KubeCon CloudNativeCon

THE LINUX FOUNDATION

OPEN SOURCE SUMMIT

AI_dev

Open Source GenAI & ML Summit

China 2024

KubeCon

CloudNativeCon

THE LINUX FOUNDATION

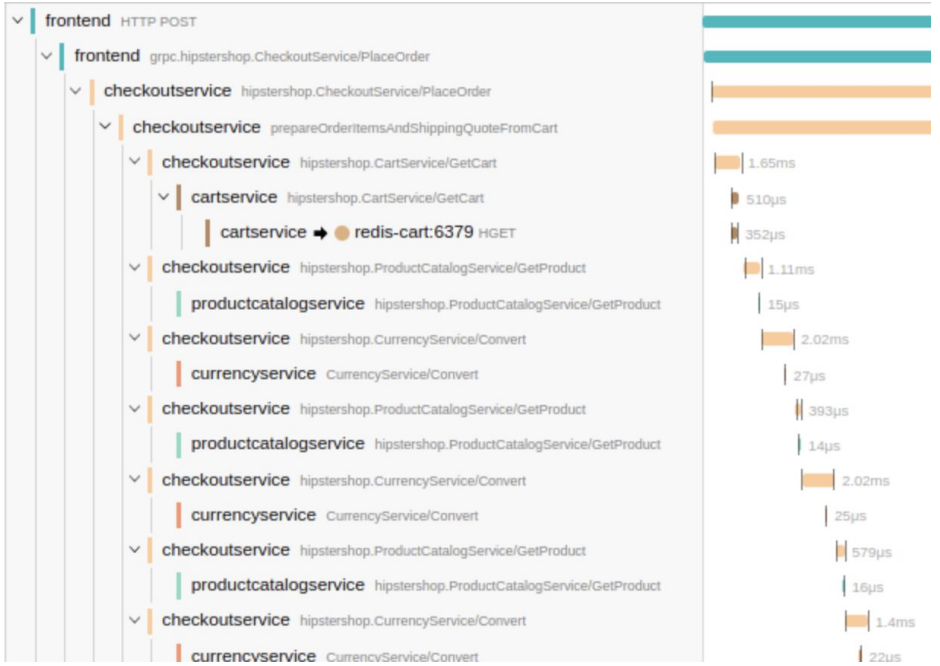
China 2024

Kelemetry

Global Control Plane Tracing for Kubernetes

Tracing

- Trace: a tree of spans
- Request-scoped
  - “Any bit of data or metadata that can be bound to *lifecycle of a single transactional object* in the system” — Peter Bourgon (2017)
  - Attach events to scope span
- RPC: synchronous call scopes define span hierarchy
- K8s: What qualifies as a transactional object?

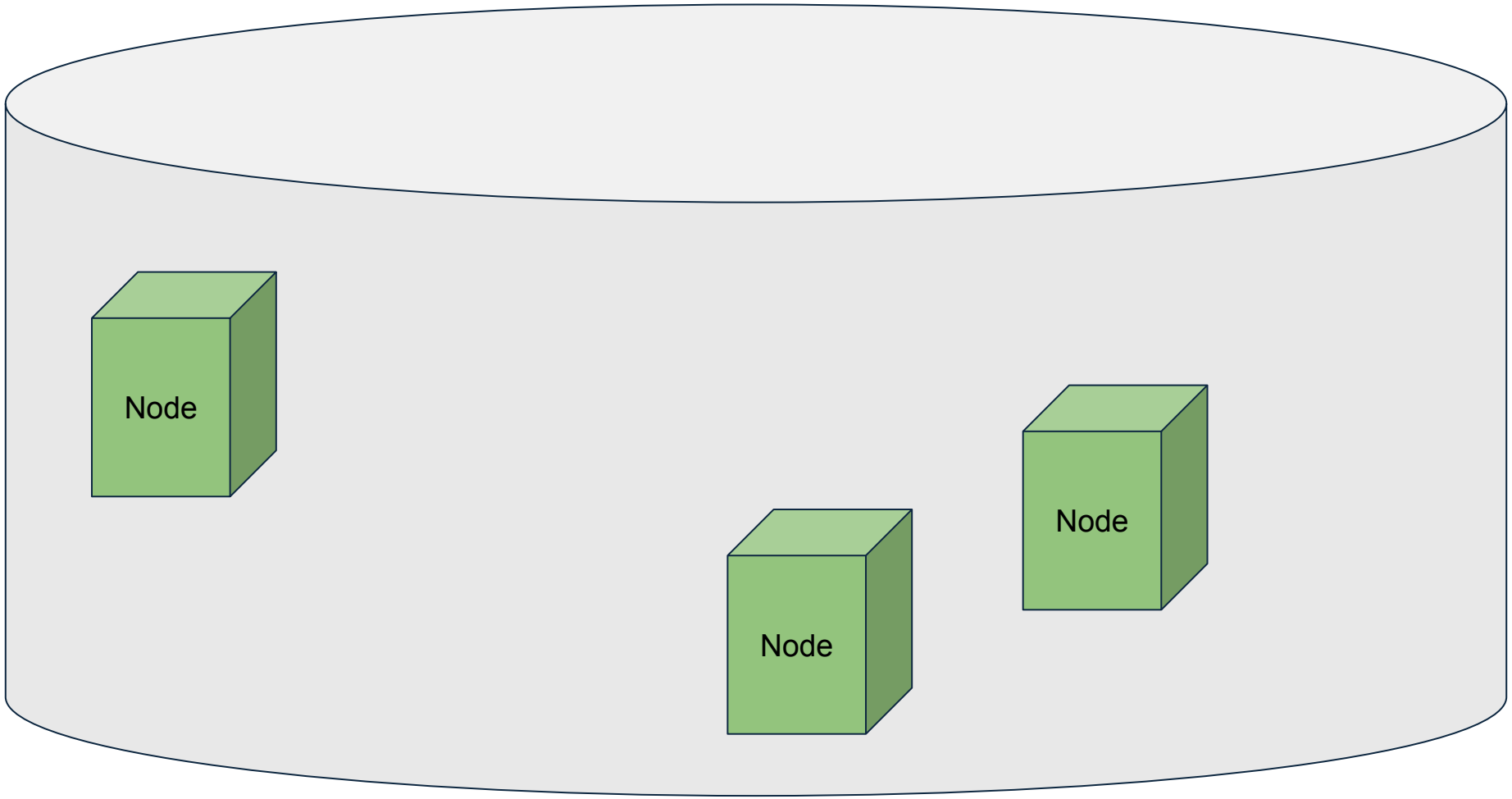


Source: https://opentelemetry.io/docs/demo/screenshots/

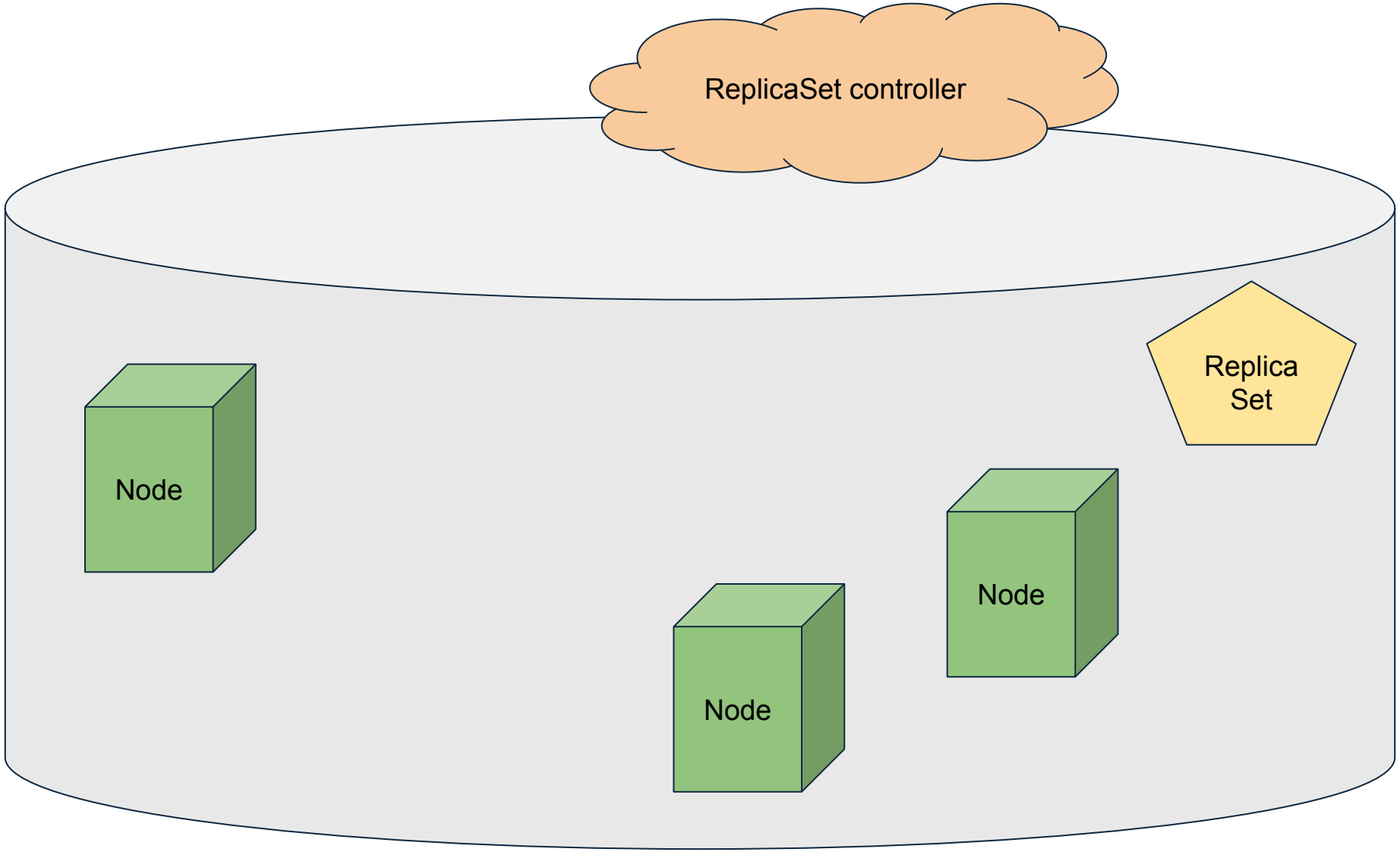
K8s: An async choreography

- K8s apiserver is just an object store
- Decentralized controllers react to object changes
- “Reconciliation”: Move system from current state to desired state
- Controllers work together to form a feedback system

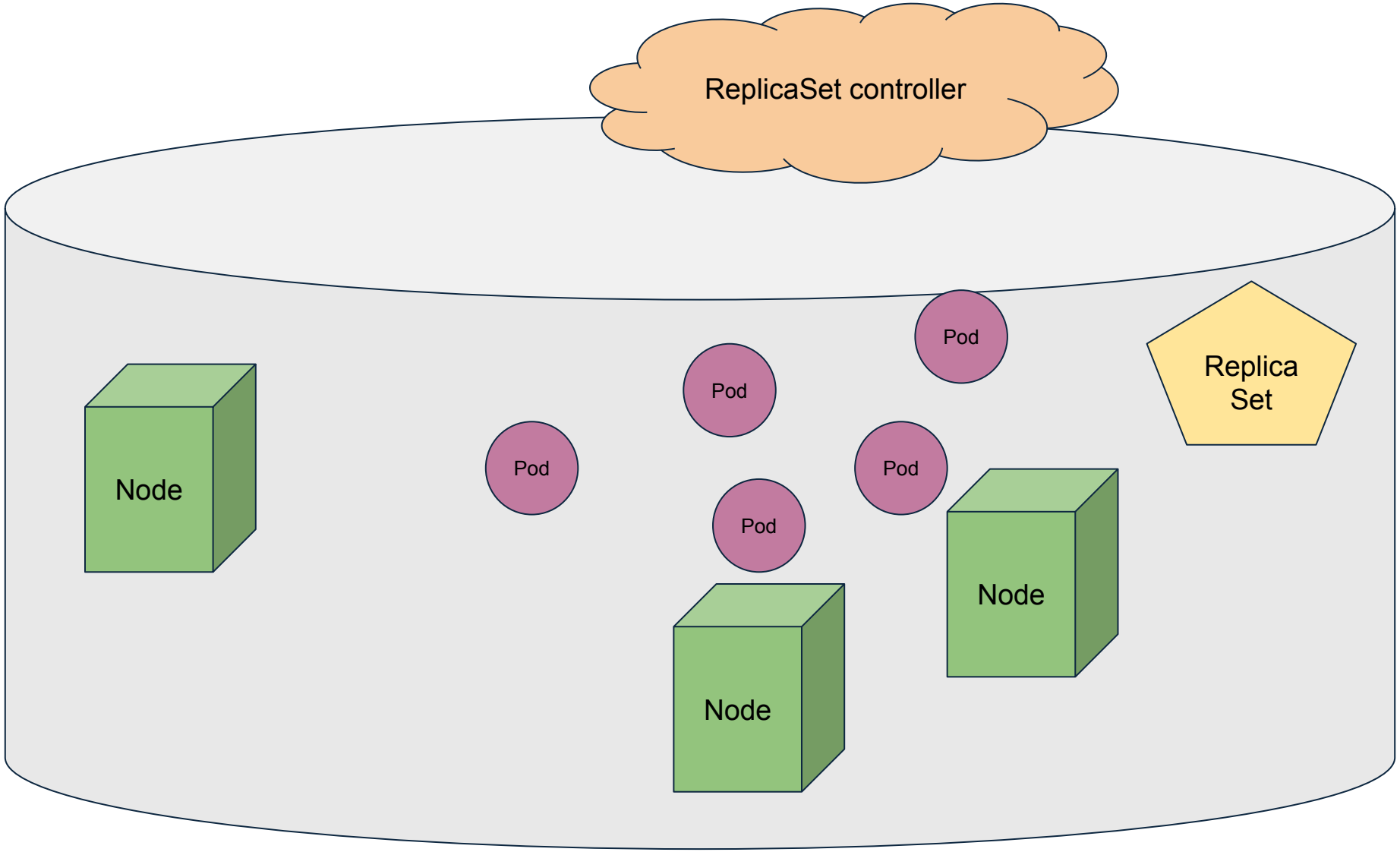
Choreography for a controller



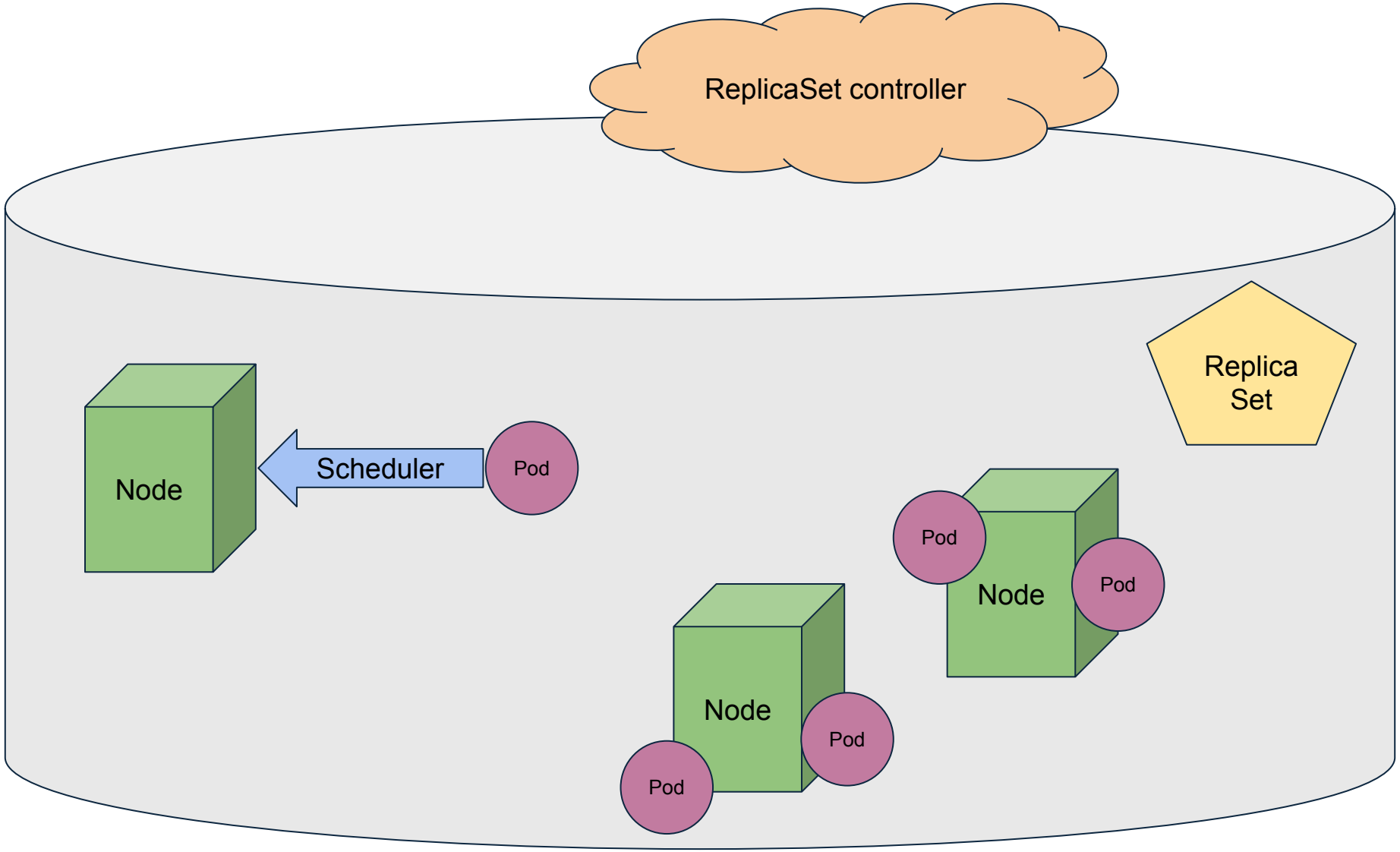
Choreography for a controller



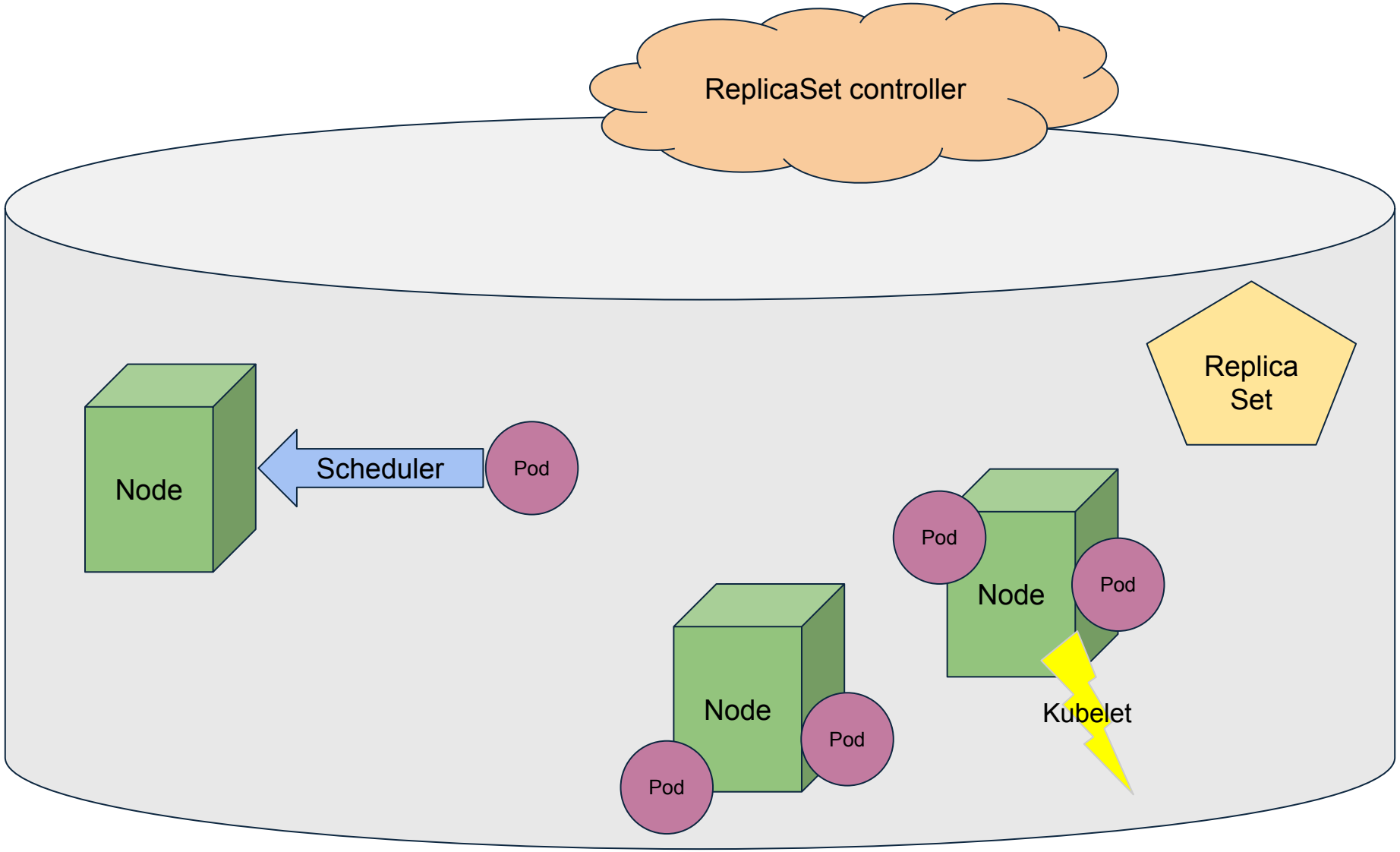
Choreography for a controller



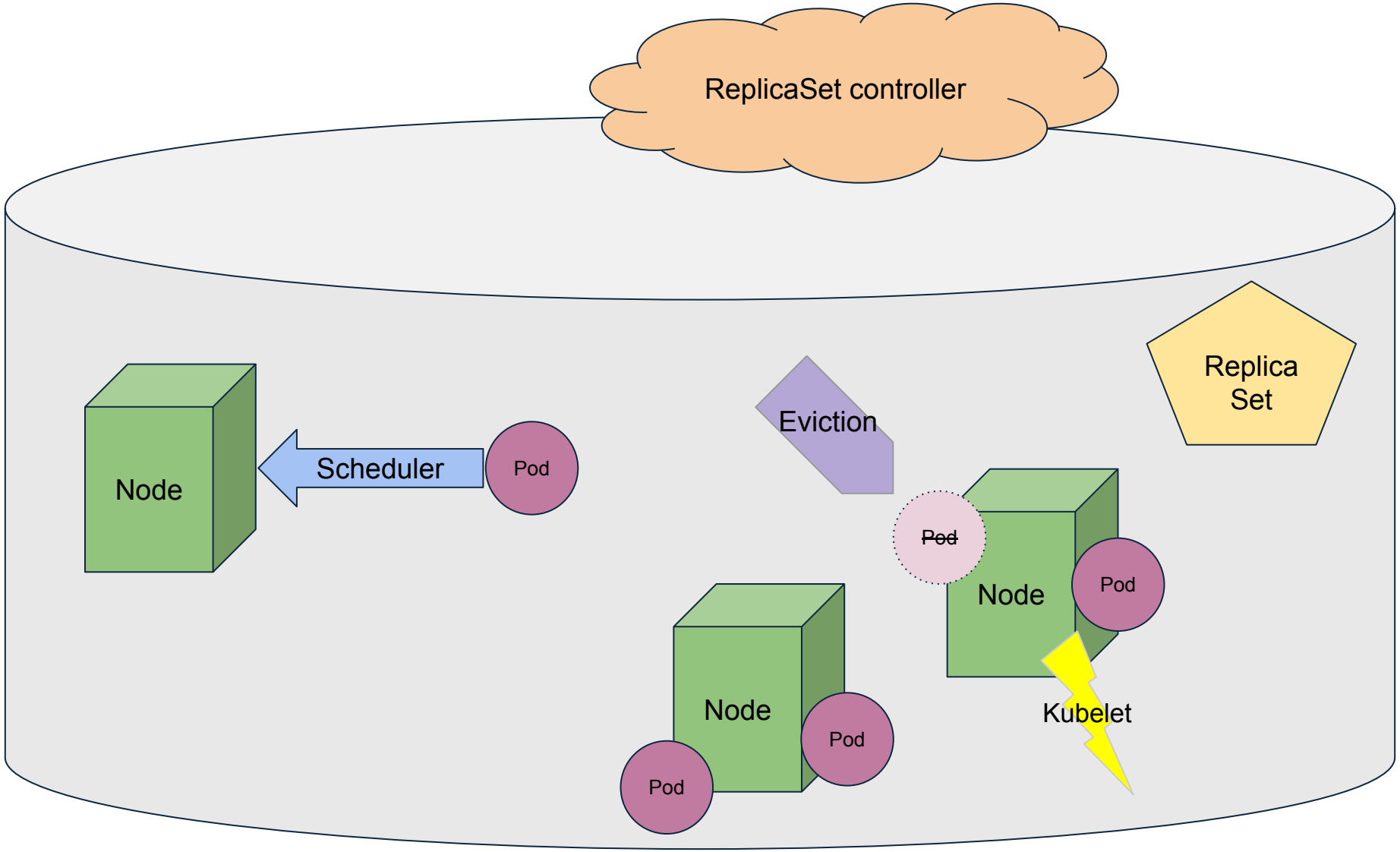
Choreography for a controller



Choreography for a controller

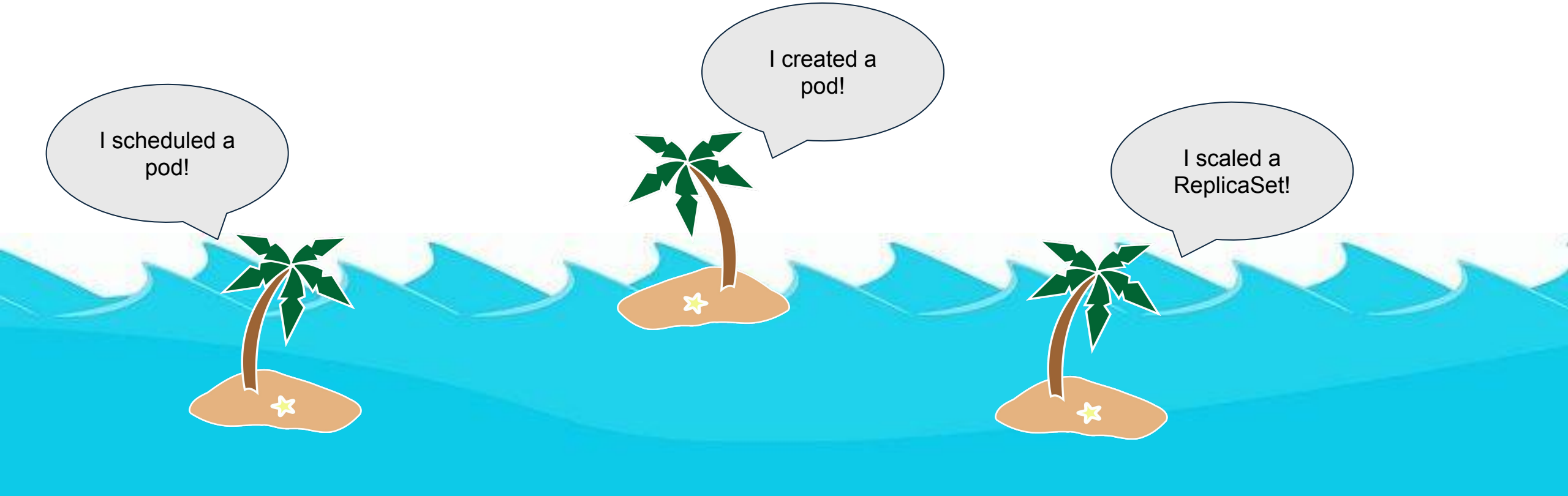


Choreography for a controller



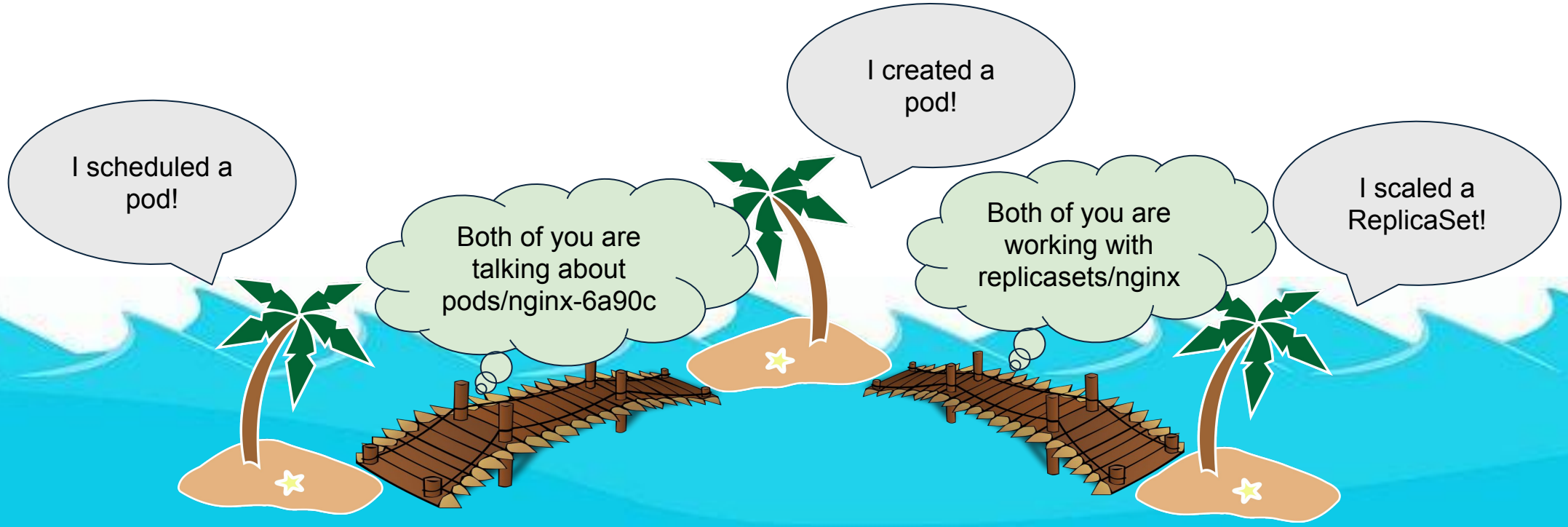
Islands of observability

- Observing single component barely gives insight into the entire system.
- Components can only export data about themselves.



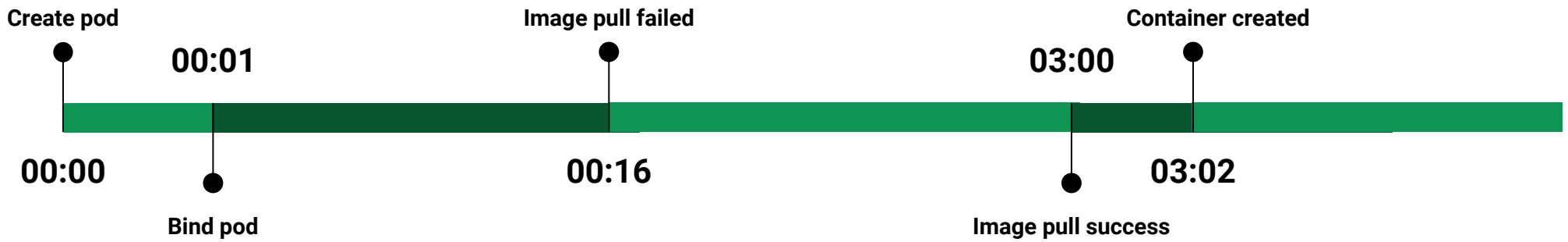
Islands of observability

- How to connect them together?
  - Operations on the same object
  - Operations on related objects



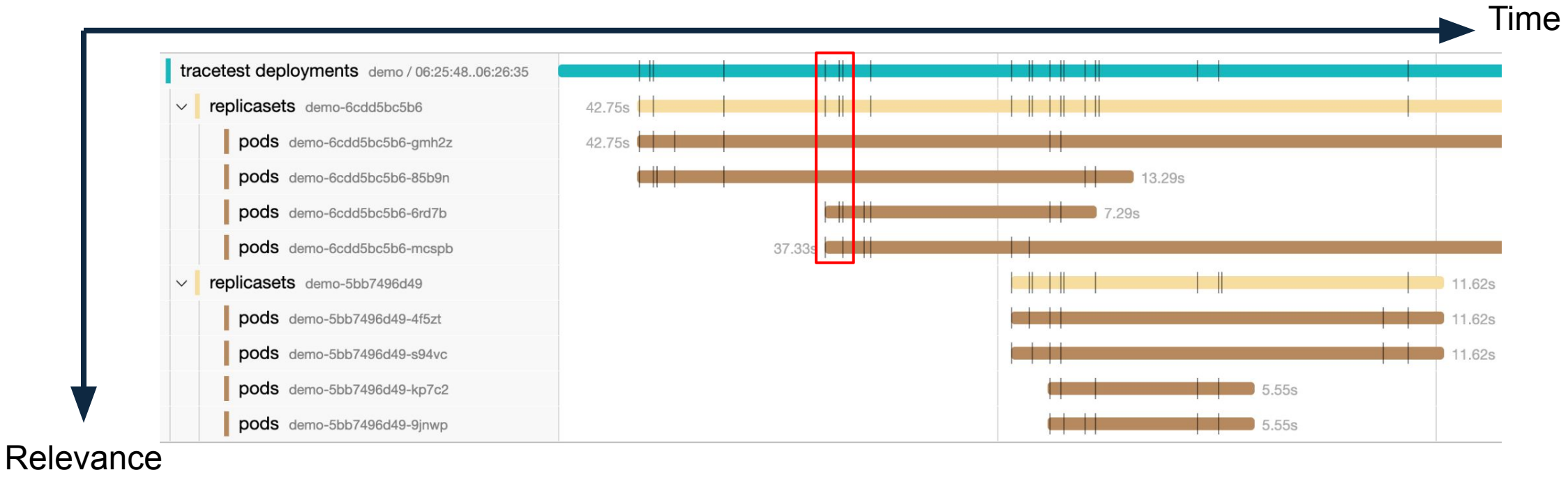
Chronological grouping

- Causal relationship: nice to have, but impossible to infer
- Consecutive events: most likely related
- Subject object for grouping, time for relevance



Object relation

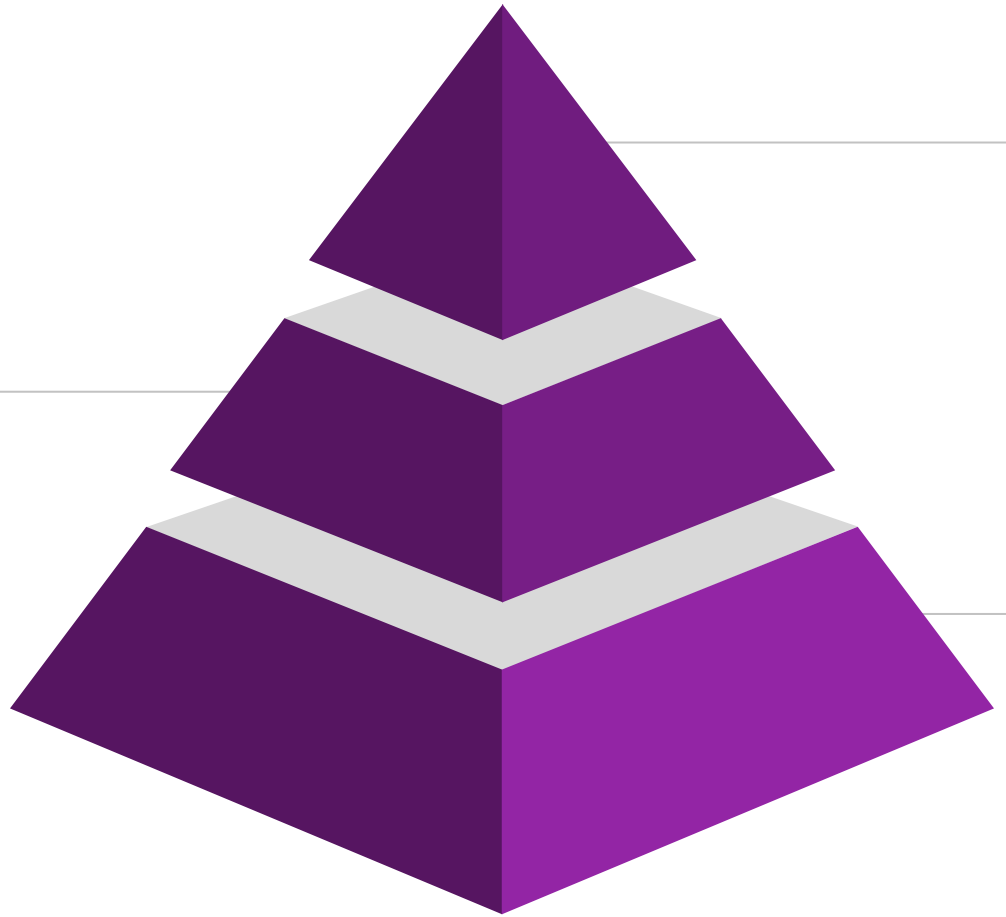
- Controllers respond to event in one object and update another object
- Add another dimension of scope: object relations
- Objects as spans, object relation trees as traces




KubeCon
CloudNativeCon
THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
AI_dev
Open Source GenAI & ML Summit
China 2024
Populating data

Events for an object

More concise



K8s events

1 High-level digest for users

Object changes

2 Component interaction mechanics

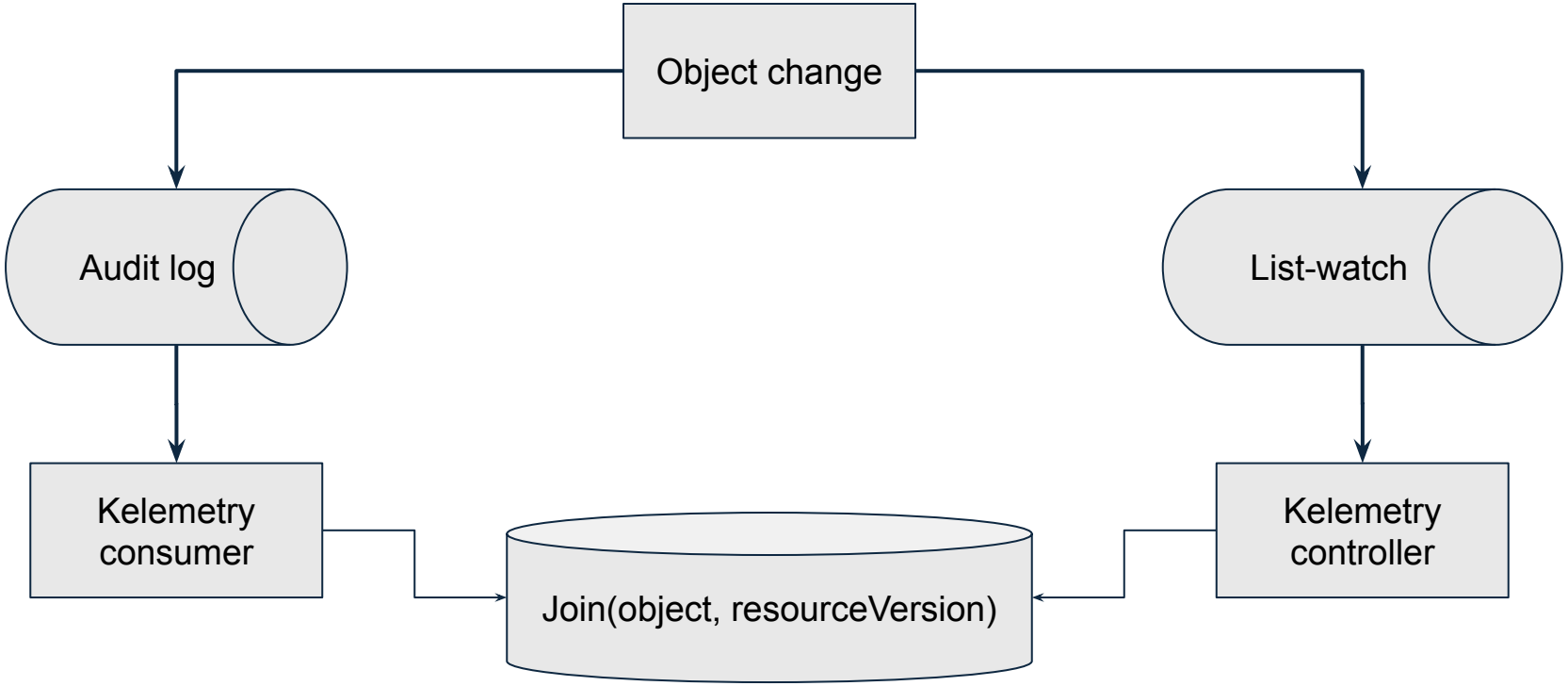
Component logs

3 Implementation details of each component

More verbose

Object changes

- Each atomic object change is an apiserver request
- Audit logs: Where, When, Who
- List-watch: What



Component logs

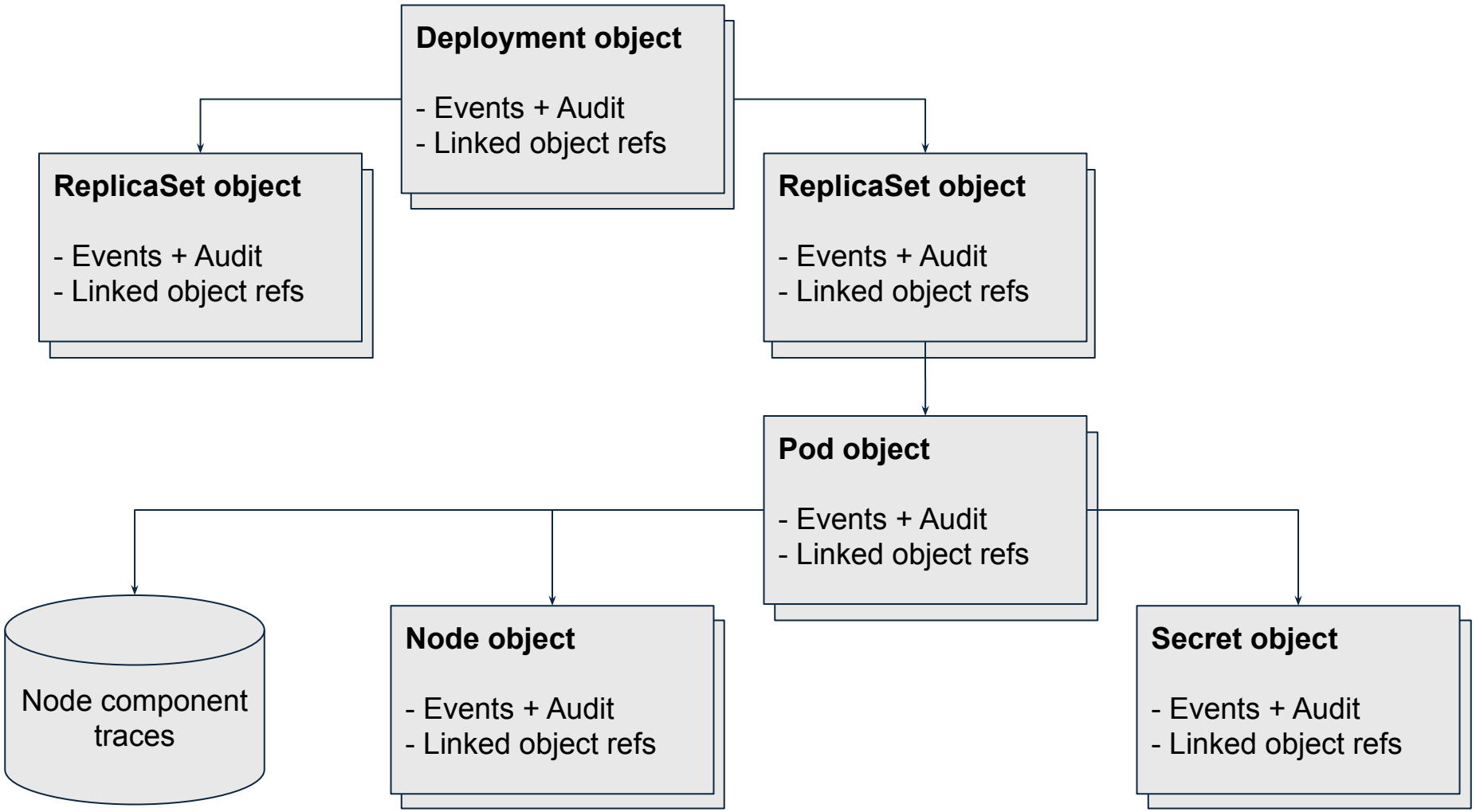
- Much greater volume than core control plane events
- Separate traces, on-demand aggregation

	Storage	**Access**
Centralized controllers	Central trace storage	Join by inferred trace tags
Node components	Local BadgerDB in node	Fetch from node directly

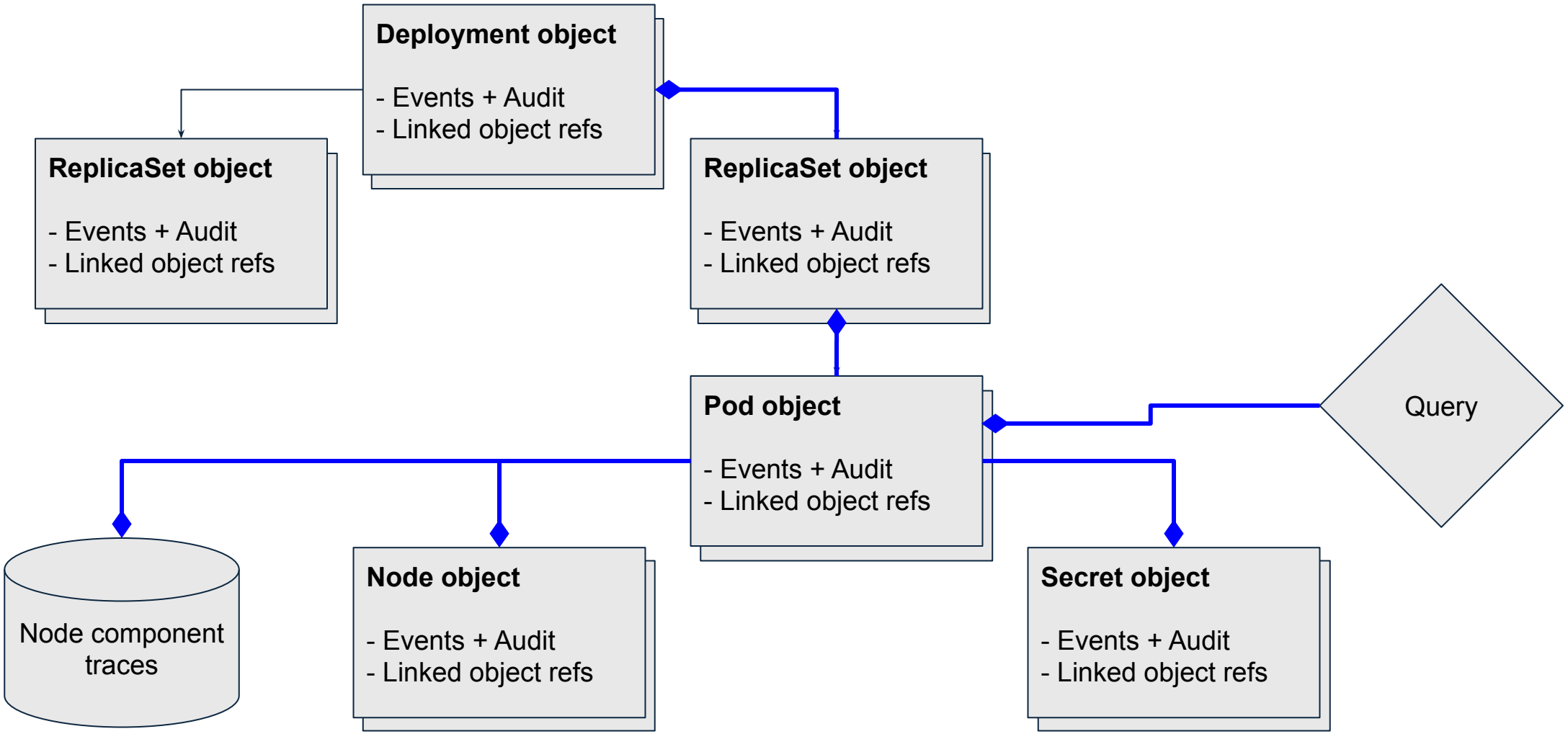
Linking objects

- Owner references: idiomatic representation of child objects
- Application-specific rules
  - Pod -> Node, Secret, etc
  - Helm release -> managed objects
  - Multi-cluster links
- Reference owner in annotation

Frontend aggregation

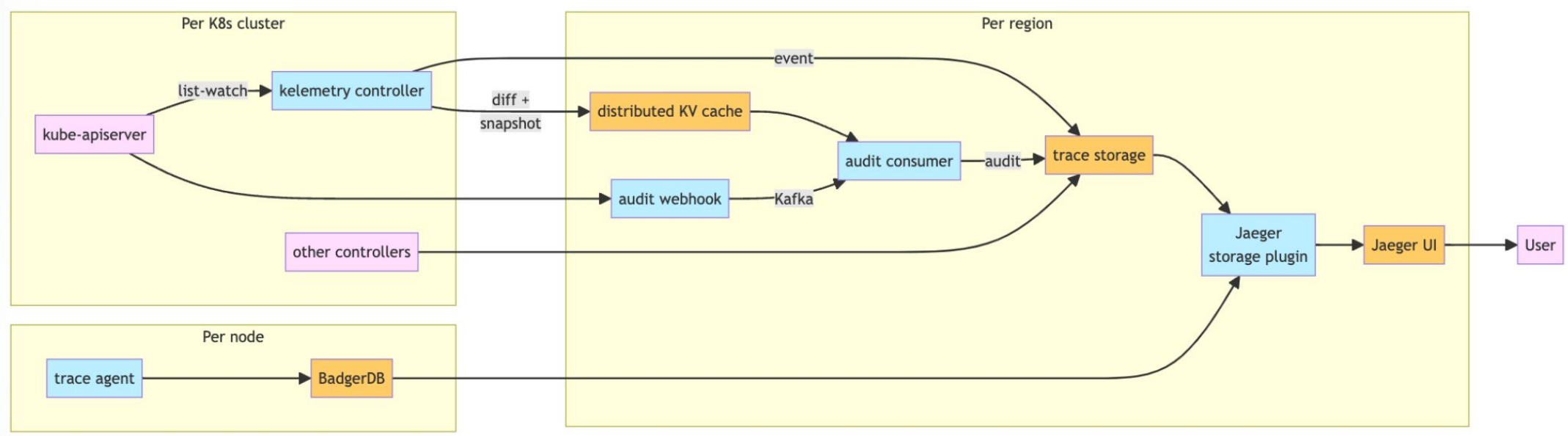


Frontend aggregation



Deploy at scale

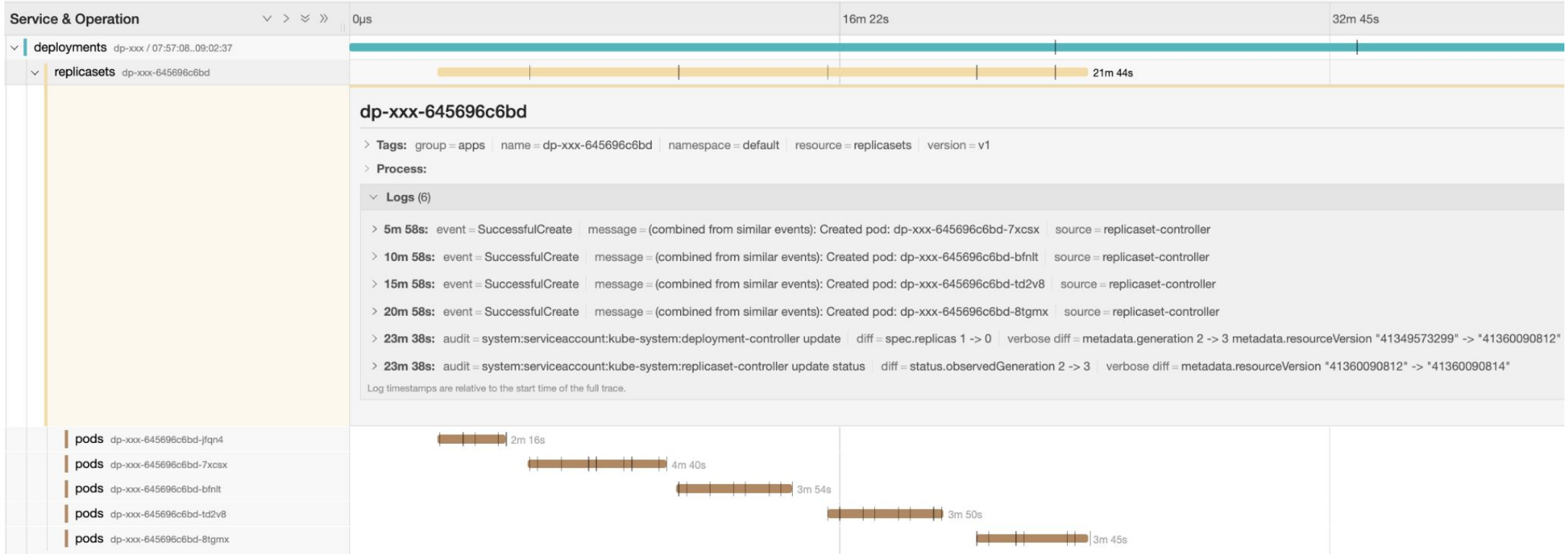
- At ByteDance:
  - >600 clusters per region
  - ~13s P99 E2E latency
  - ~10 billion events per day
- Centralized trace storage and audit consumer
- Two list-watch controllers per cluster




KubeCon
CloudNativeCon
THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
AI_dev
Open Source GenAI & ML Summit
China 2024
Applications

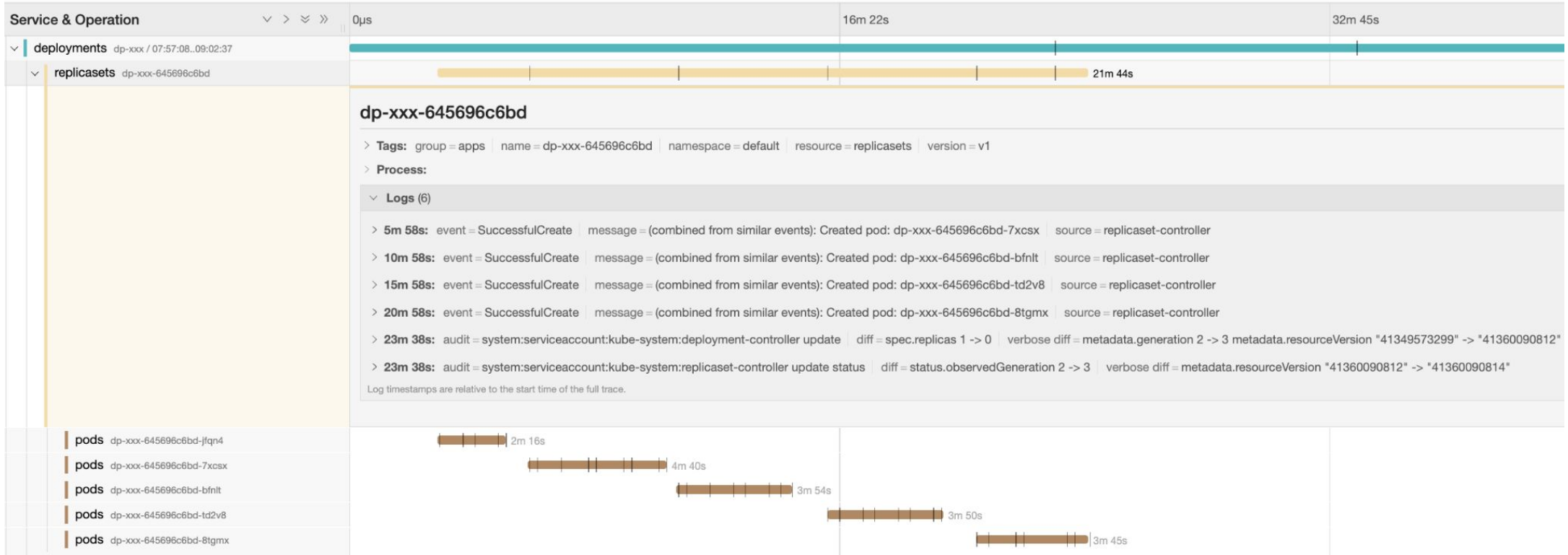
Case: Too many pods!

- Complaint: User set Deployment spec.replicas to 1, but it keeps creating new pods
- Diagnosis: Check the trace of deployment + child pods



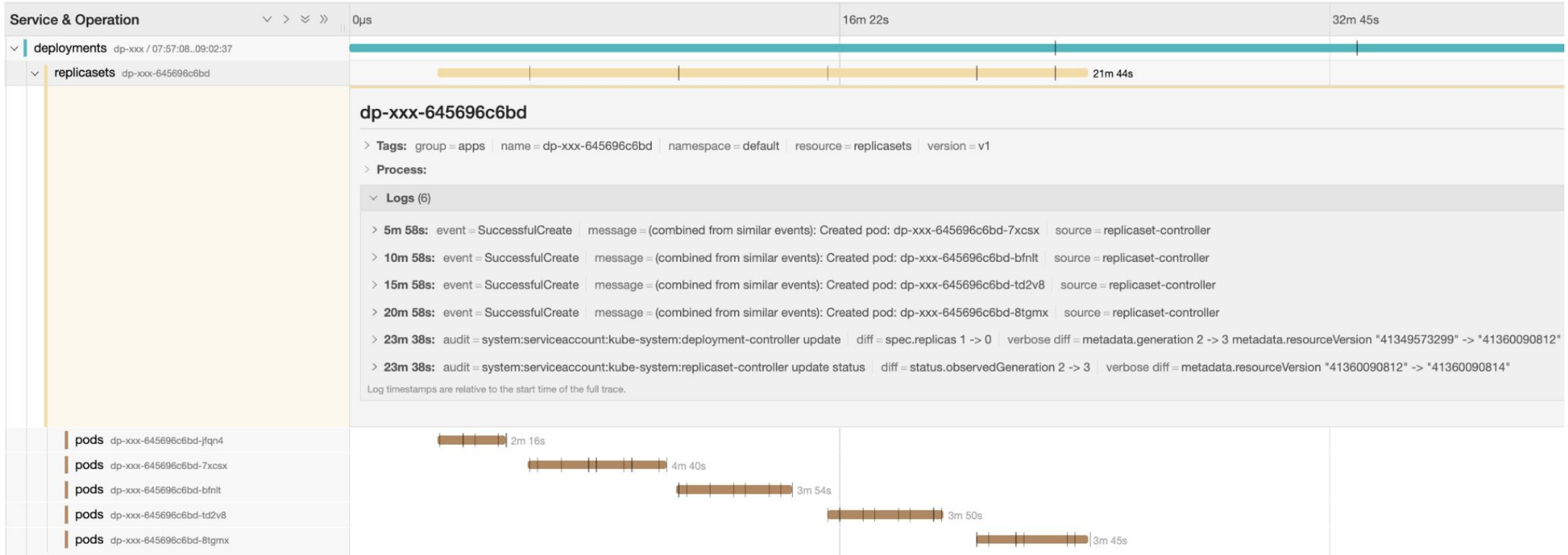
Case: Too many pods!

- A new pod is created approximately every 5 minutes
- No interaction attempted by kube-controller-manager



Case: Too many pods!

- SuccessfulCreate event was produced by ReplicaSet controller
  - Pod creation response received by controller
- ReplicaSet.status.replicas never increased
  - Computed from pod informer in controller-manager



Case: Too many pods!

- Inconsistency between CREATE requests and WATCH informer
- “5 minutes”: clear indication of ReplicaSet controller not observing created pods
- Resolution: resolve informer issues caused by consistently failing list-watch loop

```
const (
        // If a watch drops a delete event for a pod, it'll take this long
        // before a dormant controller waiting for those packets is woken up anyway. It is
        // specifically targeted at the case where some problem prevents an update
        // of expectations, without it the controller could stay asleep forever. This should
        // be set based on the expected latency of watch events.
        //
        // Currently a controller can service (create *and* observe the watch events for said
        // creation) about 10 pods a second, so it takes about 1 min to service
        // 500 pods. Just creation is limited to 20qps, and watching happens with ~10-30s
        // latency/pod at the scale of 3000 pods over 100 nodes.
        ExpectationsTimeout = 5 * time.Minute
```

https://github.com/kubernetes/kubernetes/blob/master/pkg/controller/controller_utils.go

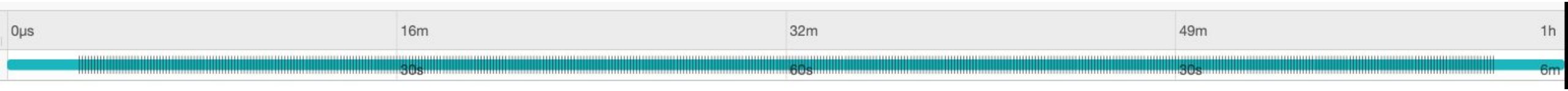
Case: Controller timeout

- Background: cluster-metrics
  - Leader-elected controller
  - Aggregates cluster compute resources
  - Writes result to a “ClusterResource” CRD every minute
- Symptom: Alarm indicates ClusterResource is outdated for many minutes

Case: Controller timeout

Object had no updates during the period

Leader lease was updated normally



Case: Controller timeout

- Burst of update requests *after* the silence period
- Component logs: burst after silence period
- Conclusion: system problem led to all klog calls blocking on disk write

```
44m 25s: audit = system:serviceaccount:tce-system:cluster-metrics update (Conflict)
44m 25s: audit = system:serviceaccount:tce-system:cluster-metrics update (Conflict)
44m 25s: audit = system:serviceaccount:tce-system:cluster-metrics update (Conflict)
44m 25s: audit = system:serviceaccount:tce-system:cluster-metrics update  | sourceIP
44m 25s: audit = system:serviceaccount:tce-system:cluster-metrics update (Conflict)
44m 25s: audit = system:serviceaccount:tce-system:cluster-metrics update (Conflict)
44m 25s: audit = system:serviceaccount:tce-system:cluster-metrics update (Conflict)
44m 25s: audit = system:serviceaccount:tce-system:cluster-metrics update (Conflict)
44m 25s: audit = system:serviceaccount:tce-system:cluster-metrics update (Conflict)
44m 25s: audit = system:serviceaccount:tce-system:cluster-metrics update (Conflict)
44m 25s: audit = system:serviceaccount:tce-system:cluster-metrics update (Conflict)
44m 25s: audit = system:serviceaccount:tce-system:cluster-metrics update (Conflict)
44m 25s: audit = system:serviceaccount:tce-system:cluster-metrics update (Conflict)
44m 25s: audit = system:serviceaccount:tce-system:cluster-metrics update (Conflict)
44m 25s: audit = system:serviceaccount:tce-system:cluster-metrics update (Conflict)
44m 25s: audit = system:serviceaccount:tce-system:cluster-metrics update (Conflict)
44m 25s: audit = system:serviceaccount:tce-system:cluster-metrics update (Conflict)
44m 25s: audit = system:serviceaccount:tce-system:cluster-metrics update (Conflict)
44m 25s: audit = system:serviceaccount:tce-system:cluster-metrics update (Conflict)
44m 25s: audit = system:serviceaccount:tce-system:cluster-metrics update (Conflict)
44m 25s: audit = system:serviceaccount:tce-system:cluster-metrics update (Conflict)
44m 25s: audit = system:serviceaccount:tce-system:cluster-metrics update (Conflict)
44m 25s: audit = system:serviceaccount:tce-system:cluster-metrics update (Conflict)
```

Other integrations

- Integration/E2E tests
  - Explain the API-level changes of a flaky test
  - No need to retain test environment
- Documentation
  - Visualize component architecture with an example trace

Use Kelemetry

- Web preview on GitHub Pages
- 5-minute quickstart
- Helm chart

GitHub:
kubewharf/kelemetry

Web preview

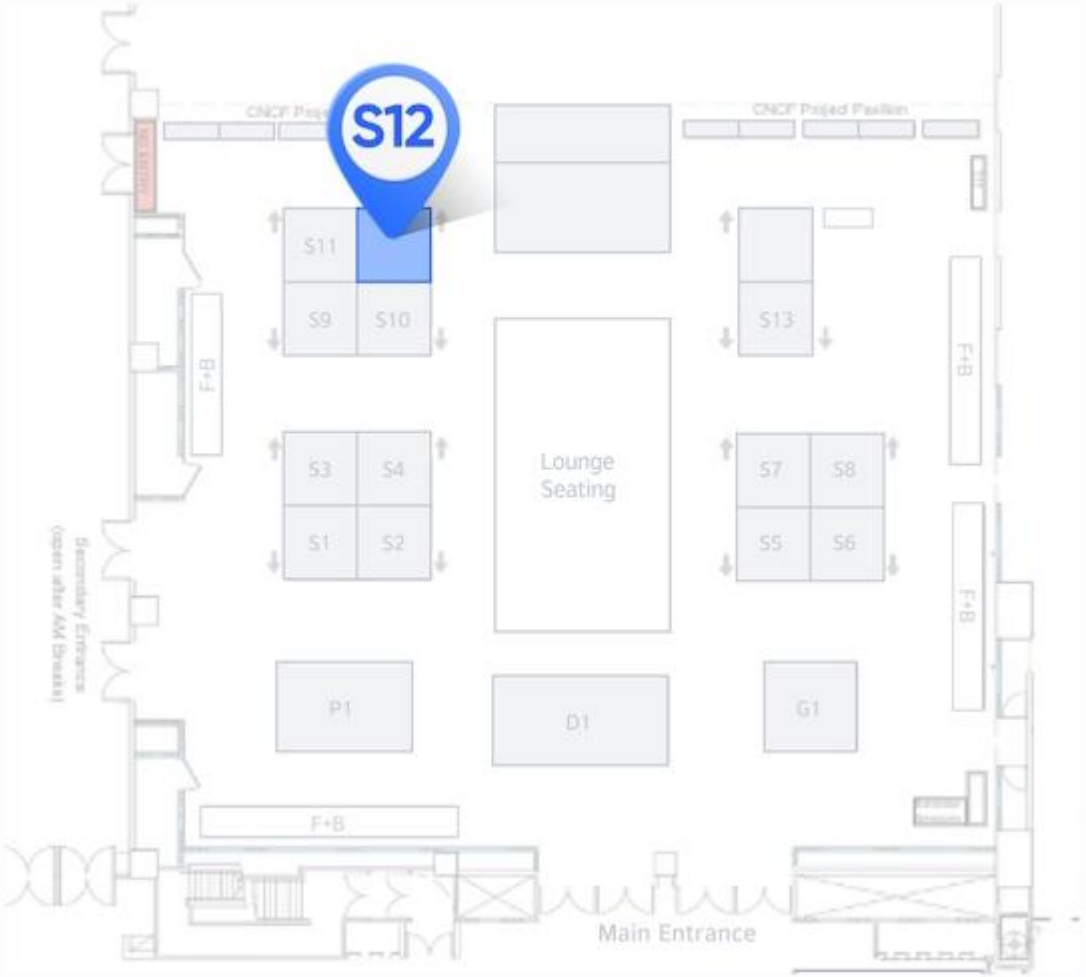
Future directions

- Scriptable linking rules
- Node trace producer by eBPF agent
- Offline trace analysis for automatic diagnosis

GitHub:
kubewharf/kelemetry

Web preview

字节跳动云原生开源项目—— KubeWharf 现场展位




KubeCon
CloudNativeCon
THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
AI_dev
Open Source GenAI & ML Summit
China 2024