KubeCon | CloudNativeCon

North America 2024

Attribution: https://www.flickr.com/photos/ktoddstorch/64580032

KubeCon | CloudNativeCon
North America 2024

Micro-Segmentation & Multi-Tenancy

The Brown M&Ms of Platform Engineering

About us

Rachael Wonnacott

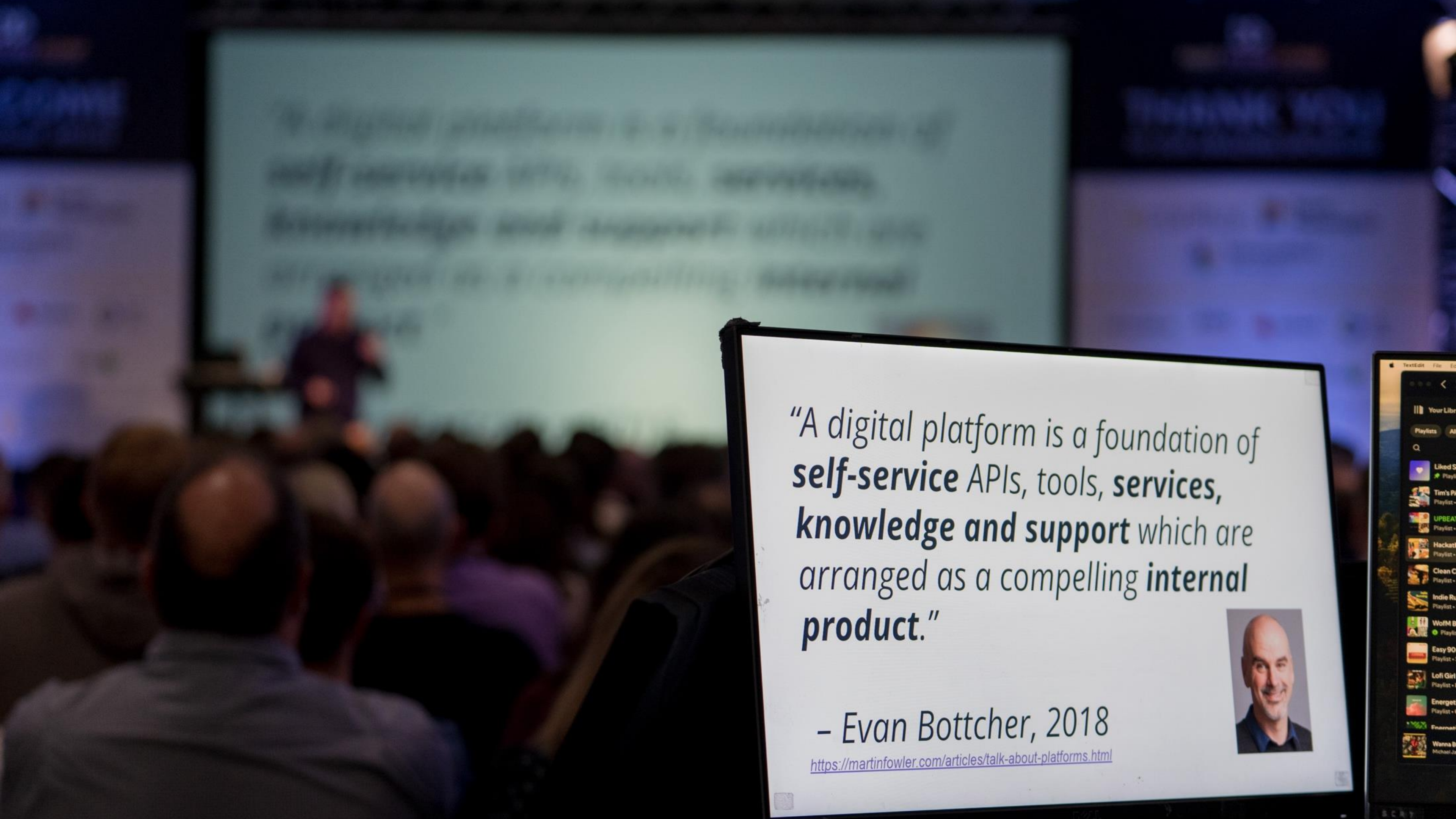
Associate Director, Platform Engineering at Fidelity International

Jim Bugwadia

Co-founder & CEO @ Nirmata; Kyverno Maintainer

Topics

- Platform Engineering
- Multi-Tenancy & Micro-Segmentation
- Demo
- Q&A



"A digital platform is a foundation of **self-service** APIs, tools, **services, knowledge and support** which are arranged as a compelling **internal product**."

– Evan Bottcher, 2018

https://martinfowler.com/articles/talk-about-platforms.html

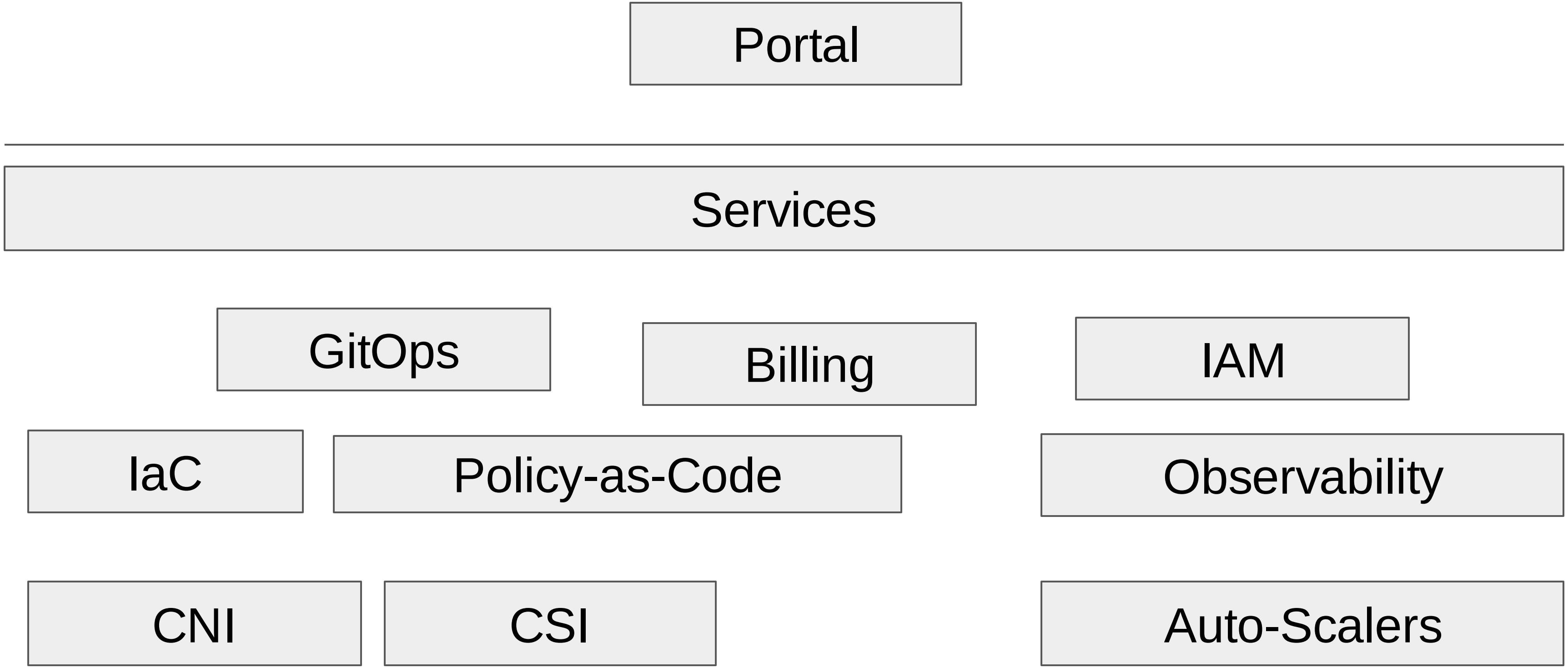
Challenges

- Growing developer ecosystem; new services, tools, languages
- Microservice architectures can suffer from sprawl
- Different teams operate in different ways
- Running in a hybrid model
- Overly distributed systems
- Potentially complex networks and/or latency
- Security threats are increasing
- Evidencing for audit

Standardization

- Platforms can enforce standardisation across workloads
- Reduced volume of unique config – more maintainable
- Simplify evidence for audit
- Reduced cognitive load:
  - all teams follow same best practice
  - leadership only need to understand one (or few) patterns
  - easier to move people between teams
- Easier to onboard new people
- Easier to roll out fixes across all applications
- More predictable spend

Platform MVP



Portal

Services

GitOps

Billing

IAM

IaC

Policy-as-Code

Observability

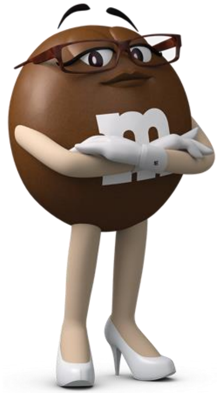
CNI

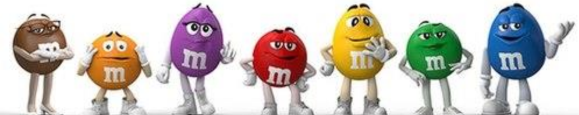
CSI

Auto-Scalers

ulti-Tenancy

&

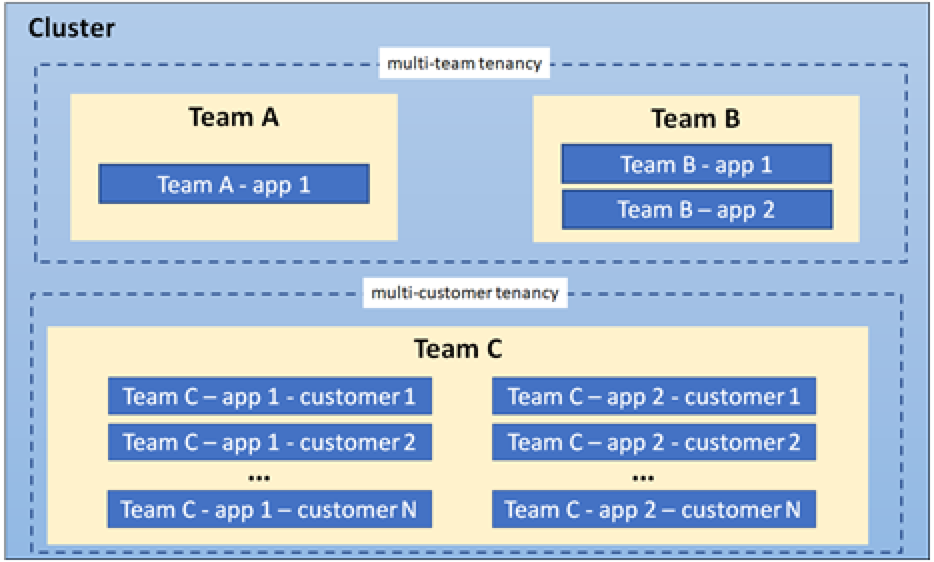
icro-Segmentation



Kubernetes Multi-Tenancy

Three models:

- Cluster-as-a-Service
- **Namespace-as-a-Service**
- Control-Plane-as-a-Service

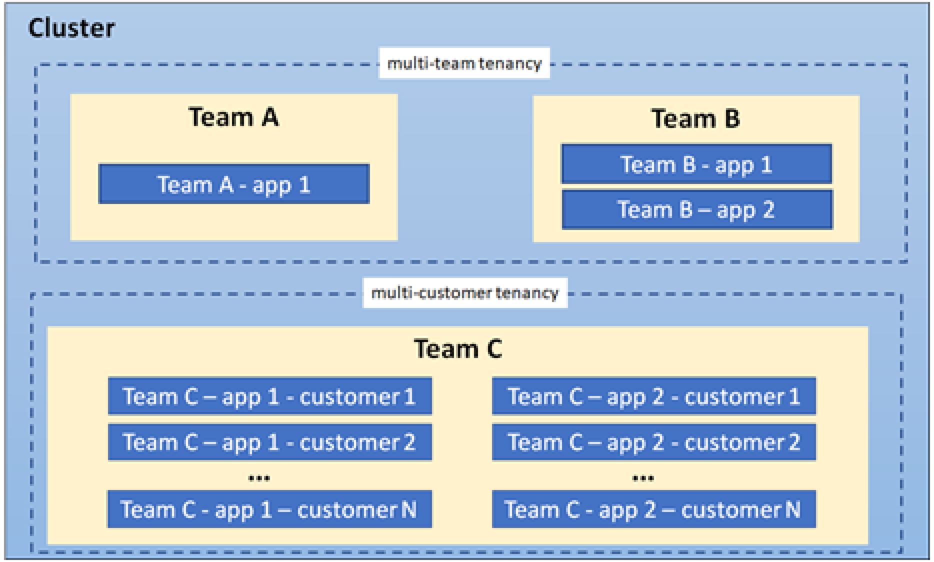

Cluster
multi-team tenancy
Team A
Team A - app 1
Team B
Team B - app 1
Team B – app 2
multi-customer tenancy
Team C
Team C – app 1 - customer 1
Team C – app 1 - customer 2
...
Team C - app 1 – customer N
Team C – app 2 - customer 1
Team C – app 2 - customer 2
...
Team C - app 2 – customer N

Namespace-as-a-Service

https://kubernetes.io/blog/2021/04/15/three-tenancy-models-for-kubernetes/

Why namespace-as-a-service?

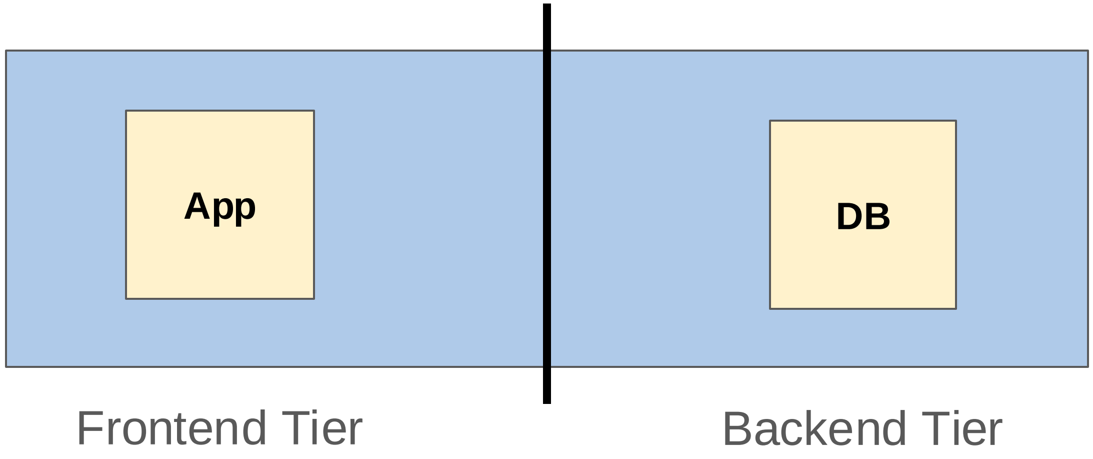
- Enforcing standardisation at the namespace level
- A simple way to isolate development environments
- The right permission boundary for RBAC
- Enables simple application tiering
- Standardised & predictable cost of centralised services
- Learning from the community – avoid cluster sprawl


Cluster
multi-team tenancy
Team A
Team A - app 1
Team B
Team B - app 1
Team B – app 2
multi-customer tenancy
Team C
Team C – app 1 - customer 1
Team C – app 1 - customer 2
...
Team C - app 1 – customer N
Team C – app 2 - customer 1
Team C – app 2 - customer 2
...
Team C - app 2 – customer N

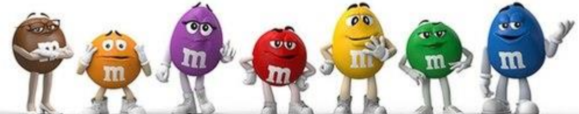
Namespace-as-a-Service

Micro-Segmentation

- Divide network into segments (tiers / zones)
- Enforce security checks to prevent unauthorized movement



Demo



Platform Requirements

1. Secure self-service multi-tenancy
2. Secure self-service network segmentation

+

=

Cilium

- Networking, Observability, Security
- High Performance CNI
- Service Mesh, Egress, Gateway API
- Transparent Encryption
- Metrics, Tracing, Flow Logs
- Network Policies

Adobe Bloomberg Capital One Robinhood The New York Times sky

Kyverno

- Policy as Code
- Low-code policy language
- Admission controller, scanner
- Validate, Mutate, Generate, Cleanup, VerifyImages
- Originally built for Kubernetes ...now works everywhere!

Spotify LinkedIn vodafone wayfair coinbase Bloomberg

Policies & Policies

Cilium Network Policies

“firewall rules”

I want to talk to “Y”

“Z” can talk to me

“App X”

Kyverno Policies

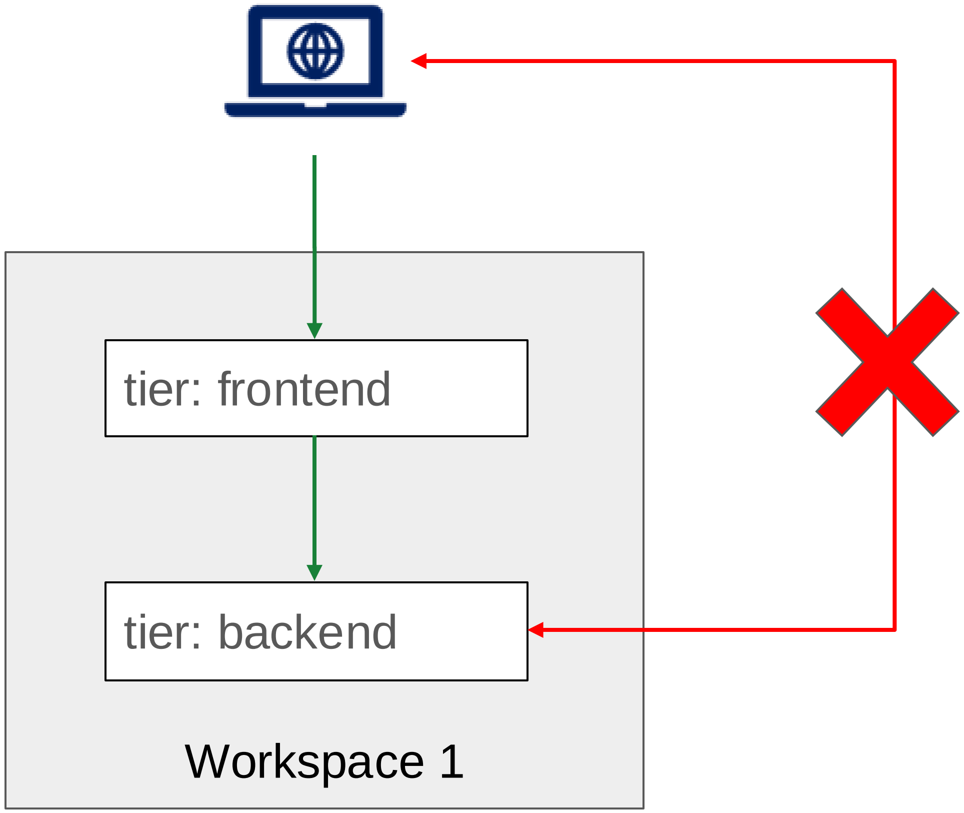
“guardrails & automation”

Is “X” → “Y” allowed?

Is “Z” → “X” allowed?

Kyverno

Tiers: Application Segmentation

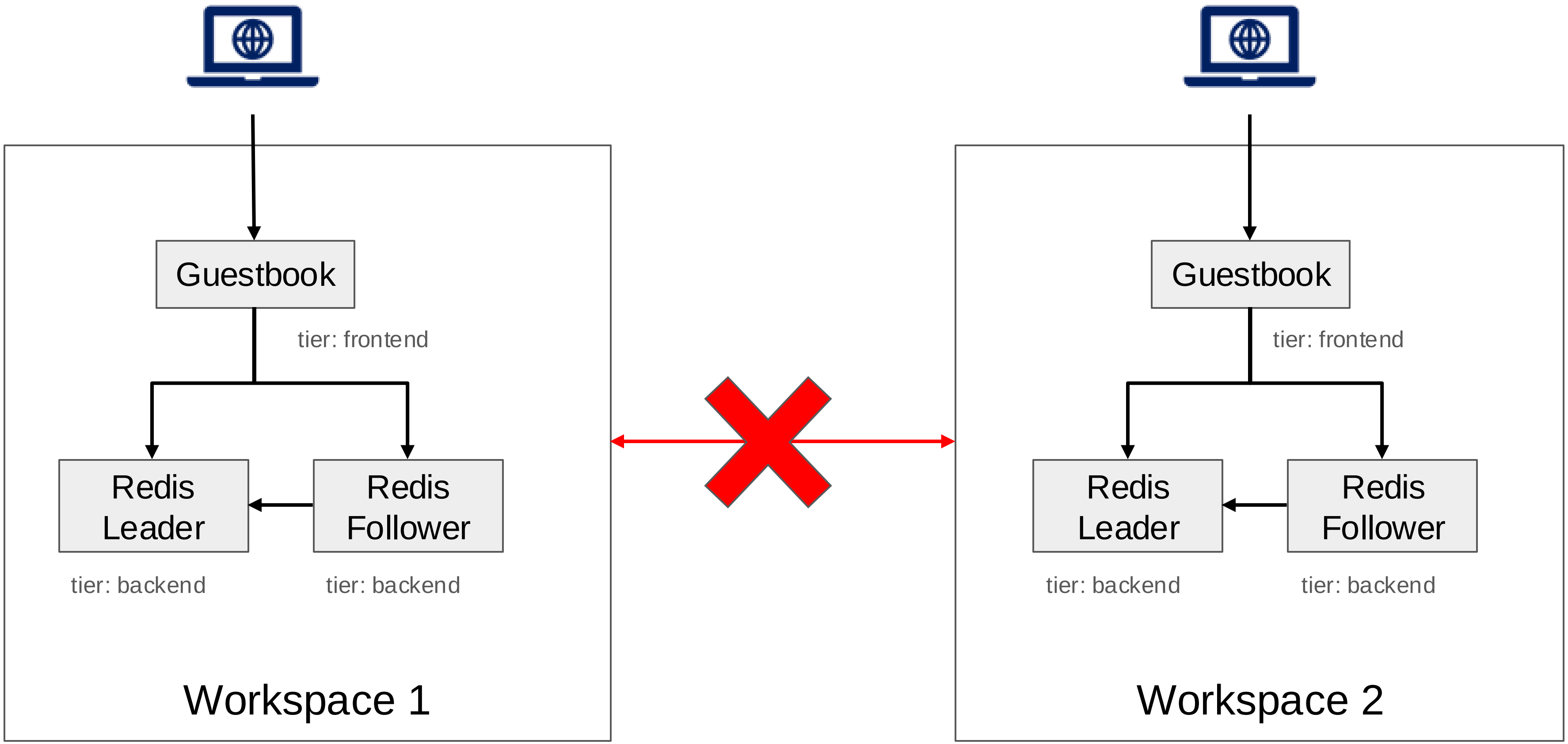


tier: frontend

tier: backend

Workspace 1

Tiers: Workspace Segmentation



Kyverno Policies

1. Require a `workspace` and `tier` labels for each namespace
2. Automatically add `workspace` and `tier` labels to pods
3. Generate network policies based namespace labels:
    - allow-dns-traffic
    - allow-ns-traffic
    - allow-workspace-traffic
4. Restrict images by tier e.g. only allow `redis` image in the `backend` tier
5. Do not allow egress traffic from `backend` tier
6. Do not allow arbitrary ingress traffic to `backend` tier

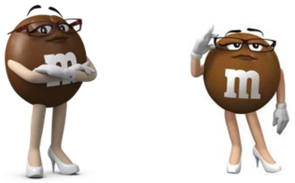
https://github.com/nirmata/kcna24-multi-tenancy-micro-segmentation/settings

Conclusion

Micro-Segmentation & Multi-Tenancy

The Brown M&Ms of Platform Engineering

1. Platforms enable agility via standardization
2. Multi-tenancy and Micro-segmentation are essential building blocks for K8s platforms
3. Cilium and Kyverno provide a powerful combination for delivering secure self-service



KubeCon | CloudNativeCon

North America 2024