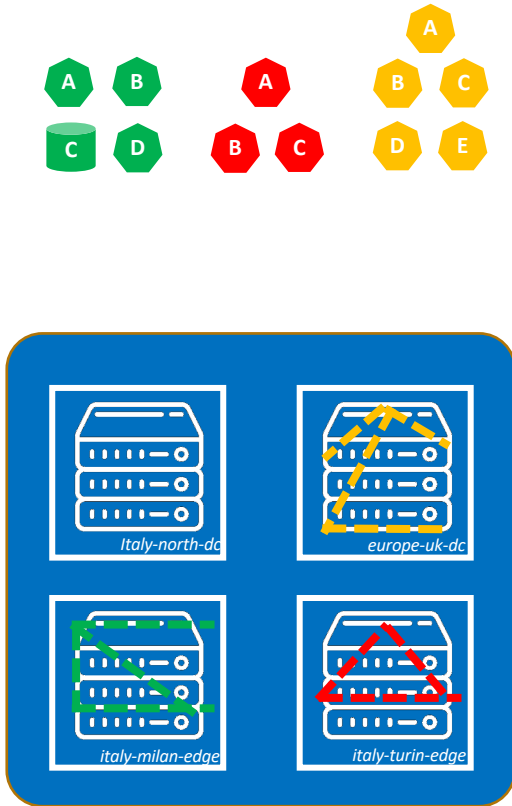
- Necessità di supportare cluster più grossi rispetto alle dimensioni "standard" di Kubernetes

## Separazione dei ruoli

- Presenza di diverse entità amministrative (es. Cluster 1 operato da un SaaS provider; Cluster 2 che consuma i servizi precedenti)

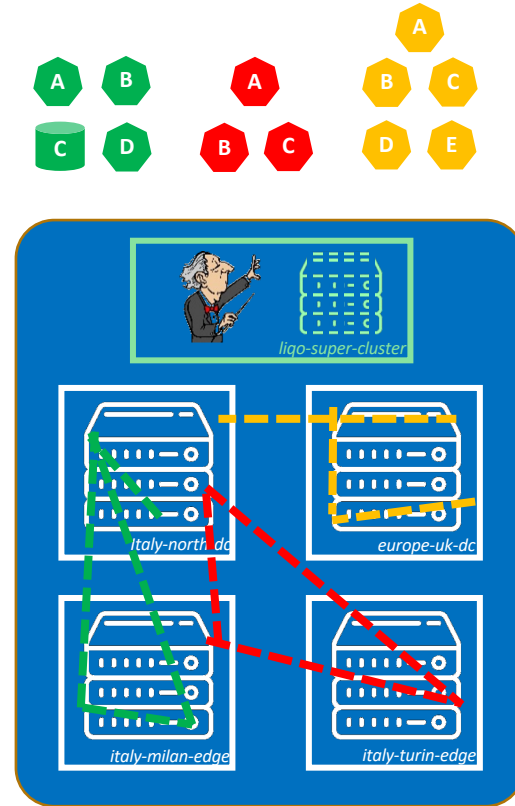
# Cloud/Edge computing

## Silos rigidi



# Liqo

## Orchestrazione senza confini



# Stato dell'arte (progetti open-source)

## Karmada

- Interazione diversa dal classico Kubernetes
- Cinese



## KubeEdge

- Orientato a gestire edge “leggeri” (es. IoT)
- Non supporta “switching fabric”
- Cinese



## Admiralty

- Sviluppo datato
- Community di sviluppo molto ridotta

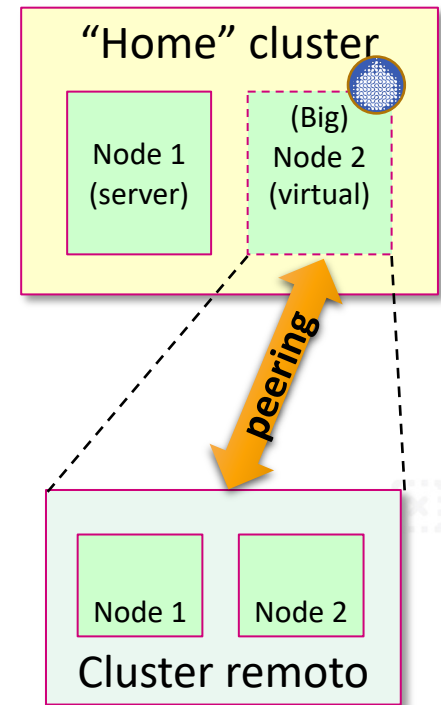


## Tech Mix

- Es. Kubefed + Submariner/Cilium Mesh
- Più complesso
- Impone vincoli su provider e configurazione di rete (CNI)

# Liqo: Virtual Clusters e Big Nodes

- Big Node: “server” virtuale aggiunto al cluster “home”
  - Il cluster “Home” aumenta di dimensione
  - I pod schedulati sul big node vengono creati automaticamente sul cluster remoto
  - Un servizio K8s può essere reso accessibile anche sul cluster remoto
  - Una “network fabric omogenea” collega tutti i pod, ovunque essa siano (indipendenza dalla CNI)
- I “Big nodes” supportano le primitive K8s usate nei cluster fisici
  - Gestione del “virtual cluster” trasparente per l’utente
- Supporta AWS, EKS, AKS, OpenShift, K8s, K3s
- Altamente dinamico e plug-and-play
  - Peering/Unpeering in 2-3 secondi



# Casi d'uso

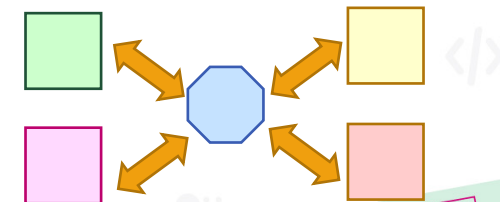
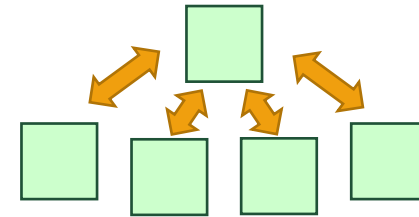
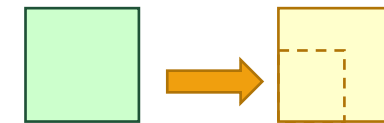
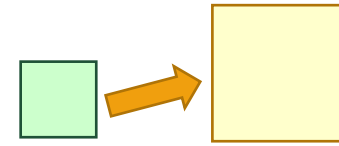
Cloud bursting (POLITO)

Risorse cloud remote (es. GPU)  
(Partner #1)

Infrastrutture distribuite  
edge/cloud (Partner #2)

Cluster virtuali multi-cloud  
(Partner #3)

Brokering di risorse e servizi  
(TOP-IX)

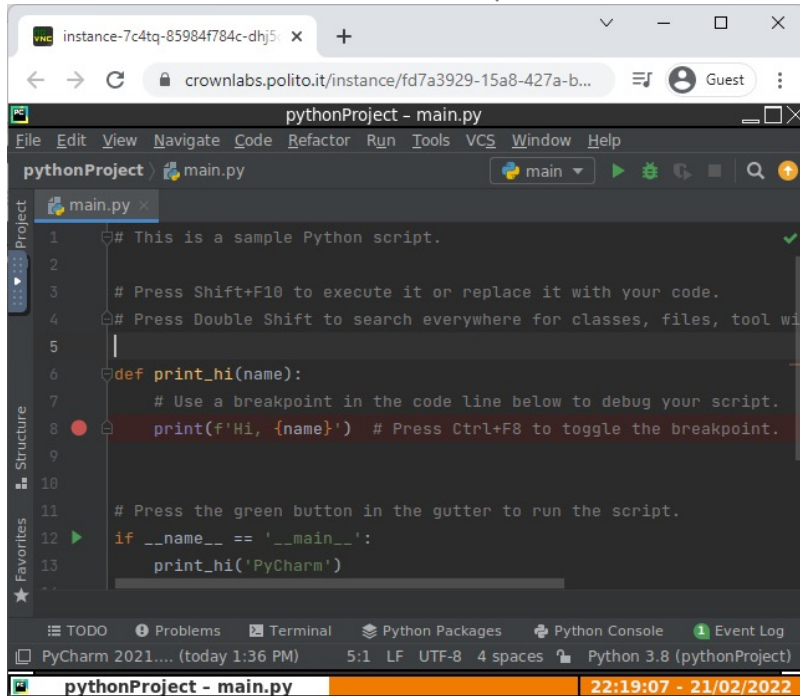


# Caso d'uso: ExamsWithCrownLabs@POLITO

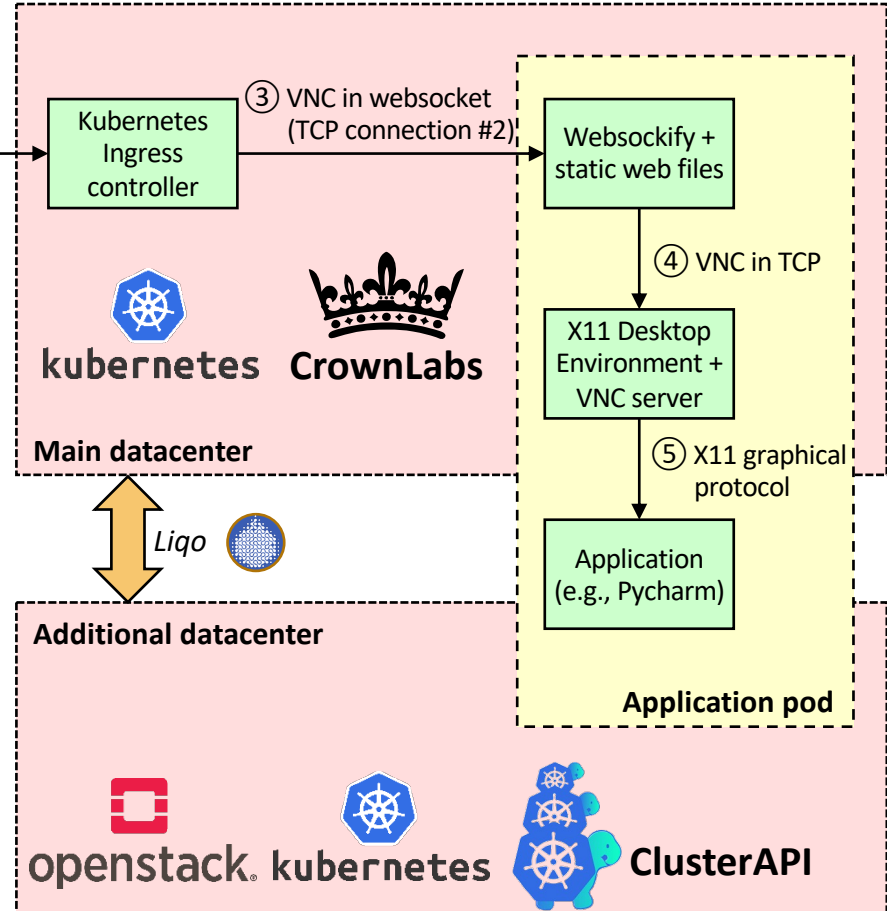
① User with standard web browser



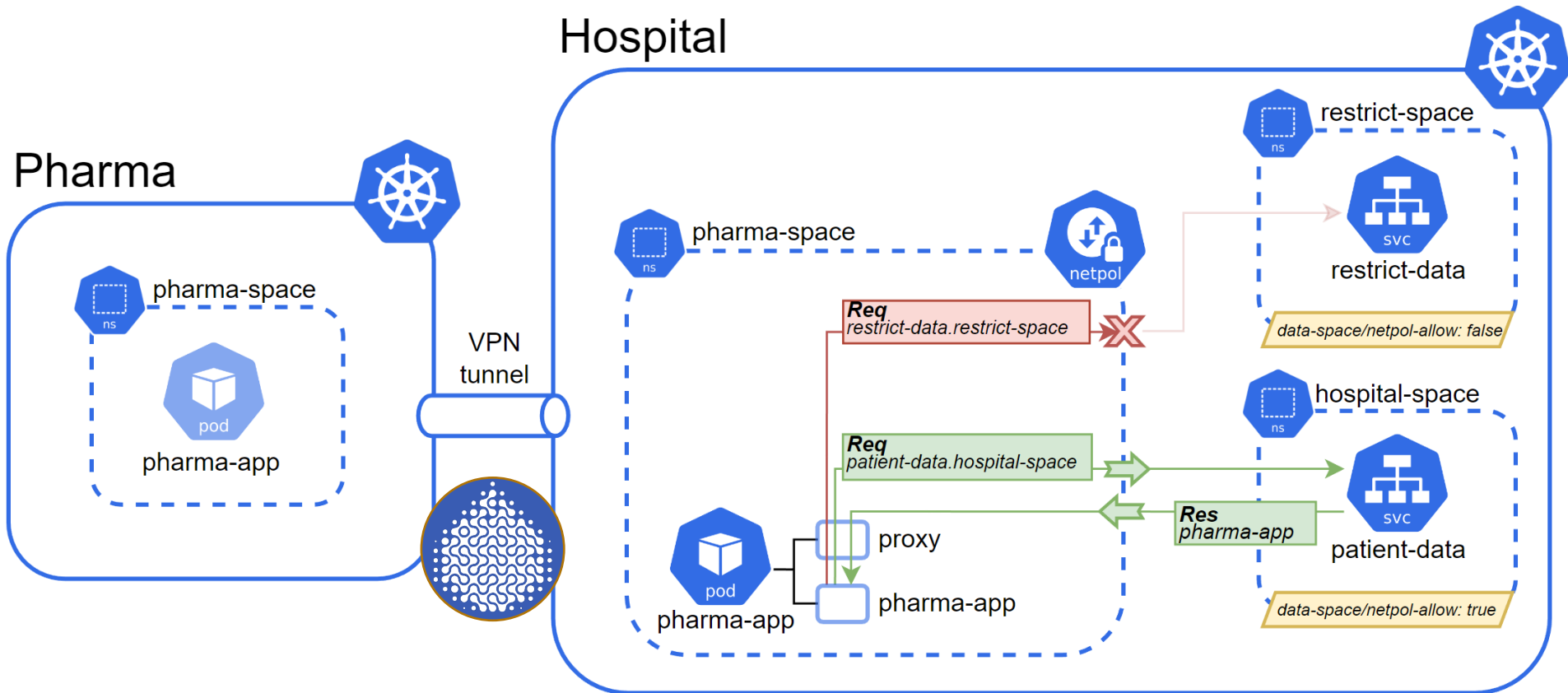
② VNC in websocket (TCP connection #1)



③ VNC in websocket (TCP connection #2)

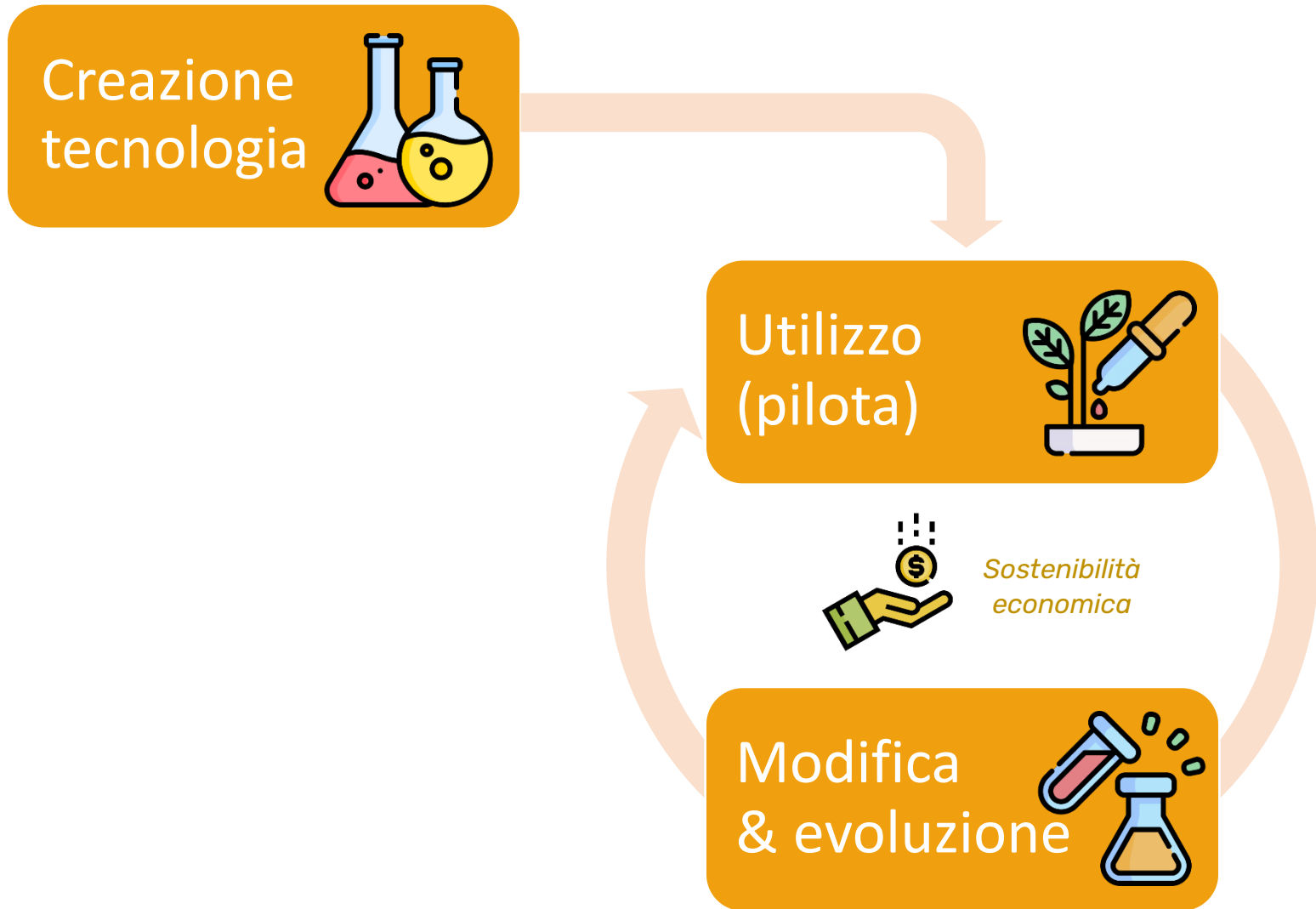


# Data Spaces a livello infrastrutturale (con GEANT)





# Il circolo virtuoso



# Cloud federato Italiano?



# Liqo per il Cloud Europeo

- Obiettivo
  - Consentire ai (piccoli) cloud provider europei di competere con gli hyperscalers
- Come?
  - (1) Rendere l'infrastruttura cloud una "commodity"
  - (2) Ridurre il costo di switching da un cloud provider all'altro
  - (3) Favorire la creazione di servizi "managed" al di fuori degli hyperscalers
- Perché Liqo?
  - Permette di unire cluster diversi in un "virtual cluster"
  - Permette di spostare pod e servizi all'interno del "virtual cluster", da un cloud provider C1 a C2
    - Ogni attore mantiene il controllo sul proprio cluster
    - I servizi/pod locali abilitati sono accessibili dal cluster remoto
  - Perché permette a servizi erogati da un cluster C1 di essere consumati da un secondo cluster C2, come se fossero locali
  - Perché permette l'estensione di un cluster su altri cloud ad es. per poter offrire risorse specializzate (es. GPU) e/o geolocalizzate (es. in UK)



# Per approfondire




<https://liqo.io>



@liqo.io



 stars 719

# Domande?

