



KubeCon



CloudNativeCon

Europe 2026

#KubeCon #CloudNativeCon

Disaster Resilient Trino on Kubernetes: Multi-Cluster Setup with Karmada & Trino Gateway

Antoine Marthey, Senior Software Engineer

Sung Yun, Team Lead, Engineering Team Lead

Managed Trino & Iceberg Platform Team @ Bloomberg



Disaster Resilient Trino on Kubernetes: Multi-Cluster Setup with Karmada & Trino Gateway

Engineering

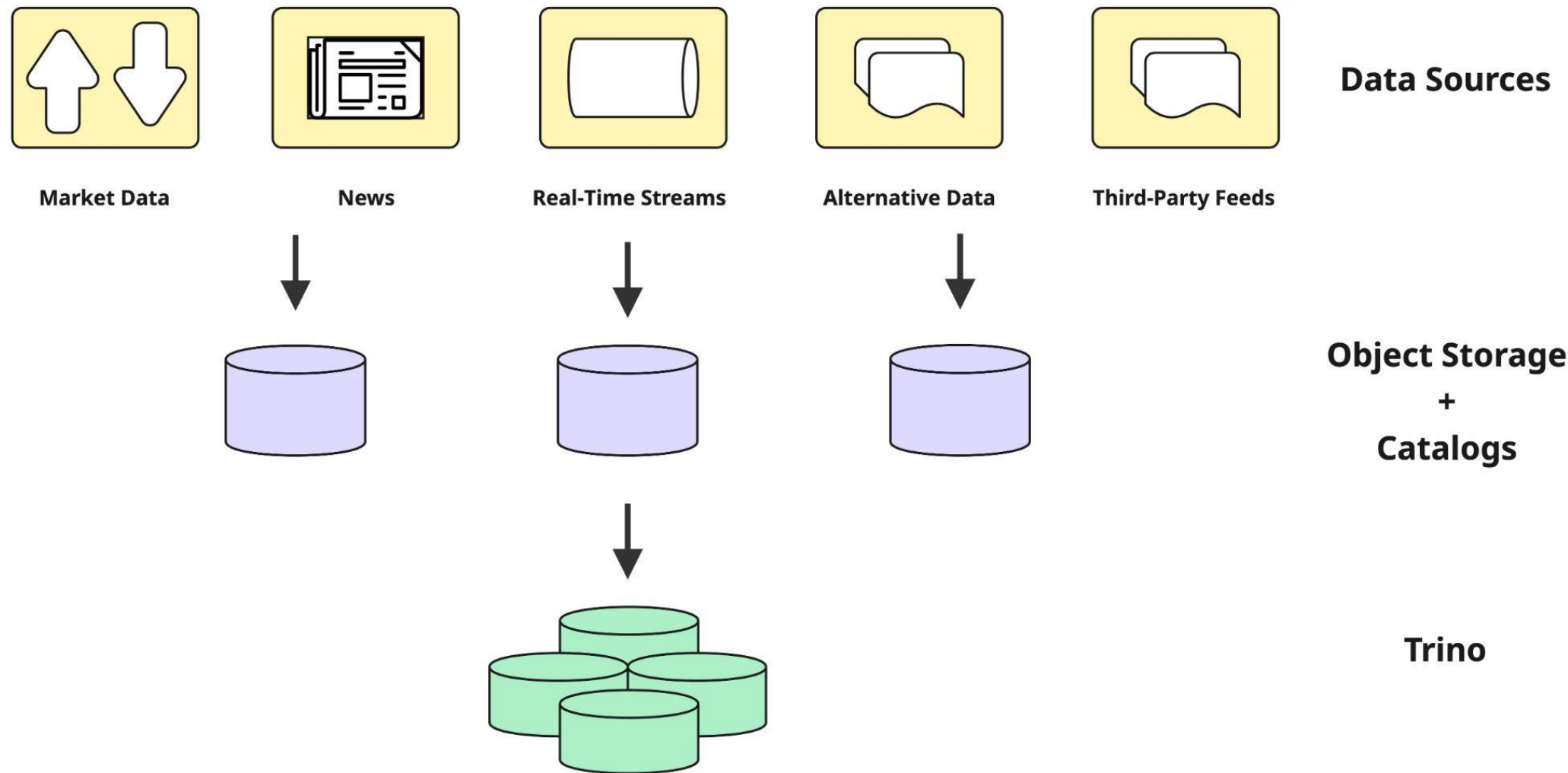
Bloomberg

KubeCon + CloudNativeCon Europe 2026
March 25, 2026

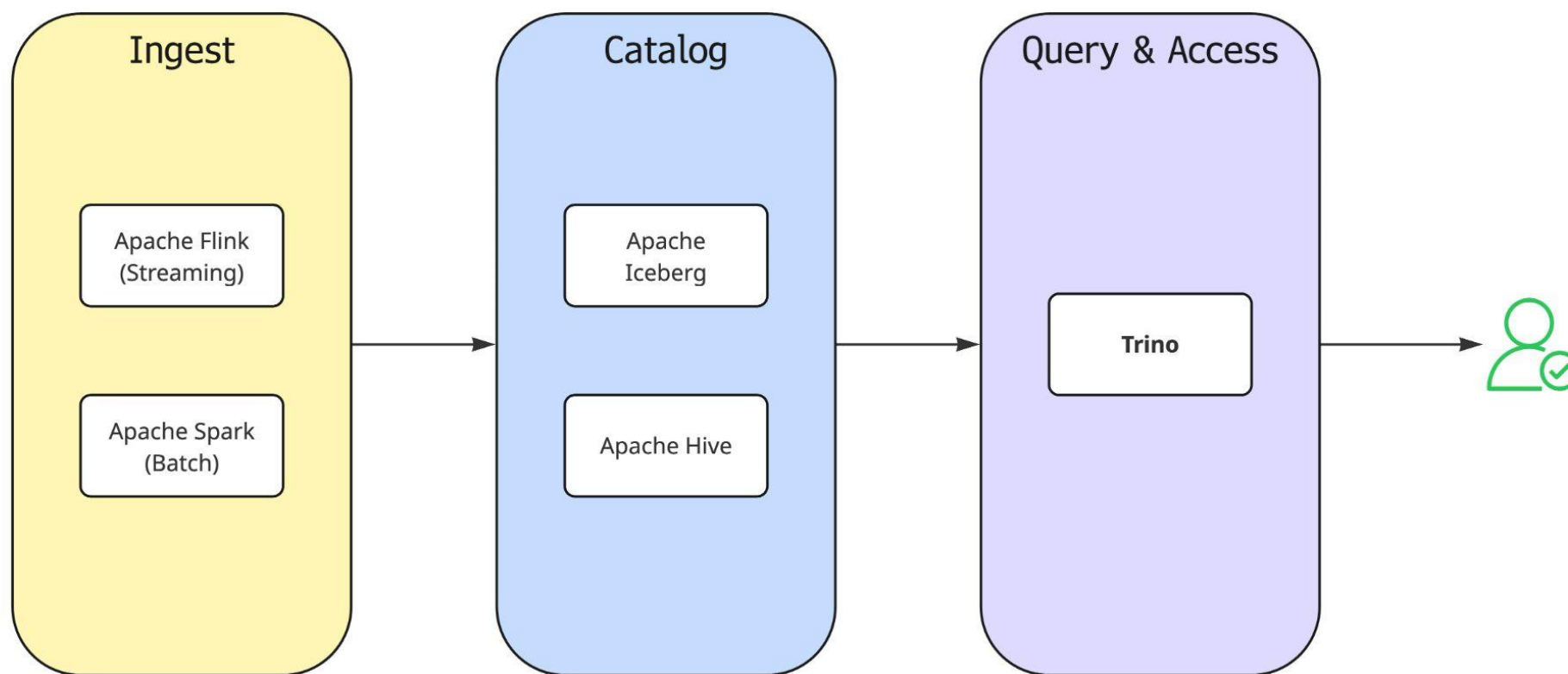
Antoine Marthey, Senior Software Engineer
Sung Yun, Team Lead, Engineering Team Lead
Managed Trino & Iceberg Platform Team, Bloomberg

TechAtBloomberg.com

The Data Environment at Bloomberg



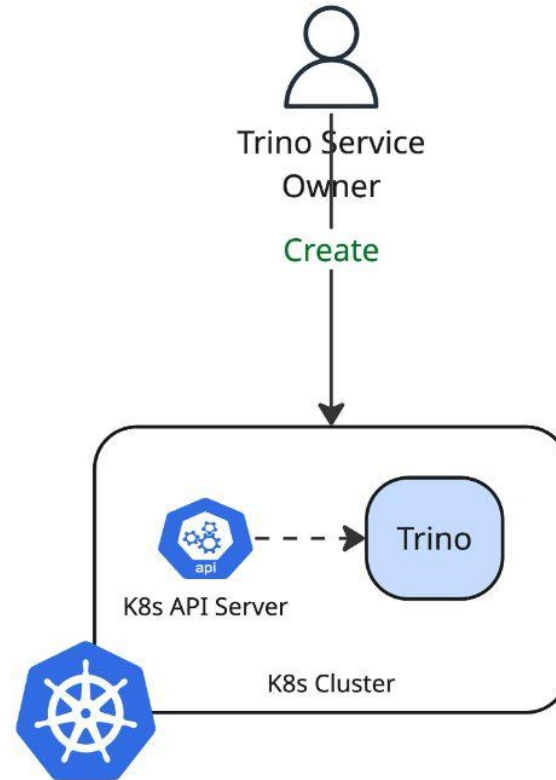
Trino's Role at Bloomberg



Single-Cluster Architecture: Personas



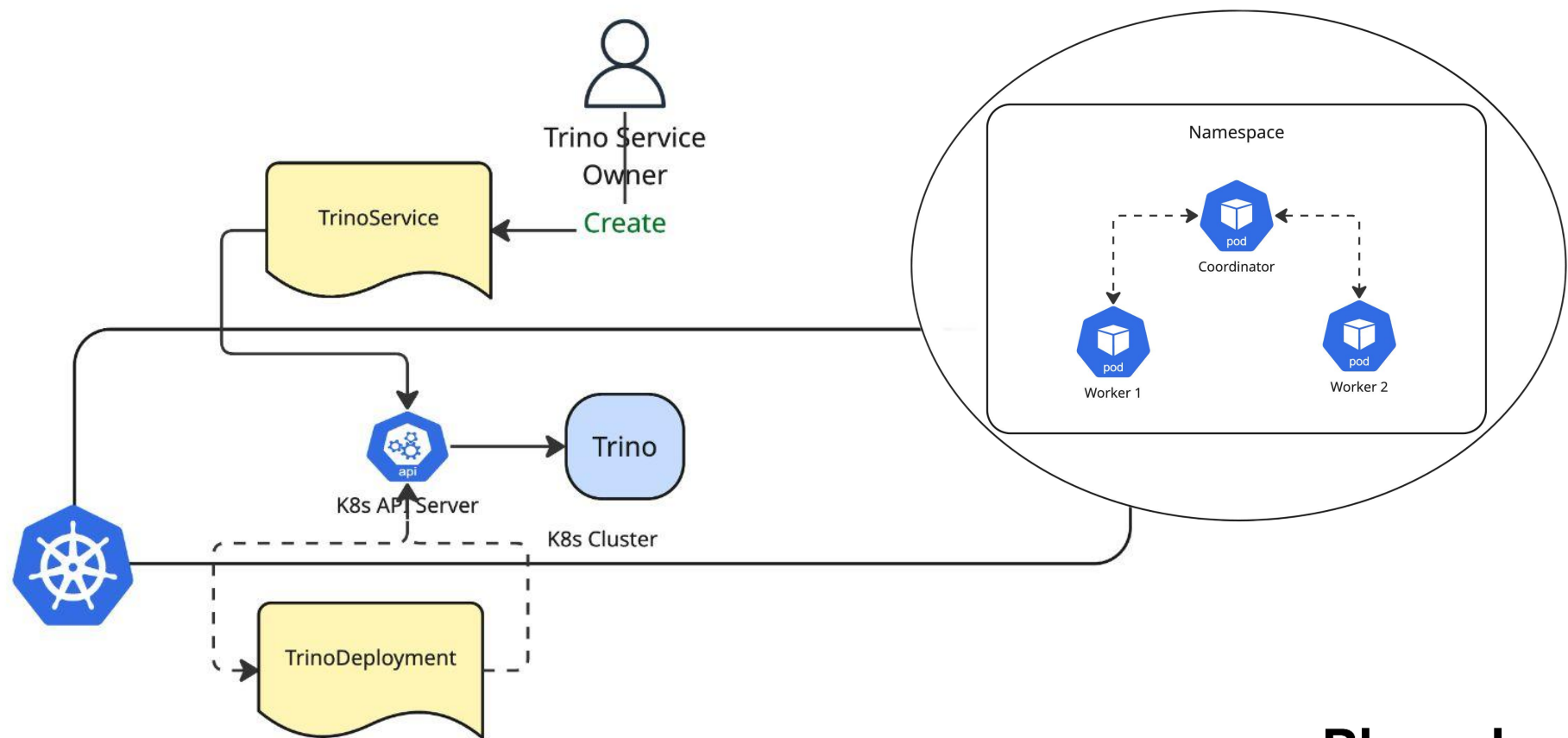
Single-Cluster Architecture: Creating a TrinoDeployment



Bloomberg

Engineering

Single-Cluster Architecture: Creating a TrinoDeployment

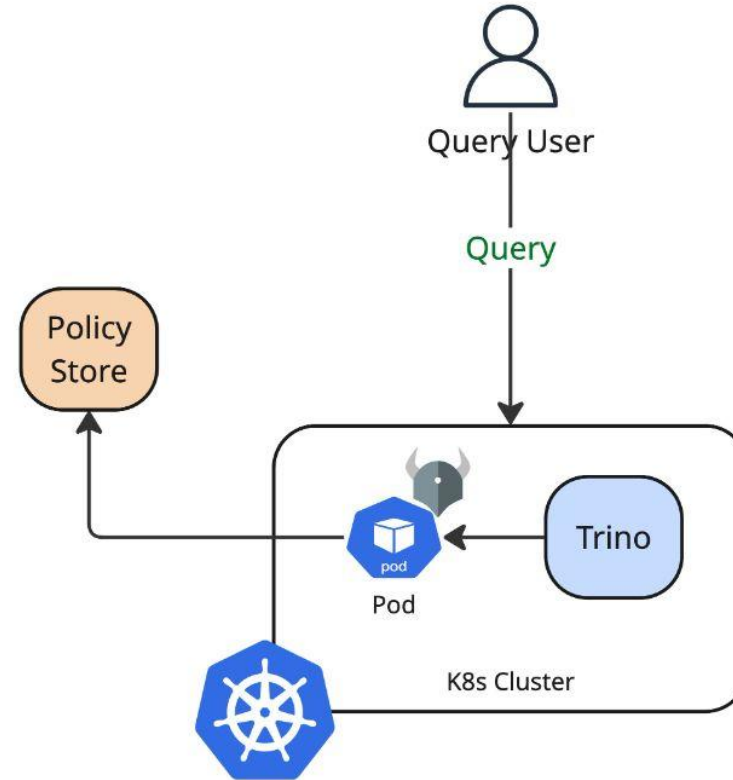


TrinoService → TrinoDeployment

```
apiVersion: trino.bloomberg.com/v1alpha1
kind: TrinoService
metadata:
  name: myts
  namespace: myns
spec:
  catalogs:
    - name: somecatalog
      namespace: dataowner
  resources:
    singleQueryLimitPercentage: 100
    totalMemory: 64Gi
status:
  ...
  endpoints:
    - name: query
      values:
        -
https://myts-myns.region1.trino.com/v1/stat
ement
```

```
apiVersion: trino.bloomberg.com/v1alpha1
kind: TrinoDeployment
metadata:
  annotations:
    propagationpolicy.karmada.io/name: trinodeployment-propagation-policy
    propagationpolicy.karmada.io/namespace: myns
    resourcebinding.karmada.io/name: myts-trinodeployment
    resourcebinding.karmada.io/namespace: myns
    resourcetemplate.karmada.io/generation: "1"
  resourceTemplate.karmada.io/managed-annotations:
    propagationpolicy.karmada.io/name: propagationpolicy.karmada.io/namespace: resourcebinding.karmada.io/name: resourcebinding.karmada.io/namespace: resourcetemplate.karmada.io/generation: resourcetemplate.karmada.io/managed-annotations:
  notations: resourcetemplate.karmada.io/managed-labels: resourcetemplate.karmada.io/uid: work.karmada.io/conflict-resolution: work.karmada.io/name: work.karmada.io/namespace
  resourceTemplate.karmada.io/managed-labels: karmada.io/managed: propagationpolicy.karmada.io/permanent-id: resourcebinding.karmada.io/permanent-id: trino.bloomberg.com/trinoservice-name: work.karmada.io/permanent-id
  resourceTemplate.karmada.io/uid: 5b3d754d-b671-47db-947f-87dca0edc60
  work.karmada.io/conflict-resolution: abort
  work.karmada.io/name: myts-5bdf69494b
  work.karmada.io/namespace: karmada-mys
  creationTimestamp: "2025-12-16T19:51:29Z"
  generation: 1
  labels:
    karmada.io/managed: "true"
  propagationpolicy.karmada.io/permanent-id: 1a5d8ee5-77ab-43cd-07d7-cd691356e447
  resourcebinding.karmada.io/permanent-id: 8da17719-5534-44fa-aebc-8ba0e5db39b3
  trino.bloomberg.com/trinoservice-name: myts
  work.karmada.io/permanent-id: 5dc9b1ee-0941-4ac3-b513-b475589f306e
  name: myts
  namespace: myns
  resourceVersion: "1246558016"
  uid: bba79580-7c5b-48b6-abe1-9b9054fc619d
  spec:
    coordinator:
      additionalAnnotations:
        sidecar.istio.io/proxyCPU: 400m
        sidecar.istio.io/proxyCPULimit: 400m
        sidecar.istio.io/proxyMemory: 512Mi
        sidecar.istio.io/proxyMemoryLimit: 512Mi
      additionalConfigFiles:
        access-control.properties: |
          access-control.name=opa
          opa.policy.uri=http://opa.opa.svc:8181/v1/data/trino_single/allow
          opa.policy.batched-uri=http://opa.opa.svc:8181/v1/data/trino_batched/allow
          opa.allow-permission-management-operations=false
          opa.log-requests=true
          opa.log-responses=true
          opa.additional-context-file=/etc/trino/opa-additional-context.json
        annotations.properties: |
          trino_service_name=${ENV:TRINO_SERVICE_NAME}
          trino_service_namespace=${ENV:TRINO_SERVICE_NAMESPACE}
          environment=${ENV:CONTROLLER_ENVIRONMENT}
          guthms_config_yaml: |
            rules:
              # JMX rules for Trino
              - help: $2 bytes in $1 memory
                labels:
                  area: $1
                  name: jvm_memory_$2_bytes
                  pattern: java.lang.type=Memory<=<Heap|NonHeap|MemoryUsage>(max|init|committed|used)
                  type: GAUGE
              - help: Total Free Memory Available to JVM
                labels:
                  area: $1
                  name: jvm_memory_$1_allocated_bytes
                  pattern: java.lang.type=OperatingSystem<=<Free|Total|MemorySize
                  type: GAUGE
              - help: Number of CPU cores available
                name: jvm_os_processors
                  pattern: java.lang.type=OperatingSystem<=<AvailableProcessors
                  type: GAUGE
              - help: $1 cpu load
                name: jvm_os_$1_cpu_load
                  pattern: java.lang.type=OperatingSystem<=<(Process|System)CpuLoad
                  type: GAUGE
              - help: JVM uptime in seconds
                name: jvm_runtime_uptime_seconds
                  pattern: java.lang.type=Runtime<=<Uptime
                  type: COUNTER
                  valueFactor: 0.001
              - help: JVM $1 thread count
                name: jvm_threading_$1_threads
                  pattern: java.lang.type=Threading<=<(Daemon|Peak|Total|Started|)ThreadCount
                  type: GAUGE
              # Trino Query Metrics
              - help: Number of queries currently in the labelled state
                name: trino_query_manager_$1_queries
                  pattern: trino.execution.name=QueryManager<=<(Queued|Running|Queries
                  type: GAUGE
              - help: Total number of queries
                name: trino_query_manager_$1_queries_total
                  pattern: trino.execution.name=QueryManager<=<(Abandoned|Canceled|Completed|Failed|Started|Submitted|Queries.TotalCount
                  type: COUNTER
              - help: Total number of failures with the labelled type
                labels:
                  failureReason: $1
                  name: trino_query_manager_failures_total
                  pattern: trino.execution.name=QueryManager<=<(External|InsufficientResources|Internal|UserError|Failures.TotalCount
                  type: COUNTER
              - help: $1 time in seconds for the labelled quantile
                labels:
                  trinoQuantile: $2
                  name: trino_query_manager_$1_seconds
                  pattern: trino.execution.name=QueryManager<=<(Execution|Queued|TimeAllTimeP|id+)
                  type: GAUGE
                  valueFactor: 0.001
              # Split Level Metrics
              # TODO: Split level metrics are broken as of 02/25.
              # The issue is documented at https://github.com/trinodb/trino/pull/22914
              # Quiesce and Node Level Metrics
              - help: Number of $1 nodes observed by the failure detector
                name: trino_failure_detector_$1_nodes
                  pattern: trino.failure.detector.name=HeartbeatFailureDetector<=<(Active|Failed|Total|Count
                  type: GAUGE
              - help: $1 in bytes
                name: trino_cluster_memory_manager_$1_bytes
                  pattern: trino.cluster.memory.manager.name=ClusterMemoryManager<=<(TotalMemory|BytesTotalMemory|Reservation|UserMemory|Reservation)
                  type: GAUGE
              - help: Total available CPU cores
```

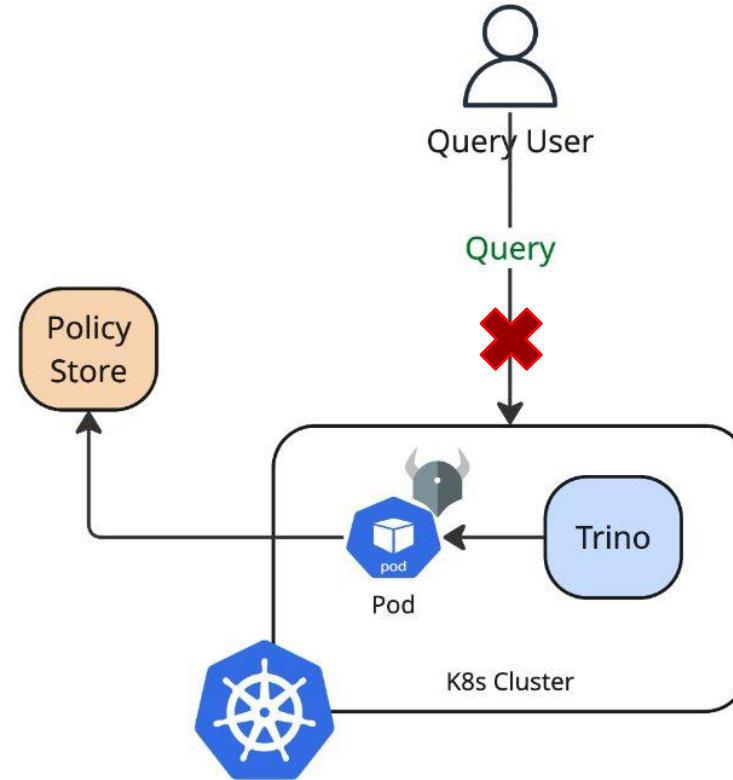
Single-Cluster Architecture: Governed Trino on Kubernetes



Bloomberg

Engineering

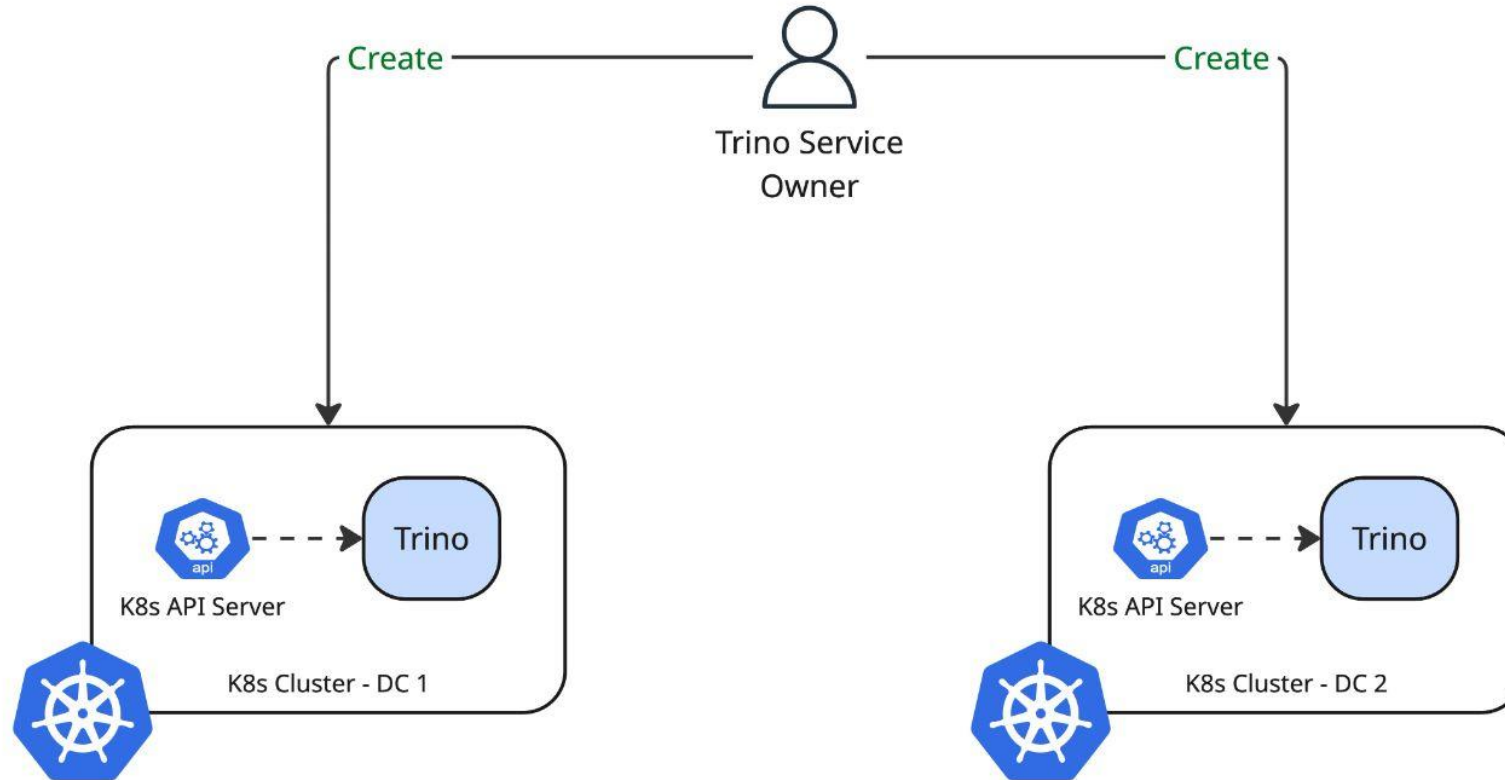
Single-Cluster Architecture: Governed Trino on Kubernetes



Bloomberg

Engineering

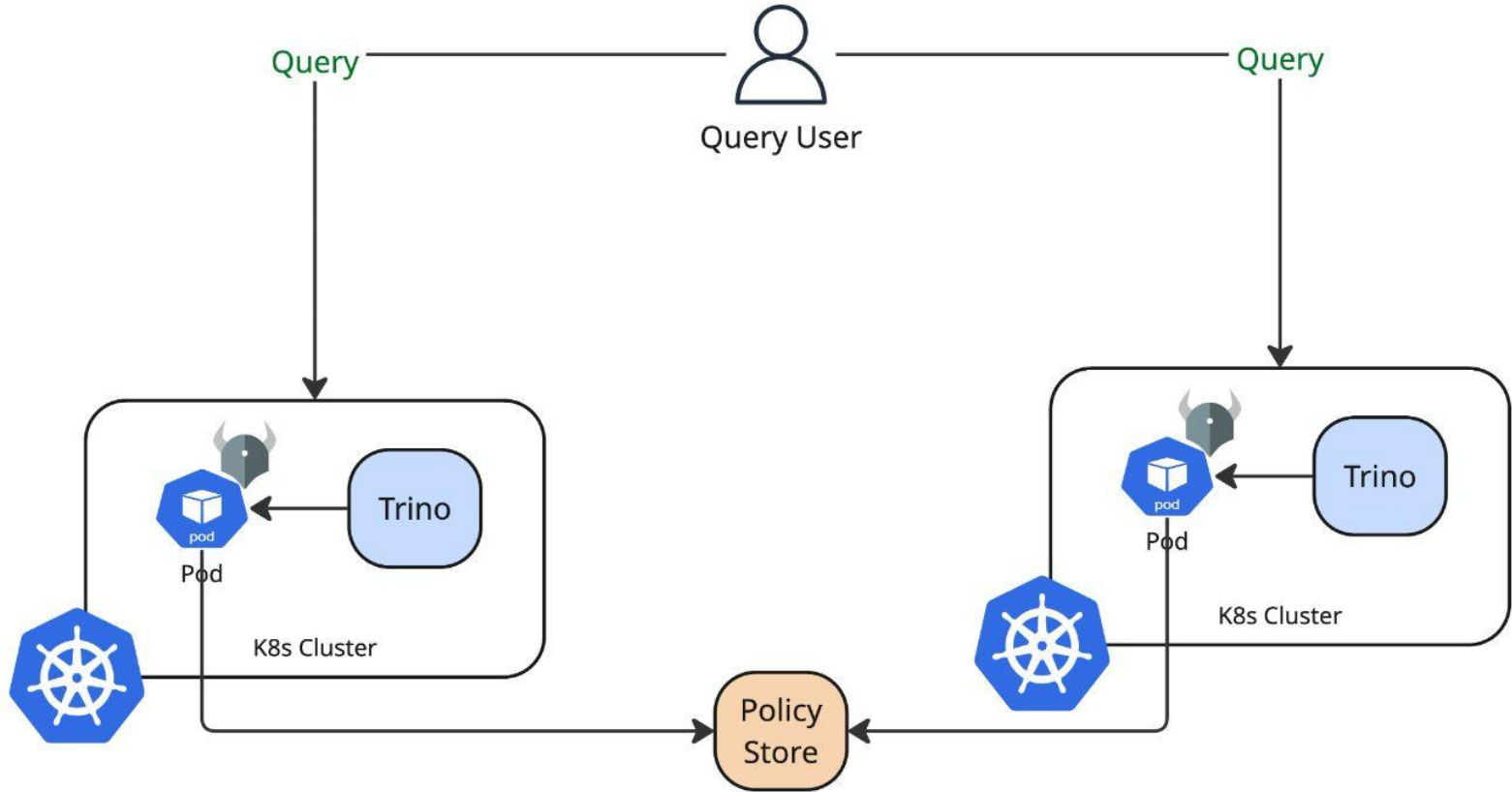
Single-Cluster Architecture: High Availability



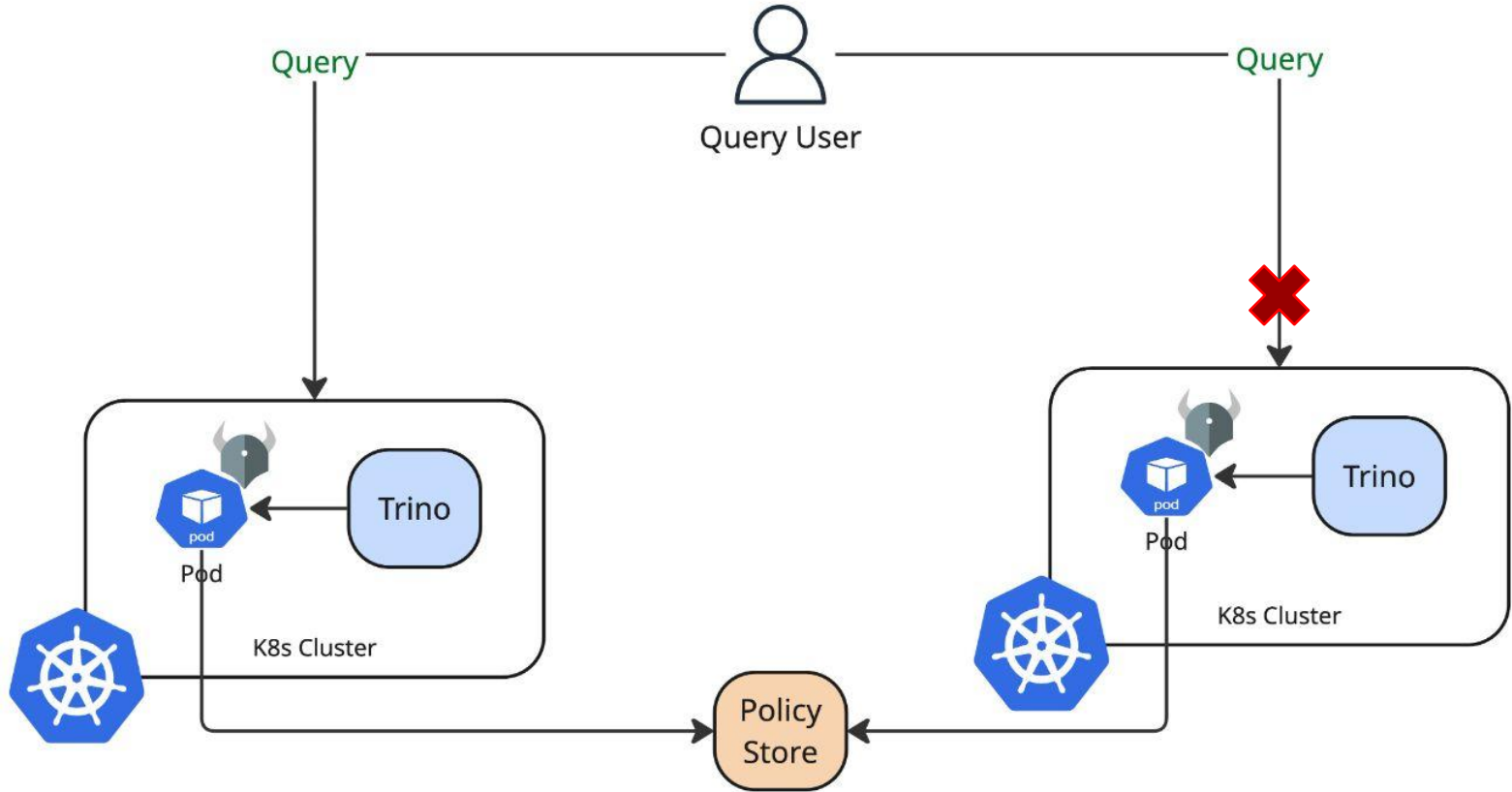
Bloomberg

Engineering

Single-Cluster Architecture: High Availability



Single-Cluster Architecture: High Availability



TrinoService CRD



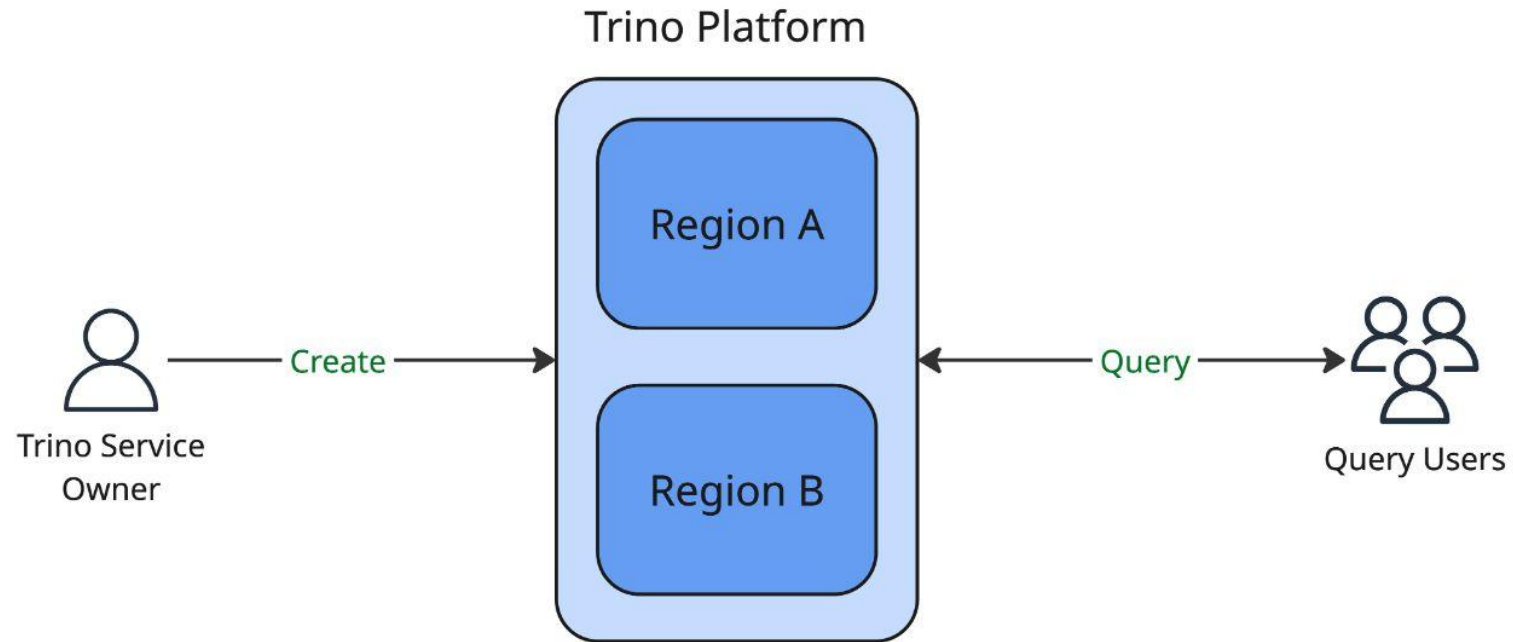
```
apiVersion: trino.bloomberg.com/v1alpha1
kind: TrinoService
metadata:
  name: myts
  namespace: myns
spec:
  catalogs:
    - name: somecatalog
      namespace: dataowner
  resources:
    singleQueryLimitPercentage: 100
    totalMemory: 64Gi
status:
  ...
  endpoints:
    - name: query
      values:
        - https://myts-myns.region1.trino.com/v1/statement
```

Multi-cluster Compatible TrinoService CRD

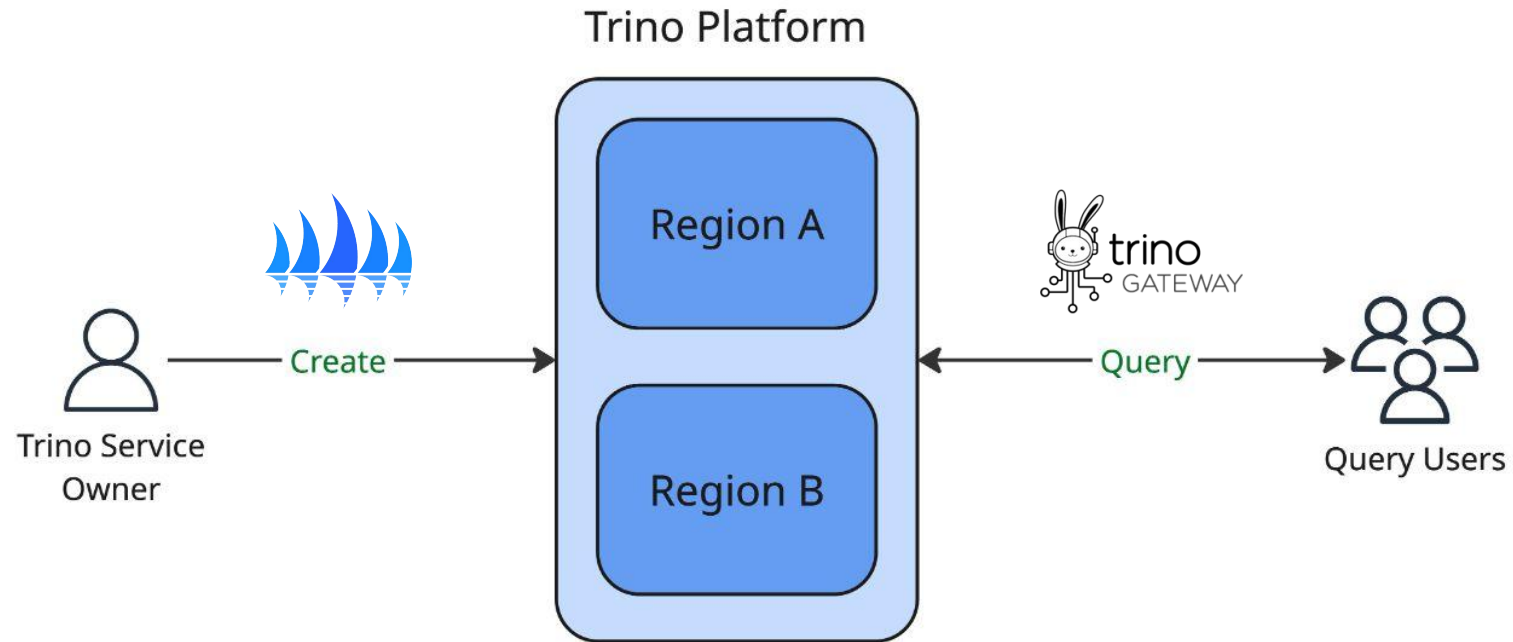


```
apiVersion: trino.bloomberg.com/v1alpha1
kind: TrinoService
metadata:
  name: myts
  namespace: myns
spec:
  catalogs:
    - name: somecatalog
      namespace: dataowner
  resources:
    singleQueryLimitPercentage: 100
    totalMemory: 64Gi
status:
  ...
  endpoints:
    - name: query
      values:
        - https://unified-endpoint.trino.com/v1/statement
```

Multi-Cluster Architecture: Disaster Resilient Trino



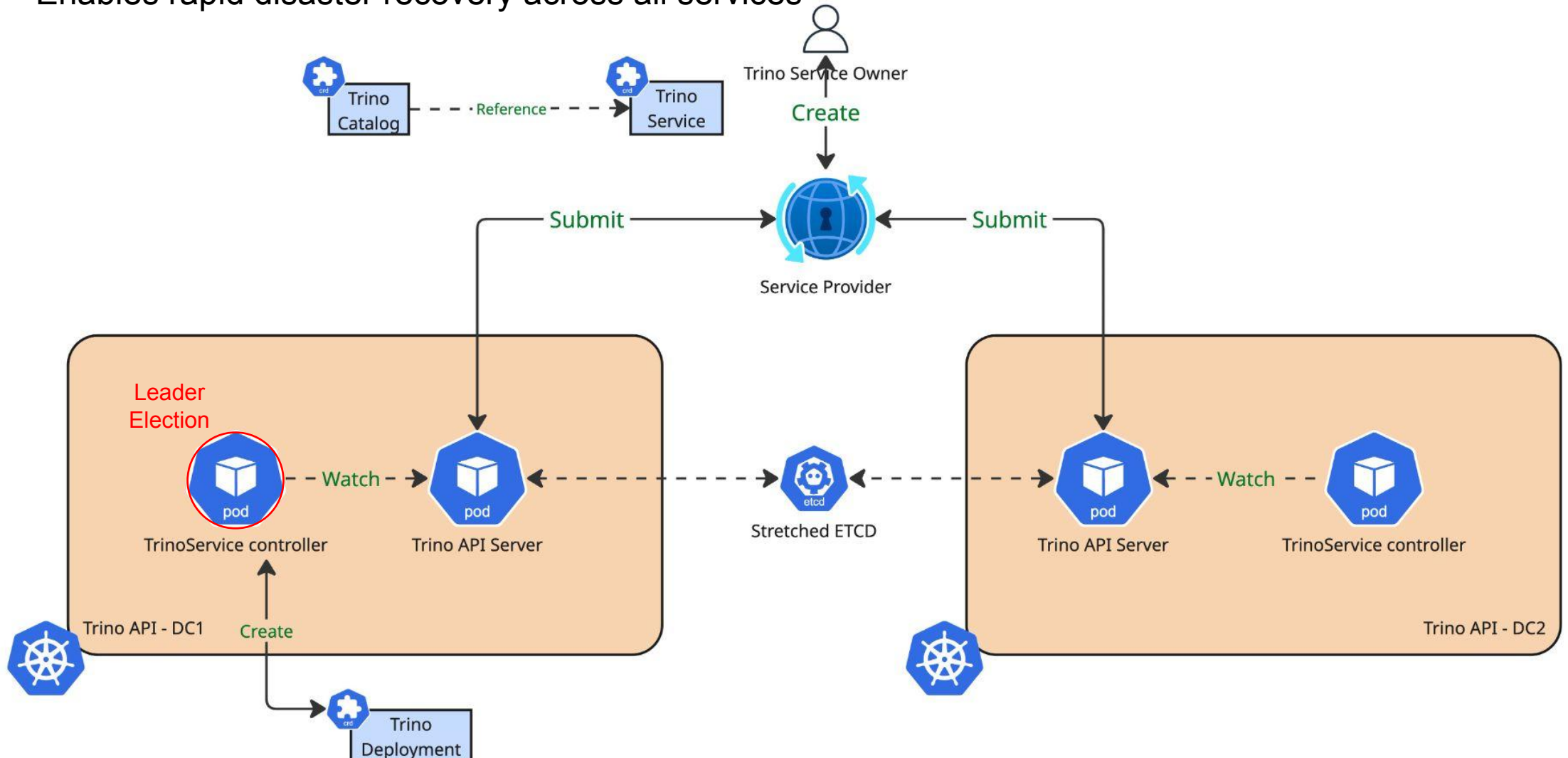
Multi-Cluster Architecture: Disaster Resilient Trino



Handoff to Antoine

Trino API with Stretched ETCD

- Single point of entry, single manifest
- Distributed, strongly consistent key-value store ensuring critical data reliability
- Enables rapid disaster recovery across all services



Karmada



Karmada is a CNCF Incubation Project

<https://karmada.io/>

TechAtBloomberg.com

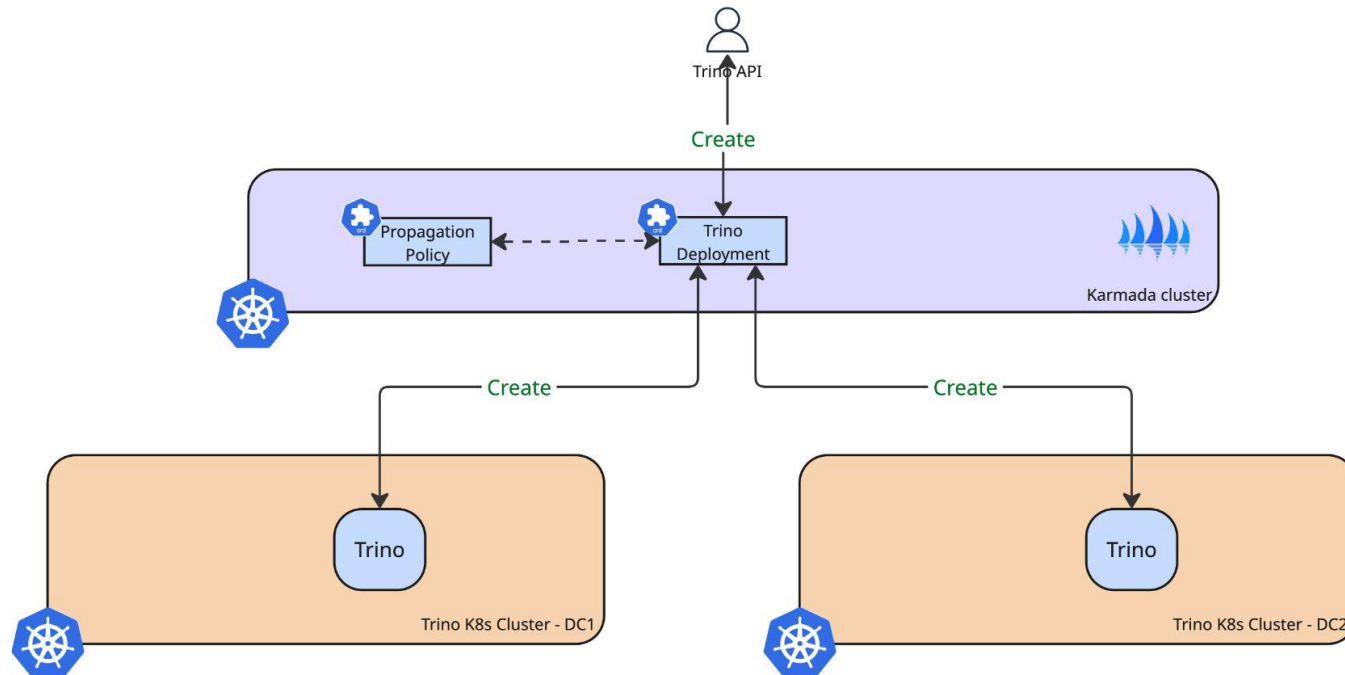
© 2026 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering

Karmada Orchestration

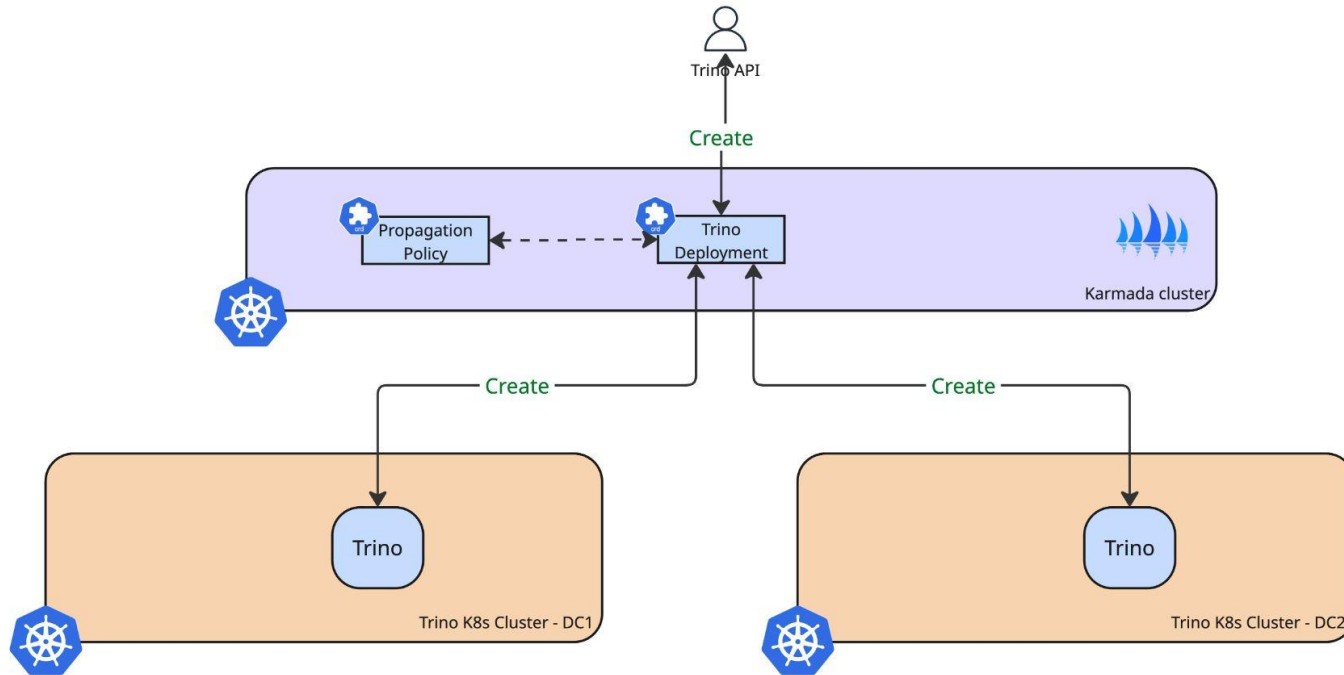
- Advanced scheduling and workload distribution capabilities
- Check for the health of TrinoServices and try to recreate them if necessary



Propagation Policy

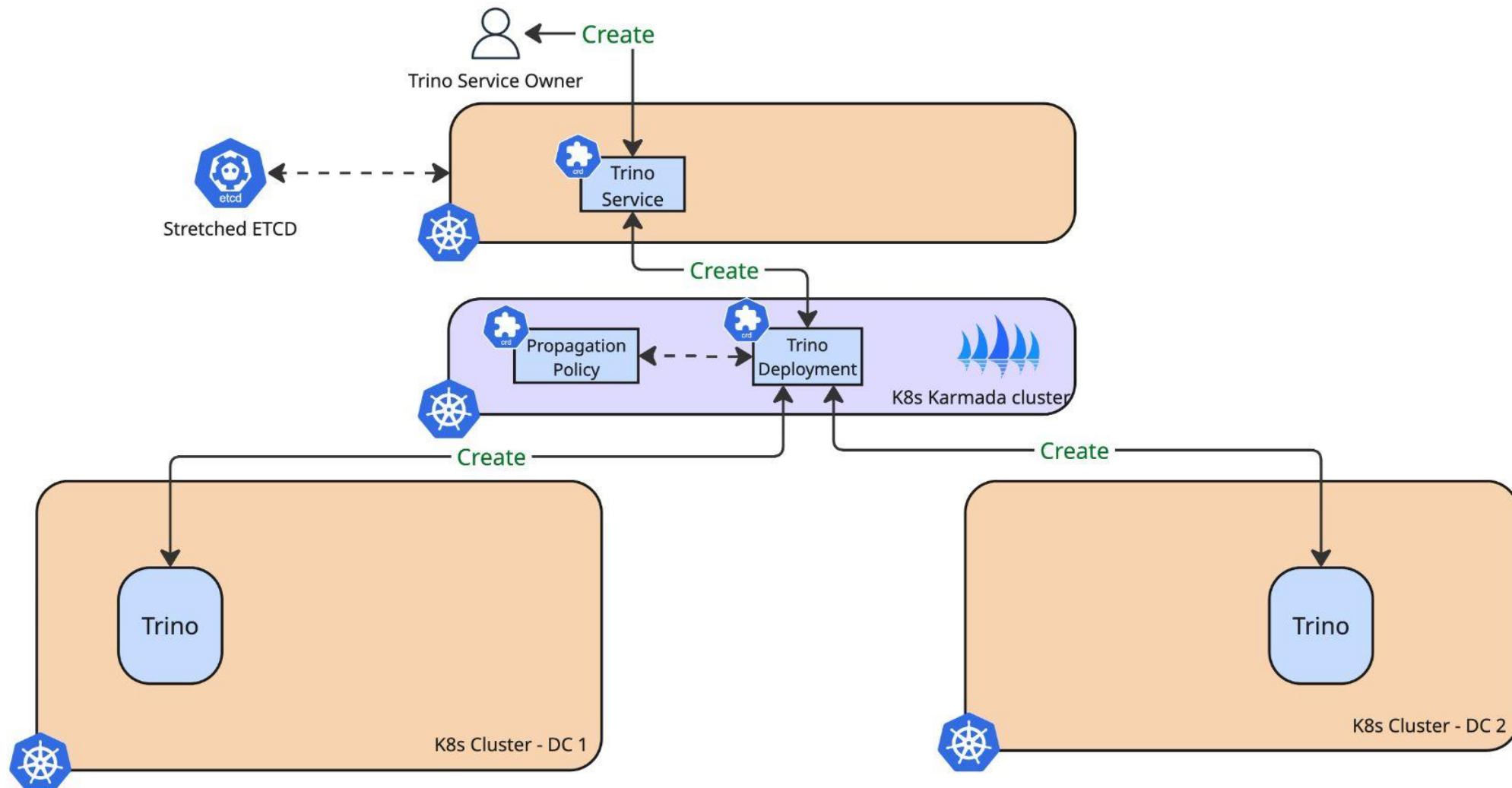
```
apiVersion: policy.karmada.io/v1alpha1
kind: PropagationPolicy
metadata:
  name: trinodeployment-propagation-policy
  namespace: myns
spec:
  conflictResolution: Abort
  placement:
    clusterTolerations:
      - effect: NoExecute
        key: cluster.karmada.io/not-ready
        operator: Exists
        tolerationSeconds: 300
      - effect: NoExecute
        key: cluster.karmada.io/unreachable
        operator: Exists
        tolerationSeconds: 300
    replicaScheduling:
      replicaSchedulingType: Duplicated
  preemption: Never
  priority: 0
  resourceSelectors:
    - apiVersion: trino.bloomberg.com/v1alpha1
      kind: TrinoDeployment
      namespace: s-cncstrino
  schedulerName: default-scheduler
```

Status Aggregation (ResourceInterpreterCustomization)

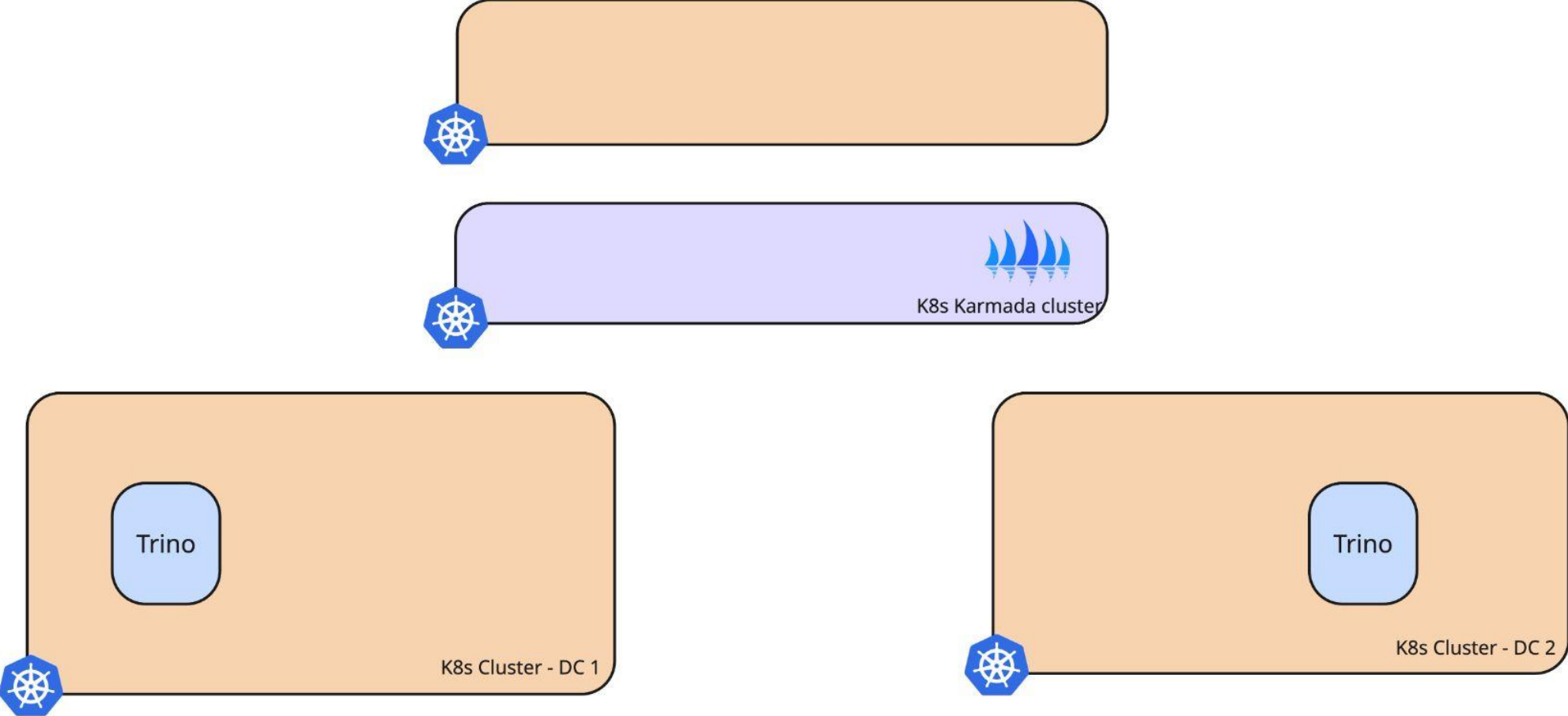


```
---
apiVersion: config.karmada.io/v1alpha1
kind: ResourceInterpreterCustomization
metadata:
  name: td-declarative-resource-interpreter
spec:
  target:
    apiVersion: trino.bloomberg.com/v1alpha1
    kind: TrinoDeployment
  customizations:
    statusAggregation:
      luaScript: >
        function AggregateStatus(desiredObj, statusItems)
          if desiredObj.status == nil then desiredObj.status = {} end
          if statusItems == nil then
            return desiredObj
          end
          for i = 1, #statusItems do
            clusterName = statusItems[i].clusterName
            if statusItems[i].status ~= nil and statusItems[i].status.conditions ~=
nil then
              for j, condition in ipairs(statusItems[i].status.conditions or {}) do
                ...
              end
            end
          end
        end
```

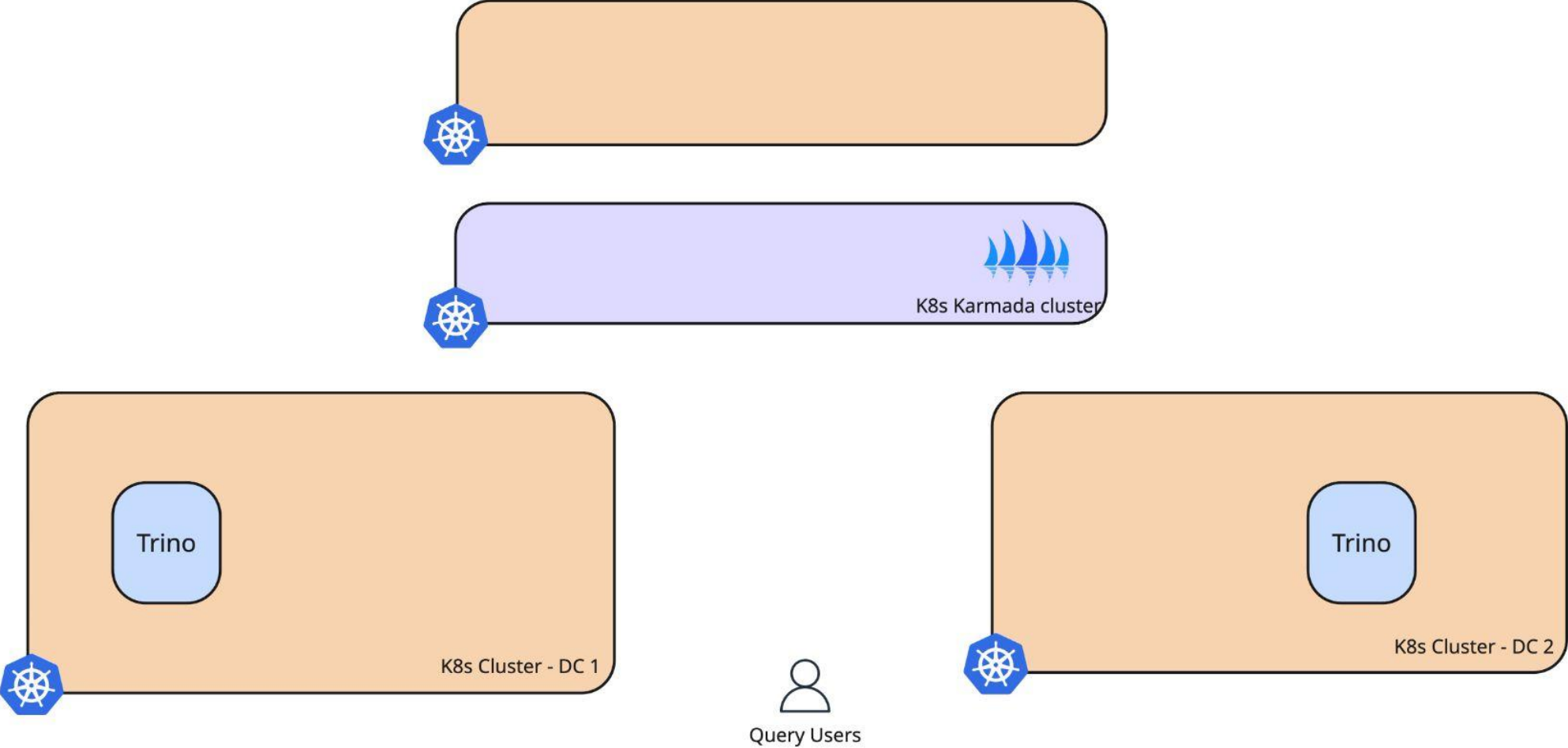
Trino Service Owner Workflow



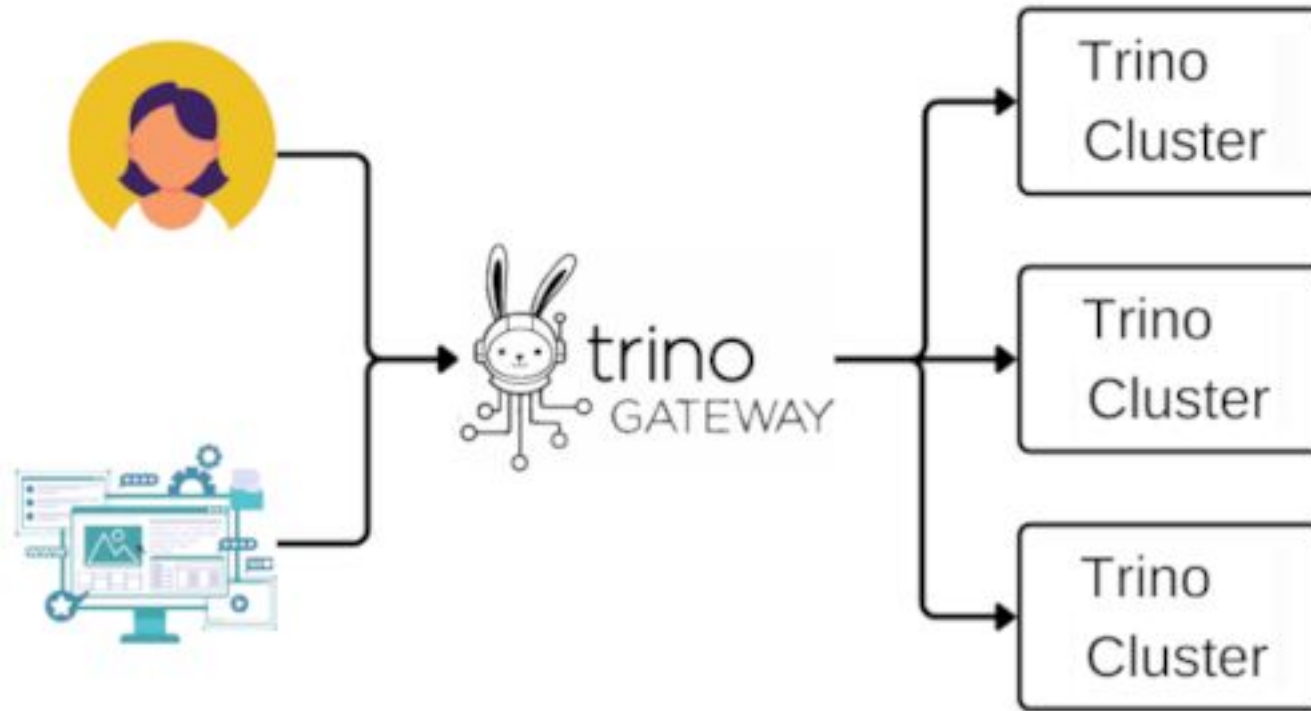
Query User Workflow



Query User Workflow



Trino Gateway



<https://trinodb.github.io/trino-gateway/>

Trino Gateway is open source software licensed under the [Apache License 2.0](#) and supported by the [Trino Software Foundation](#).

TechAtBloomberg.com

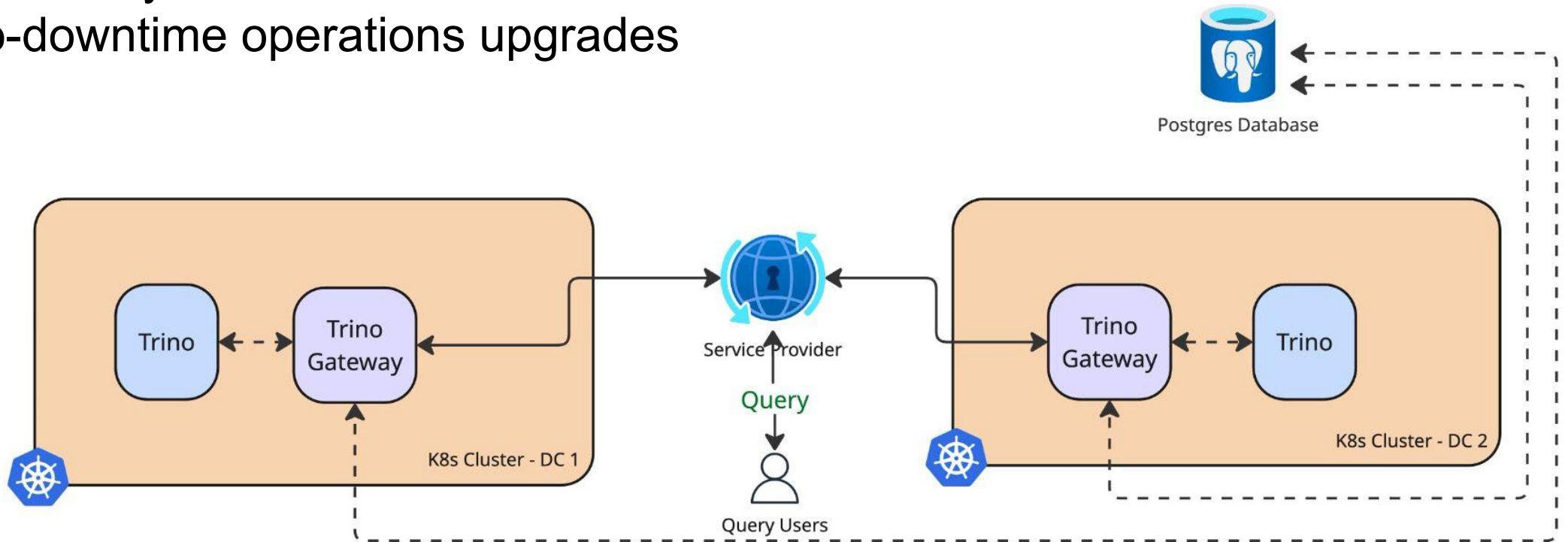
© 2026 Bloomberg Finance L.P. All rights reserved.

Bloomberg

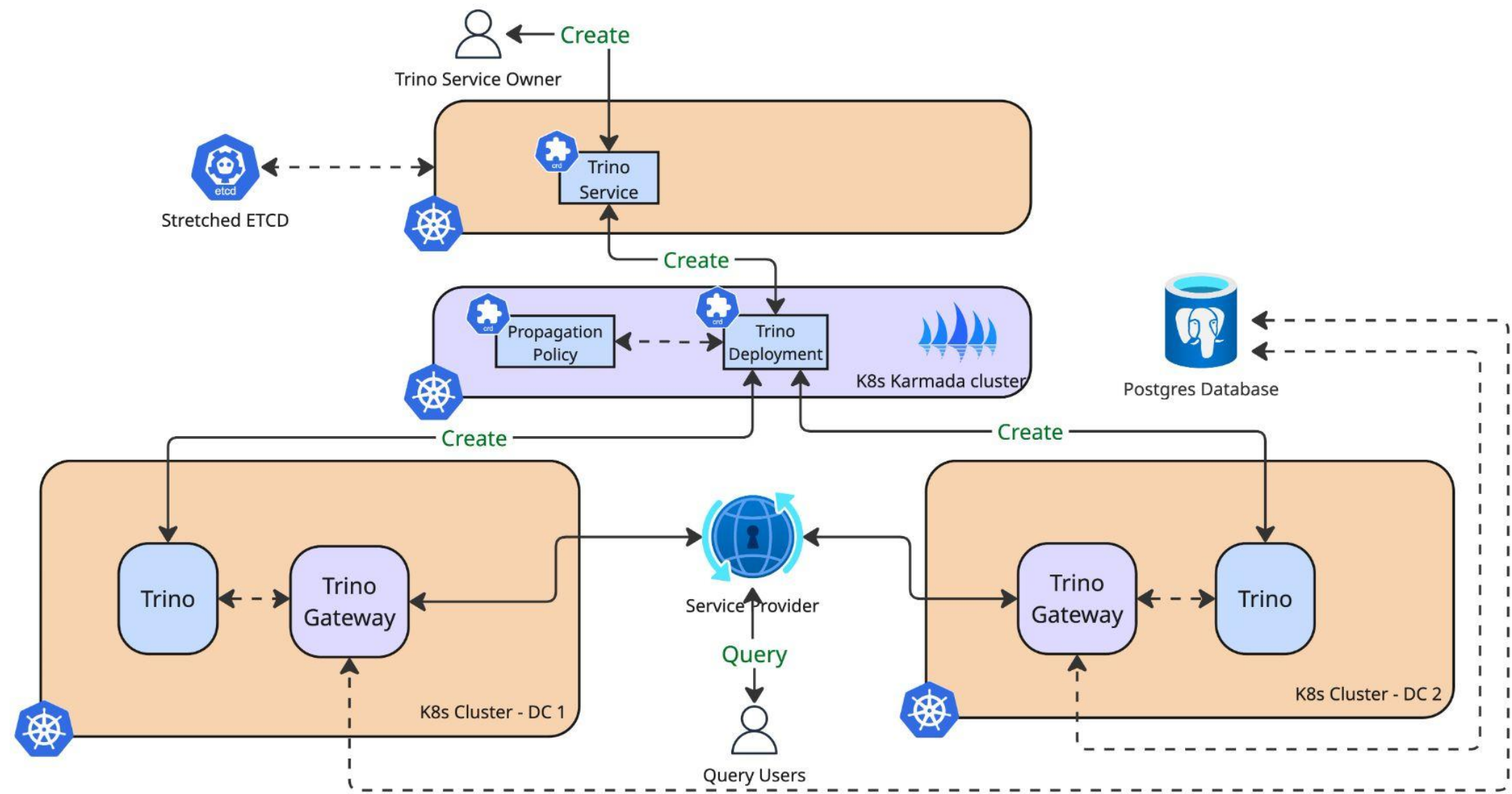
Engineering

Trino Gateway

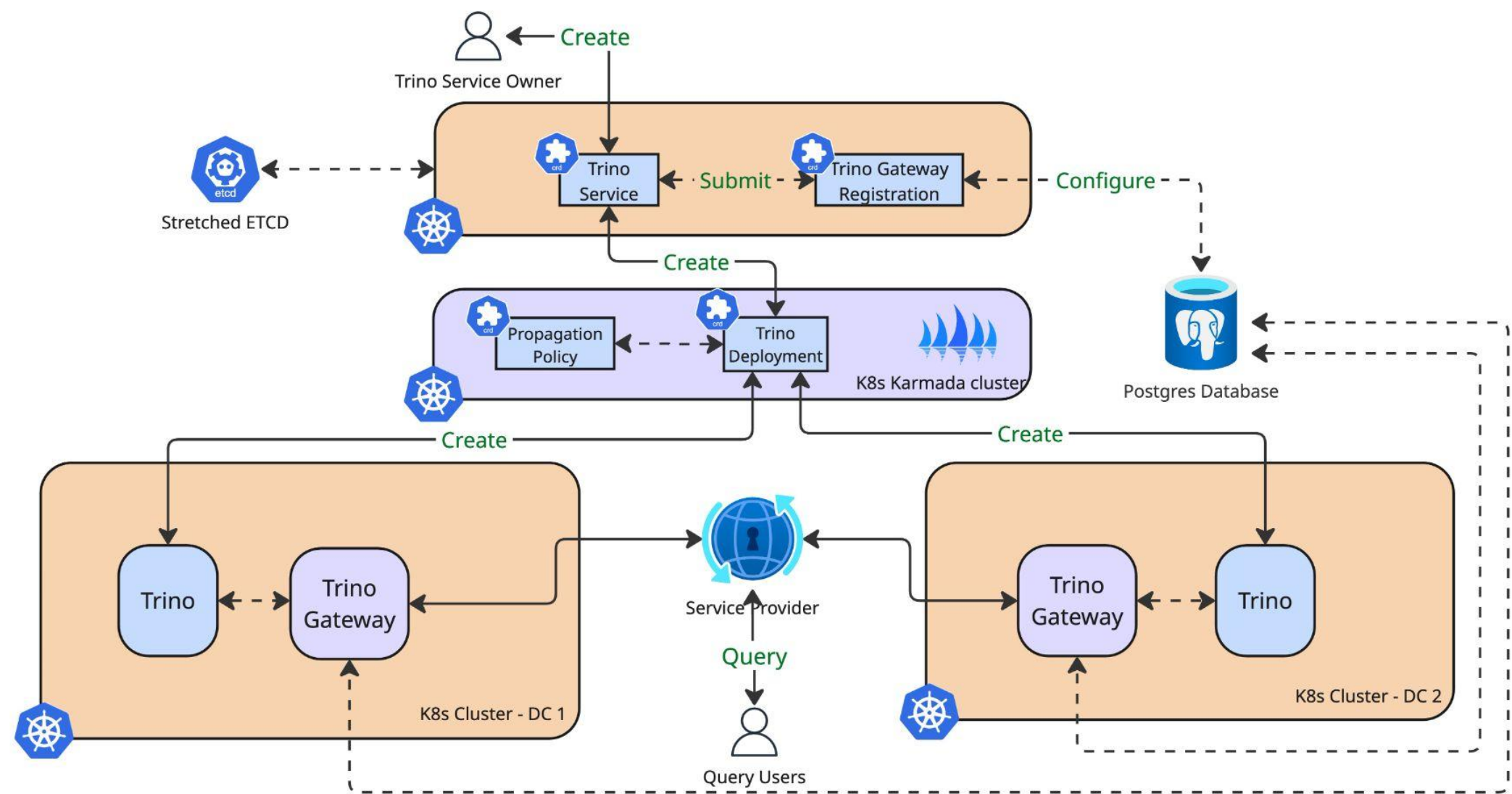
- Intelligent reverse proxy that routes queries to Trino clusters, providing high availability
- Dynamic Back-end Management
- Workload isolation through routing groups
- Single Entry Point
- Zero-downtime operations upgrades



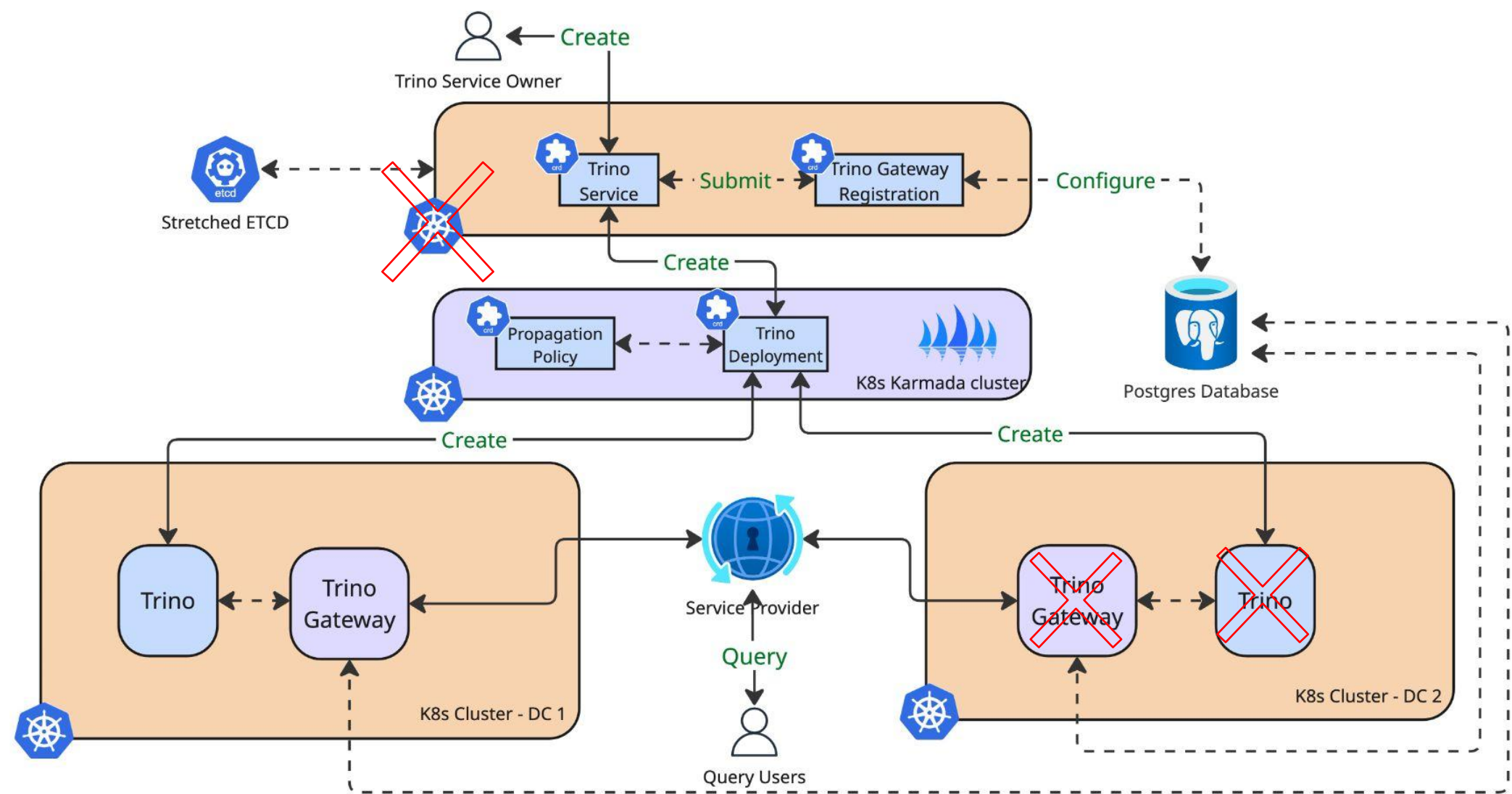
Trino Gateway



Trino Gateway Registration



Disaster Recovery



Key Benefits

- **One Simple Manifest**

- *TrinoService Owner*: Deploy one Single TrinoService manifest definition

- **Single Point of Entry**

- *TrinoService Owner*: Submit one TrinoService to a unified Trino API cluster
- *Query Users*: Get a single query endpoint via TrinoGateway through which all TrinoServices can be accessed

- **Zero-Downtime**

- *TrinoService Owner*: TrinoService CR automatically recovers from cluster failures. State persisted in stretched etcd enables rapid recovery
- *Query Users*: Your query endpoints remain stable and available. Queries are automatically routed to healthy clusters

Future Work

- Expansion to additional data centers
- Optimizing resource allocation
 - Horizontal Pod Autoscaling
- Query audit logging via event listeners
- Resource group enforcement

Thank you!

We are hiring: bloomberg.com/engineering

Bloomberg
Engineering

Contact Our Recruiters



TechAtBloomberg.com

© 2026 Bloomberg Finance L.P. All rights reserved.