We built a GKE platform

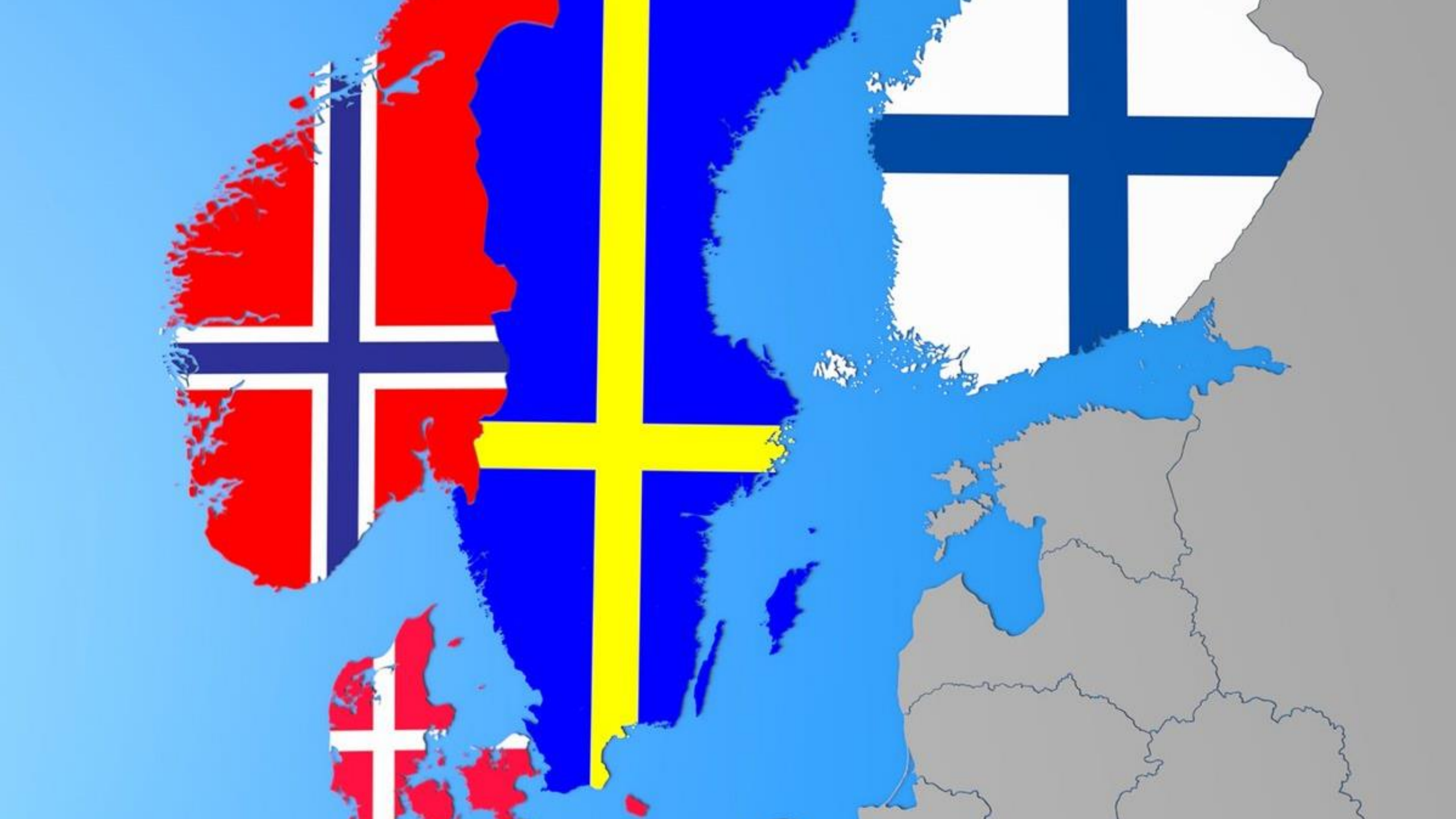
With only 2 engineers and lots of coffee 

telenor

nova

cloud.telenor.no

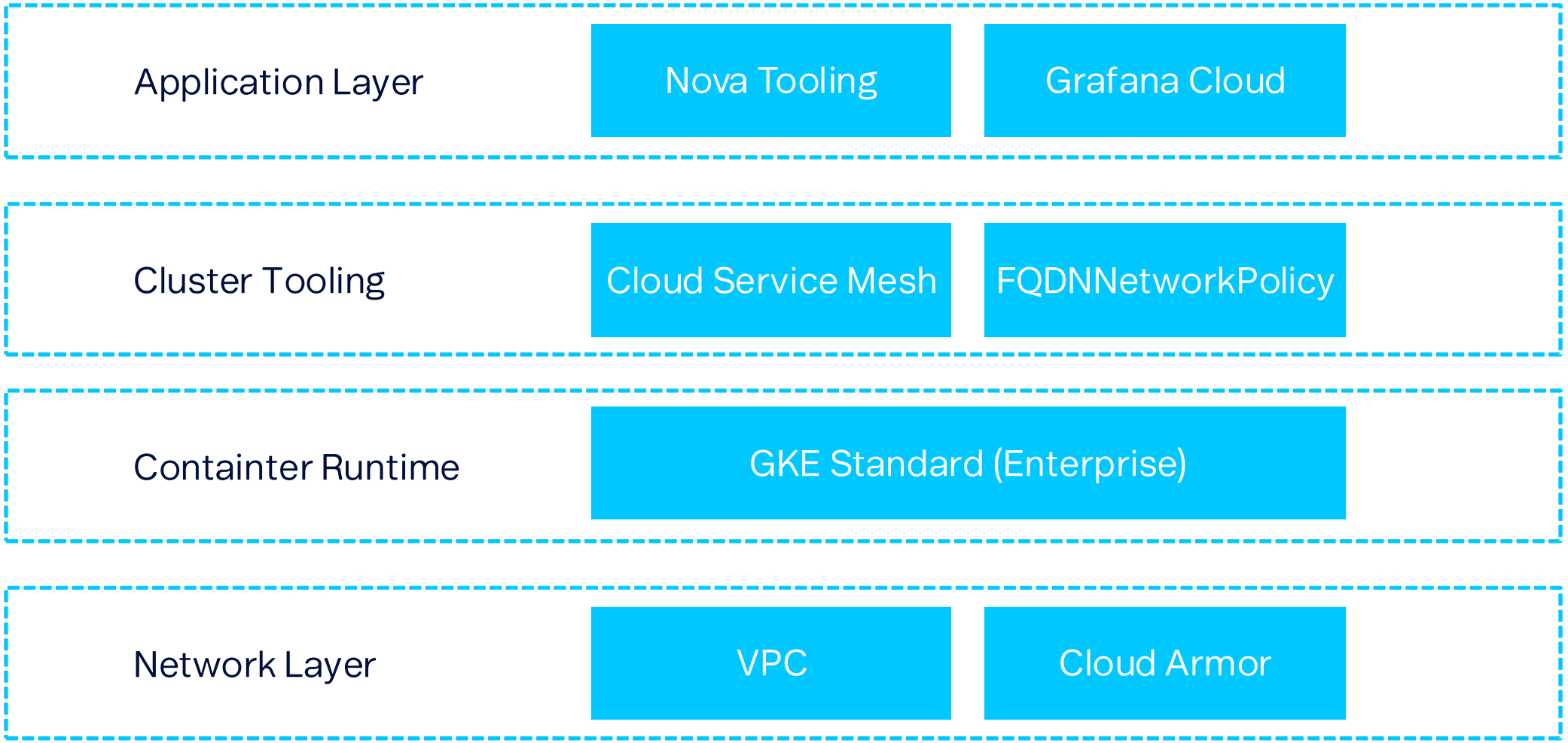
Eline Henriksen
Lead Product Lead
Telenor - Cloud Platforms



We’re a small team

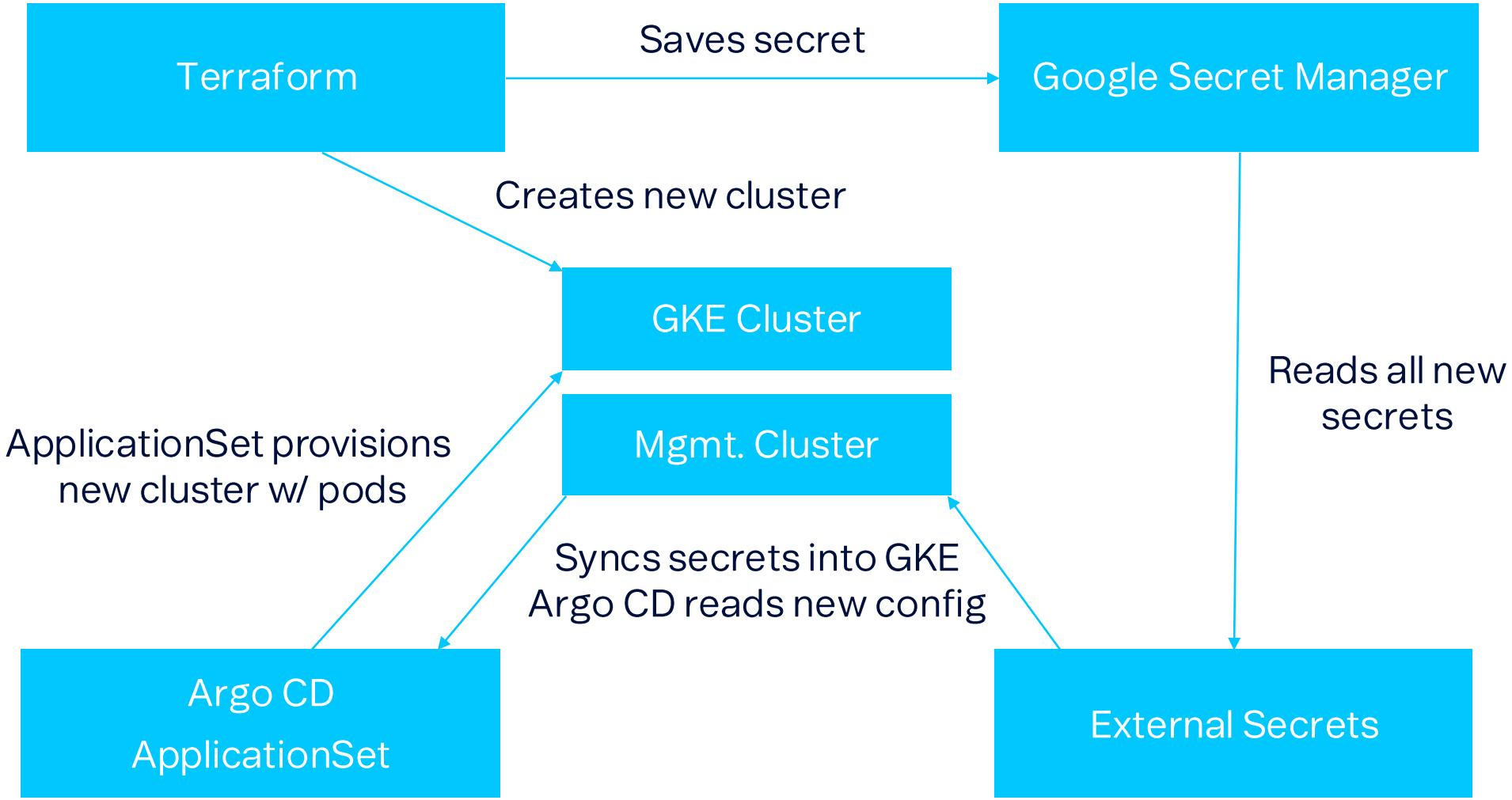
Can we still deliver world-class developer experience on GKE?

telenor



Automation is key

telenor

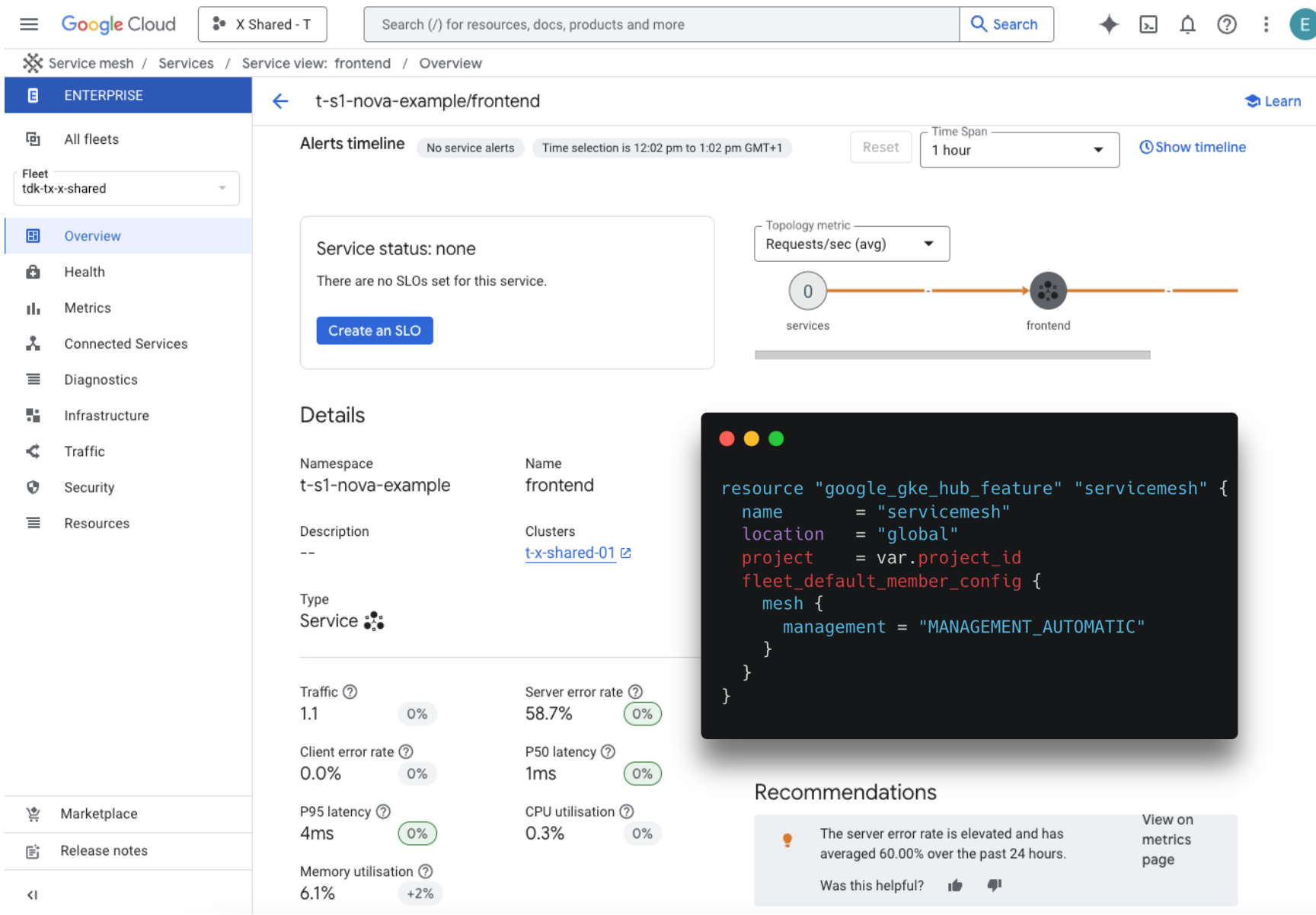


Managed offerings

telenor

Cloud Service Mesh

- DA mandates mTLS
- We have (not great) experience with Istio
- We wanted as managed an offering as possible - Cloud Service Mesh!
- CSM has two modes, we use Traffic Director
- Largely one-"click" install
- In theory should make multi-cluster mesh easy too (not tested)
- GKE Sandbox is not friends with CSM TD



Use built-in features

telenor

FQDNNetwork Policy

- GKE Dataplane v2 is based on Cilium
- Cilium has L7 NetworkPolicy
- GKE made their own, FQDNNetworkPolicy
- Works well
- GKE also provides CiliumClusterwideNetworkPolicy

```yaml
apiVersion: networking.gke.io/v1alpha1
kind: FQDNNetworkPolicy
metadata:
  name: allow-out-fqdnnp
spec:
  podSelector:
    matchLabels:
      app: curl-client
  egress:
  - matches:
    - pattern: "*.yourdomain.com"
    - name: "www.google.com"
    ports:
    - protocol: "TCP"
      port: 443
```

Don’t write code when you don’t have to

telenor

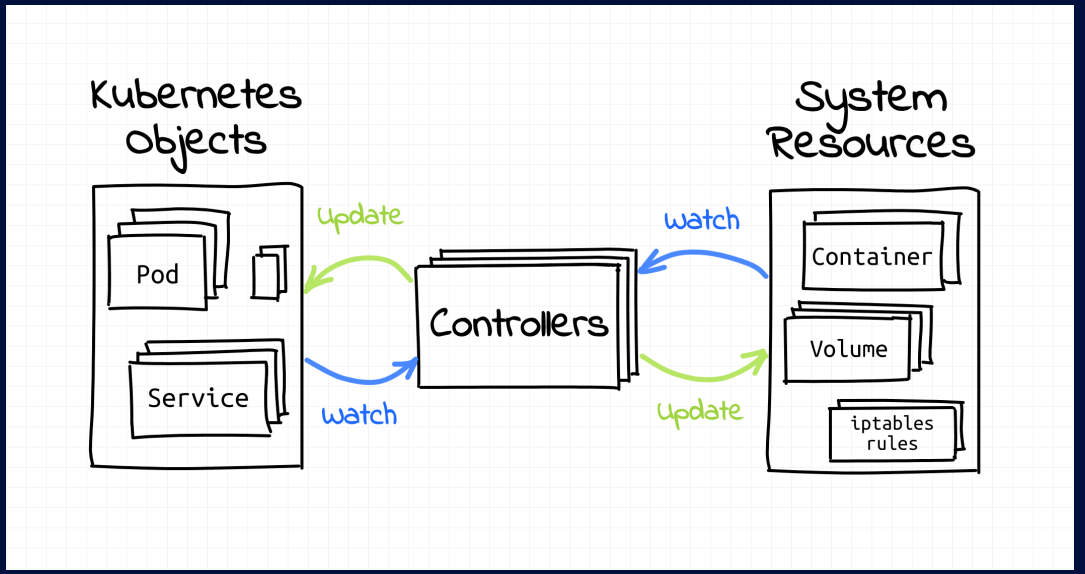
controller-runtime & operator pattern

```go
// Setup a new controller to reconcile ReplicaSets
entryLog.Info( msg: "Setting up controller")
c, err := controller.New( name: "foo-controller", mgr, controller.Options{
    Reconciler: &reconcileReplicaSet{client: mgr.GetClient(), log: log.WithName( name: "reconciler")},
})
if err != nil {
    entryLog.Error(err,  msg: "unable to set up individual controller")
    os.Exit( code: 1)
}

// Watch ReplicaSets and enqueue ReplicaSet object key
if err := c.Watch(&source.Kind{Type: &appsv1.ReplicaSet{}}, &handler.EnqueueRequestForObject{}); err != nil {
    entryLog.Error(err,  msg: "unable to watch ReplicaSets")
    os.Exit( code: 1)
}

// Watch Pods and e
if err := c.Watch(&
    &handler.Enqueu
    entryLog.Error(
    os.Exit( code: 1
}
```

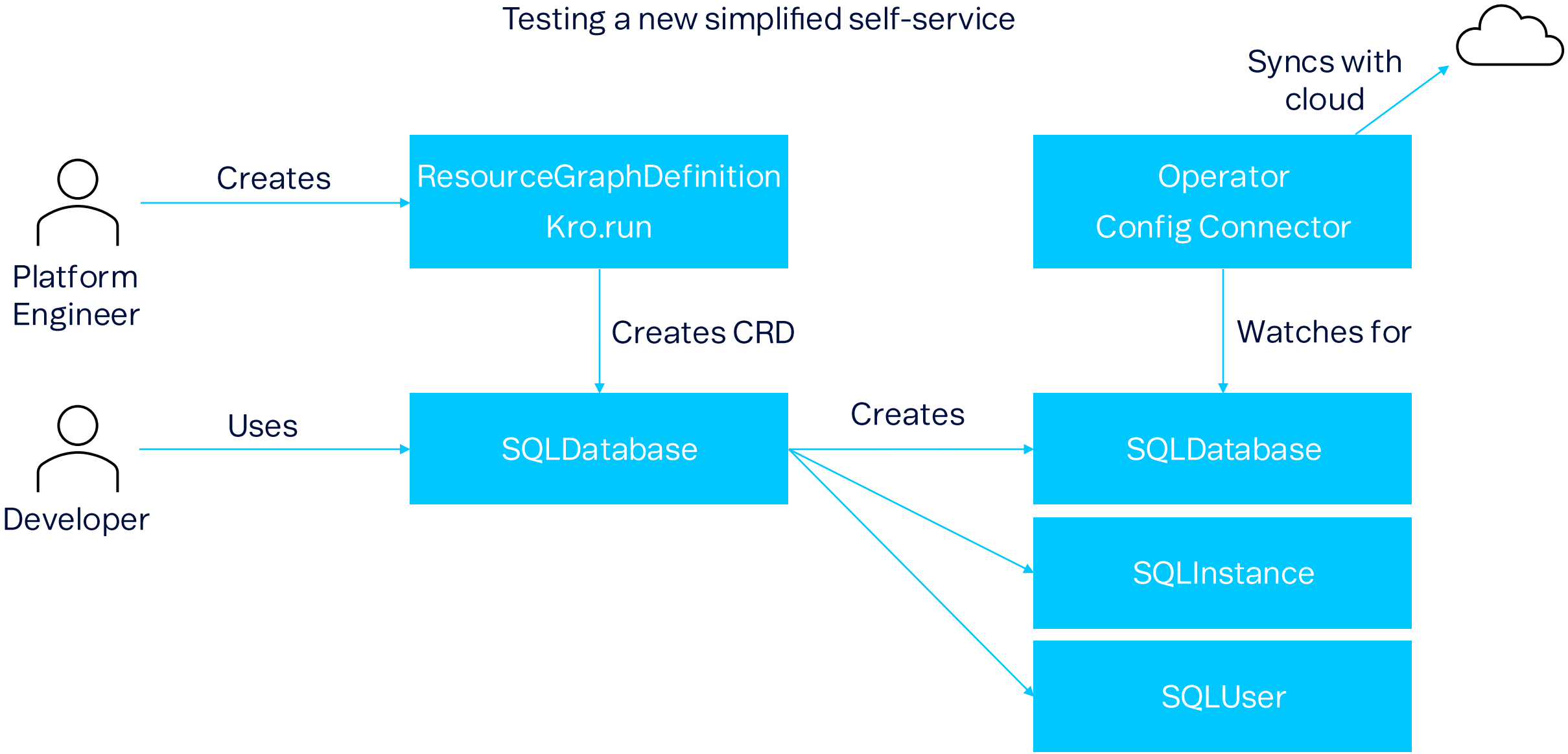
kubernetes-sigs/
controller-runtime

Repo for the controller-runtime subproject of kubebuilder (sig-apimachinery)

352 Contributors · 15 Used by · 3k Stars · 1k Forks



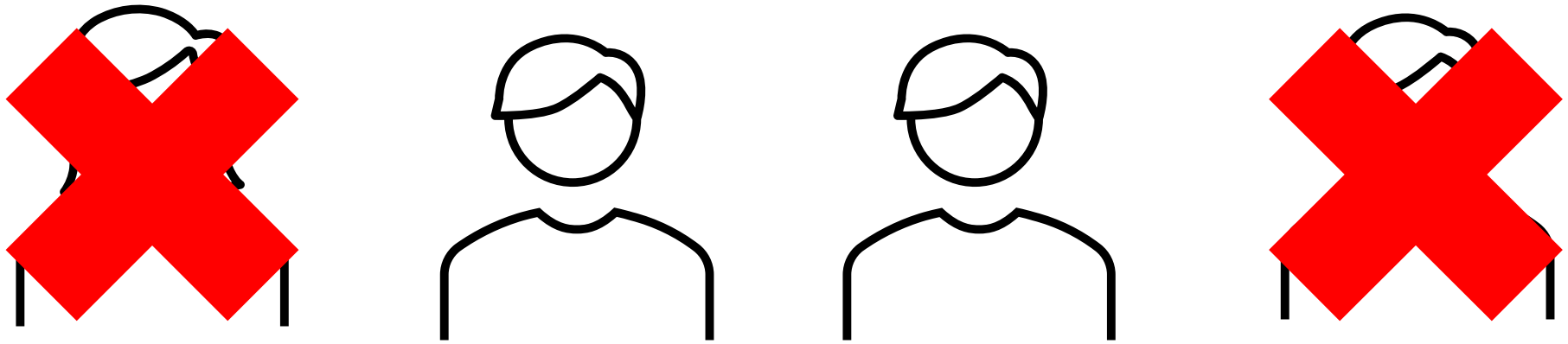
Testing a new simplified self-service



Lessons learned

telenor

Biggest challenge – Attrition risk



Promoted | Transferred | Hired | Quit

Lessons learned

- Attrition risk manageable if you keep complexity low and use industry standard tools
- Build as little as possible yourself
- Use managed services
- Use built-in Kubernetes stuff (VAP, PSS)
- Automation is key
- Consider Autopilot

And finally:
It’s possible to run a serious GKE operation with a handful of people if you play your cards right!

Thanks!

telenor