High Availability Drools

Massimiliano Dessi

Linux Day 2019

GULCh
Gruppo Utenti Linux Cagliari h...?

Speaker

Massimiliano Dessì
desmax74

RedHatter, GDG Sardegna, JUG Sardegna, Spring AOP Author, Java & Golang

Red Hat

eXistenZ | Oasis | Indastria

https://twitter.com/desmax74

Block or report user

Organizations

Red Hat

Drools

Massimiliano Dessi has more than 19 years of experience in programming, plus experience with cloud and automation.

He works for Red Hat in the Drools core team.

He's husband and father of three, Manager of GDG Sardegna, Co-founder of JugSardegna, member of the Cagliari Linux User Group (Gulch).

GULCh
Gruppo Utenti Linux Cagliari h...?

GDG Sardegna

Drools is a Business Rules Management System (BRMS).

A Rule Engine allows you to define “What to Do” instead of “How to do it”

```
rule  <rule_name>
   when
      <conditions>

   then
      <actions>
end
```

Writing and using Rules,
you create a repository of knowledge
(a knowledge base) which is executable.

A business rule is a piece of logic that captures "what to do" depending on a context (Facts)

```
rule Thought problem
    when
        Question of Life

    then
        Answer is 42
end
```

Side Effects (actions not involving working memory)

- Have to be confined in lambda expressions in rule consequences
- Only leader executes them and put their (eventual) result on a queue
- Replicas don’t execute them but read their results (when necessary) from the queue

```
rule StockLog when
  $s : StockTickEvent()
then
    DroolsExecutor.getInstance().execute( () -> {
        String id = UUID.randomUUID().toString();
        System.out.println("Price for " + $s.getCompany() + " is " +
                           $s.getPrice() + " id:" + id);
        return id;
    });
end
```

Drools HA

Our goal is to

provide an architecture for

High Availability Drools

(especially for Complex Event

Processing scenarios) to support failover that

automatically recovers from a node failure.

Drools HA

How to reach our HA goal

without reinvent the wheel ?

HA with Openshift/k8s

Openshift/k8s provides the cluster foundations to have a desidered number of replica defined on a yaml and the infrastructure function to elect a leader in a cluster and expose the API to interact with Drools, let go to describe the architecture

Leader Election

Multiple instances of our application

compete for a leadership .

Only one of them can become the leader,

the others are replicas.

Leader Election

All the instances periodically try to get a leadership and whichever comes first - becomes a leader. The new leader remain until its pod die or it yields the leadership. When the leader die, one of the other replica pods can become the new leader.

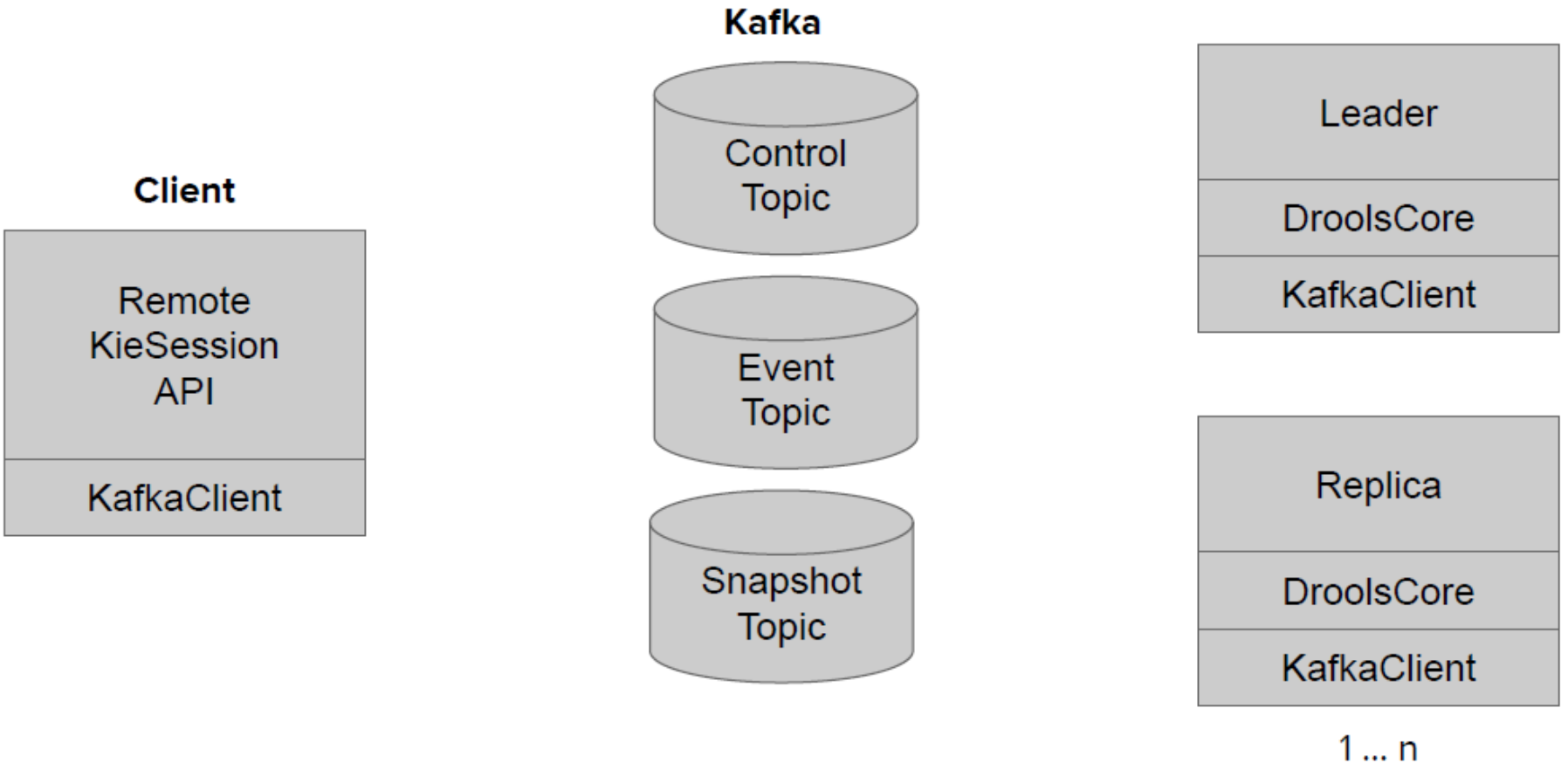
Leader election use ConfigMap

The ConfigMap API object holds key-value pairs of configuration data that can be consumed in pods or used to store configuration data for system components such as controllers. ConfigMap is similar to secrets, but designed to more conveniently support working with strings that do not contain sensitive information.

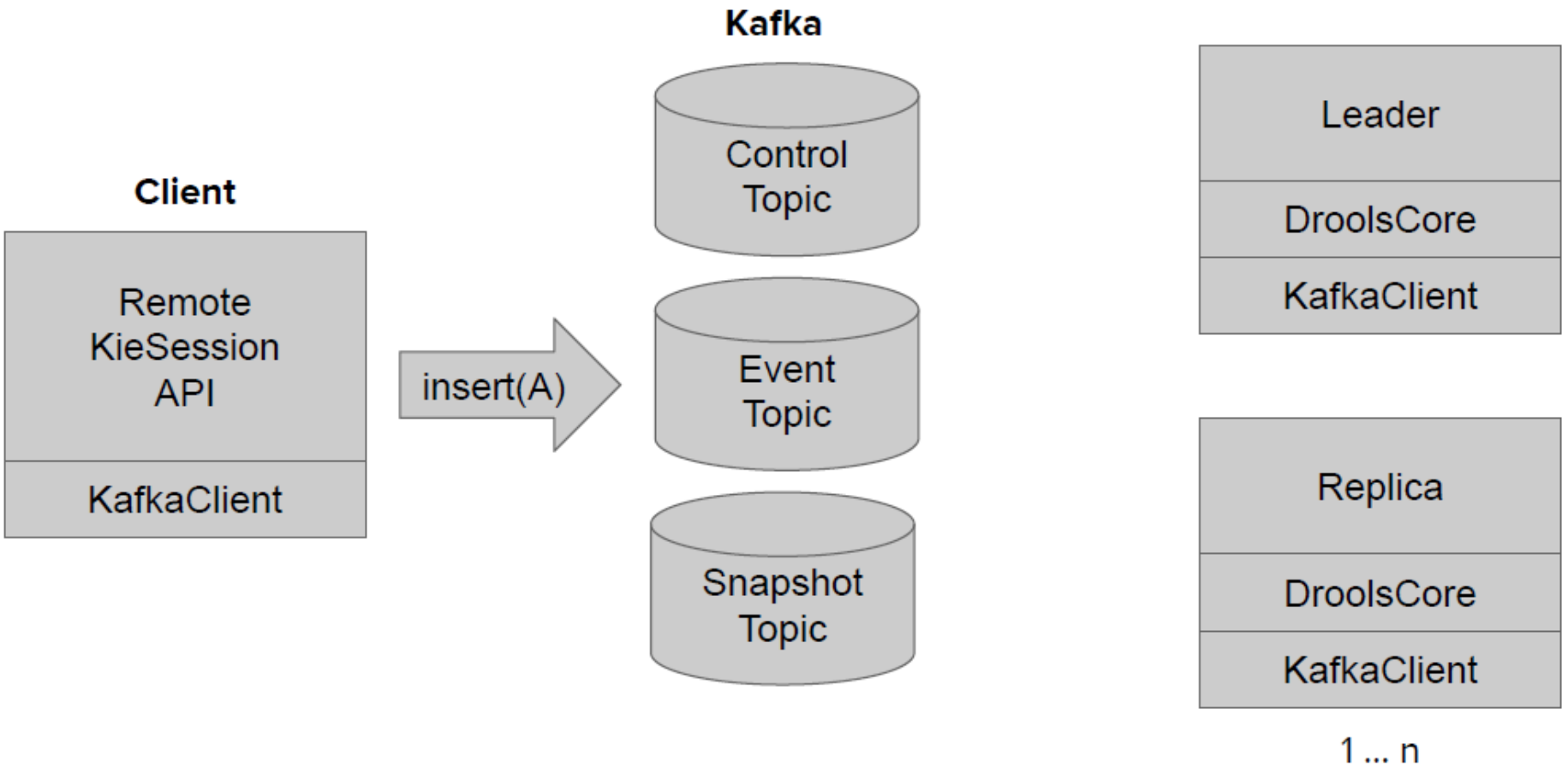
Leader - Replica Coordination

Leader – Replica coordination
and
events from outside the cluster
are
based on a distributed messaging system

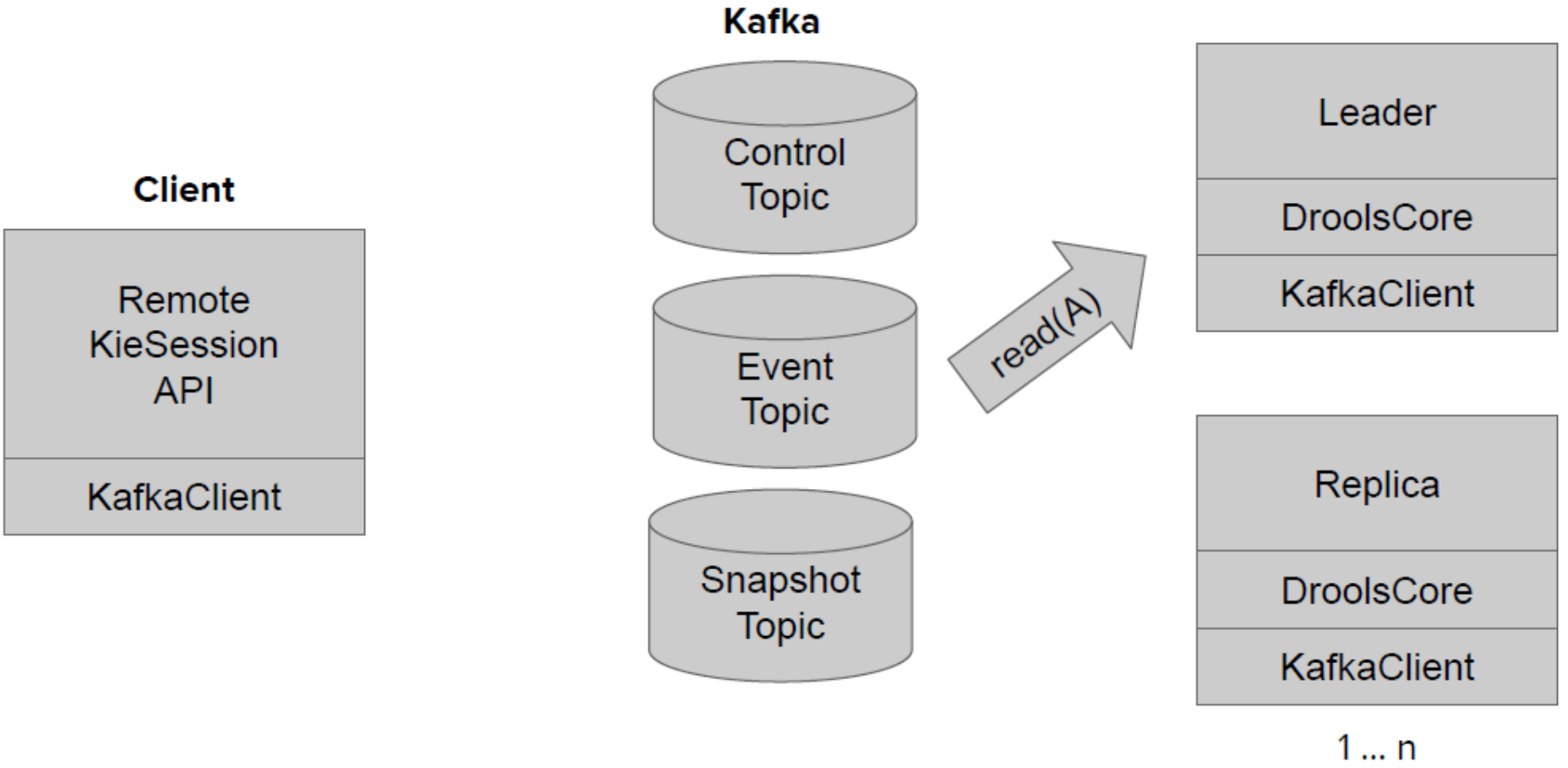
General architecture



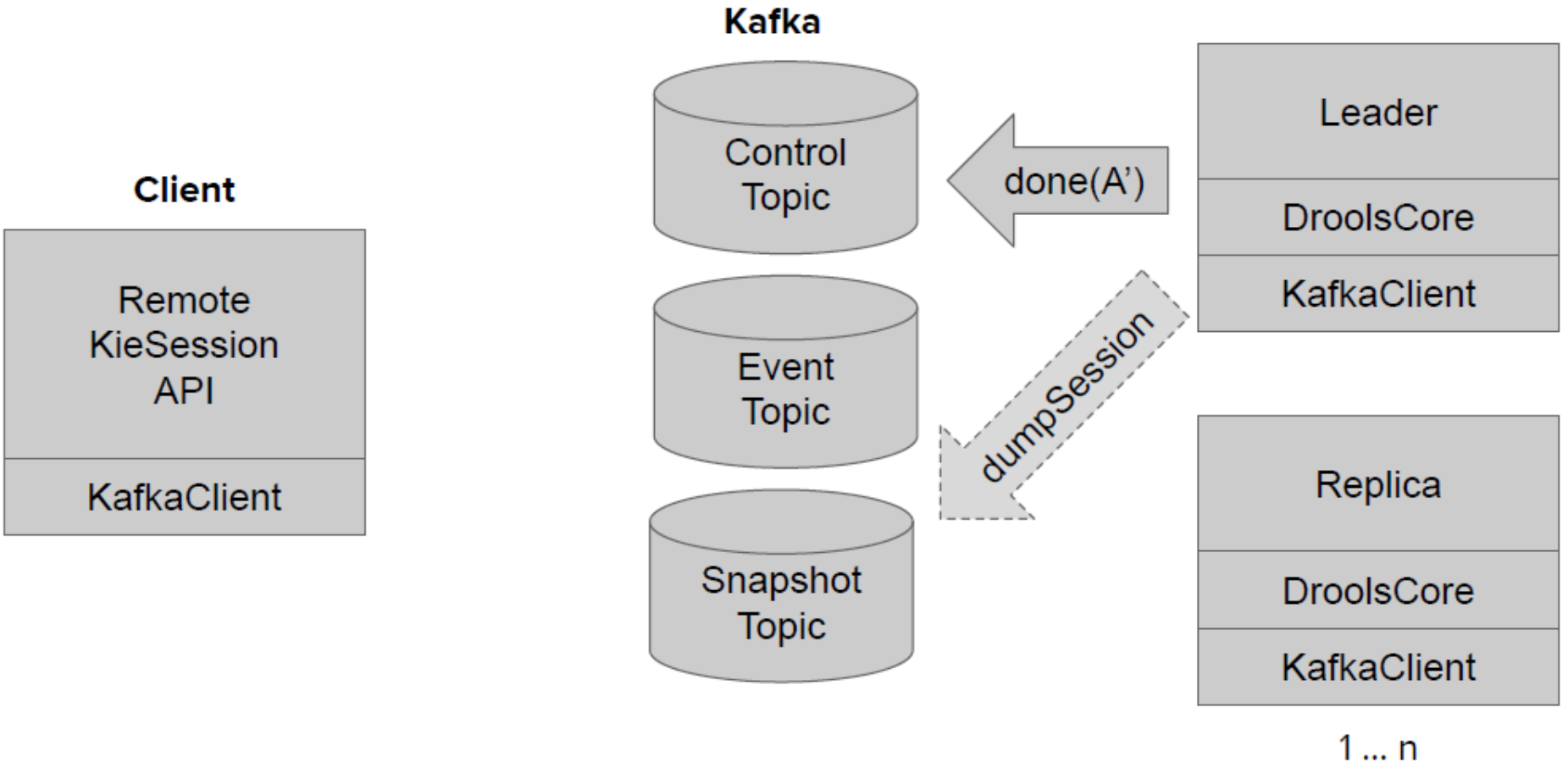
Leader - Replica Coordination



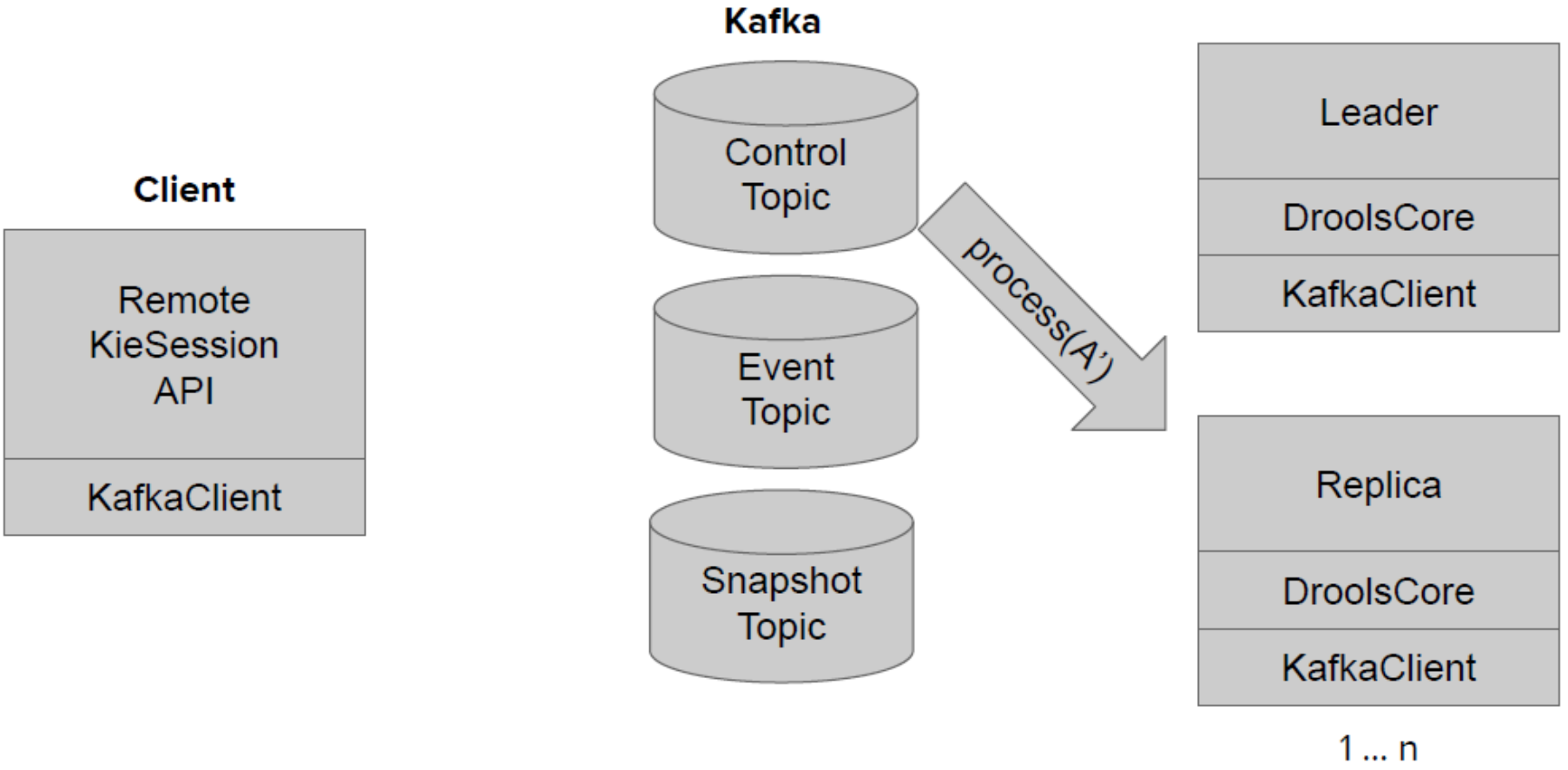
Leader - Replica Coordination



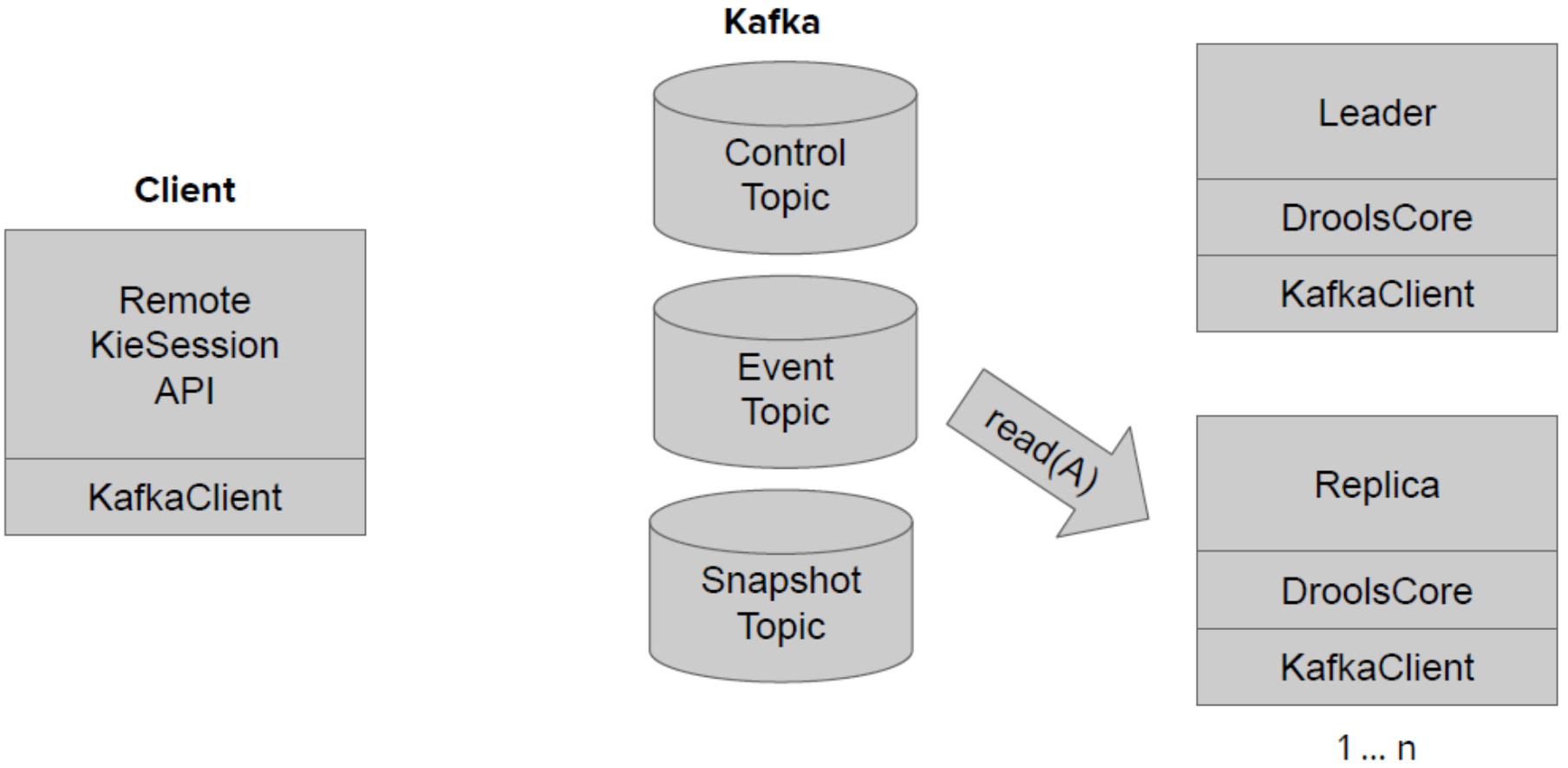
Leader - Replica Coordination



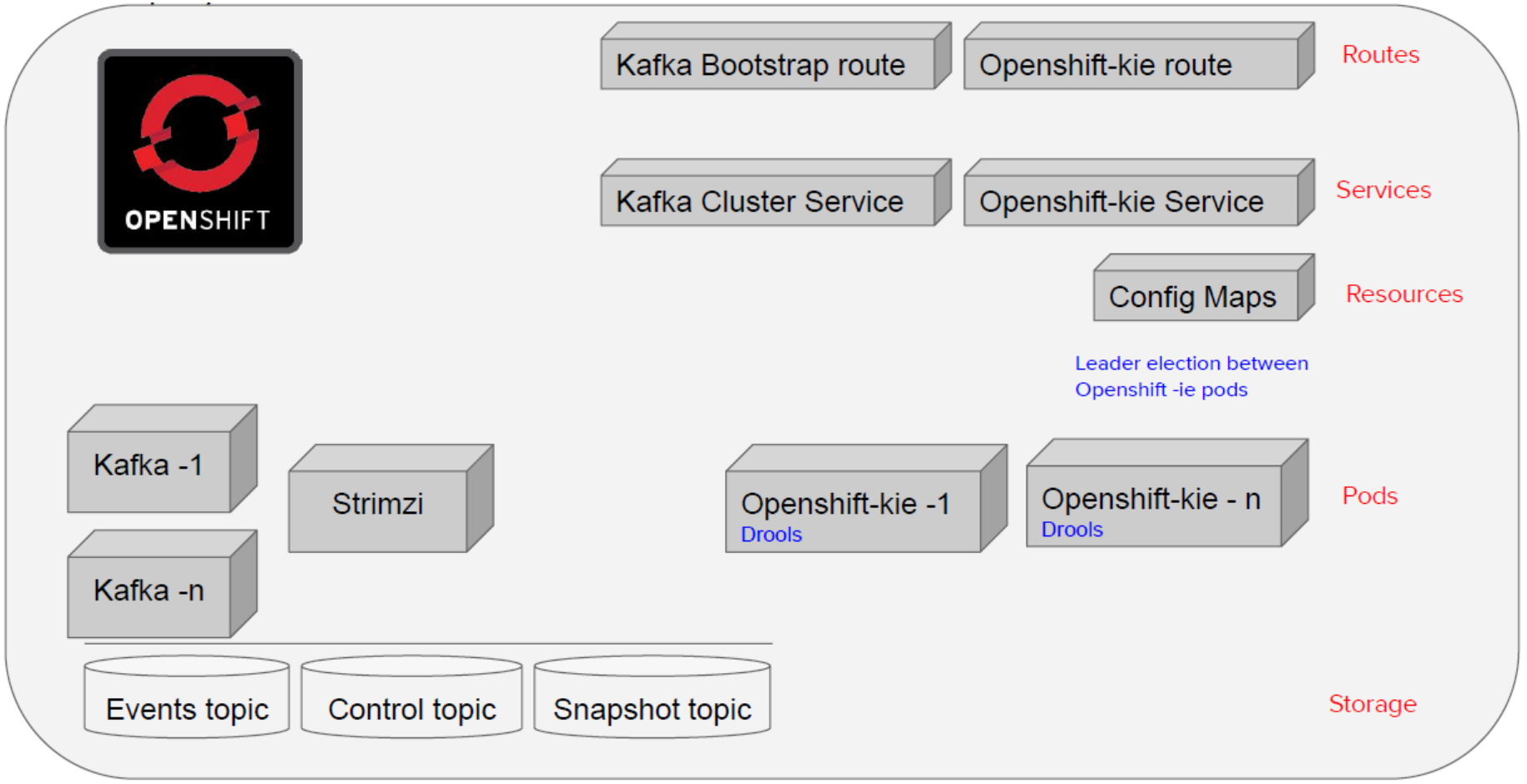
Leader - Replica Coordination



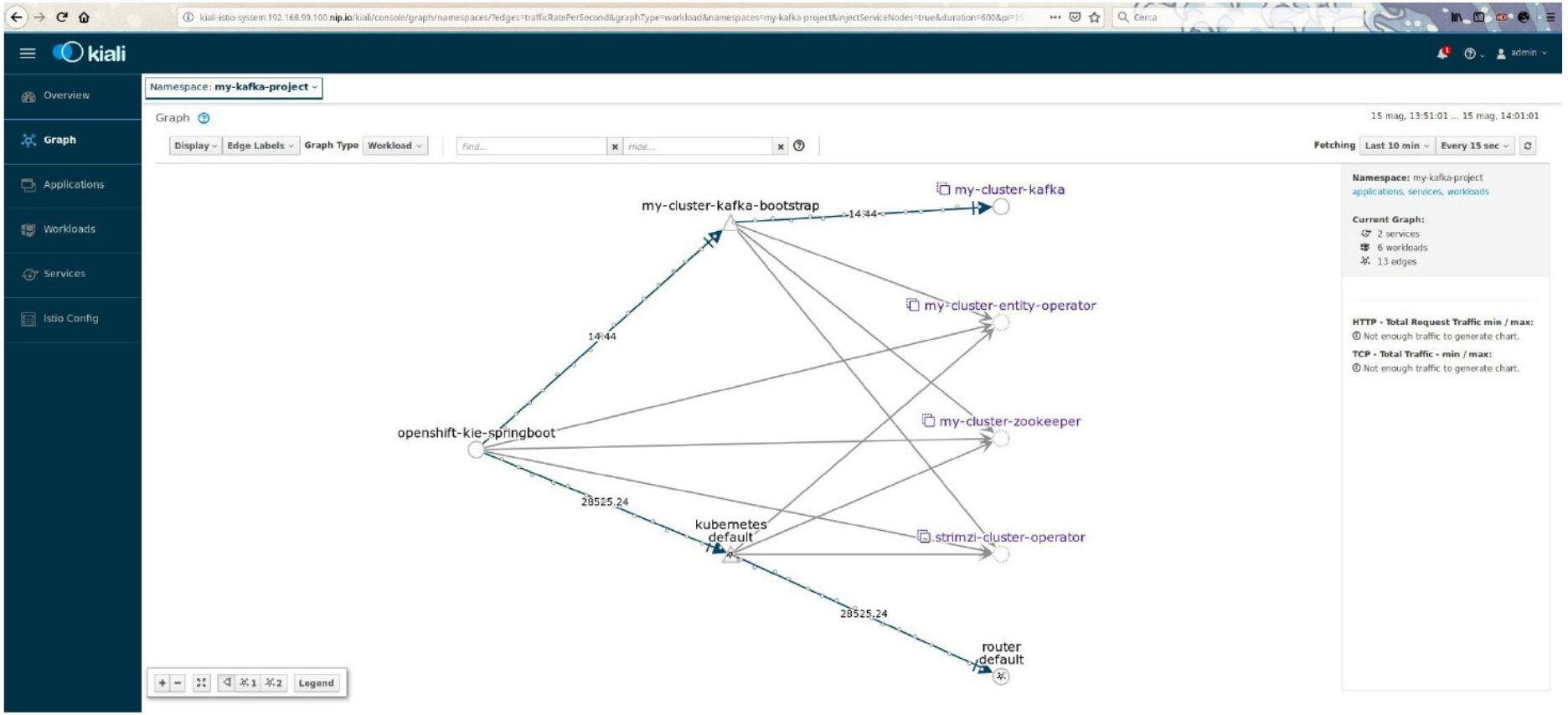
Leader - Replica Coordination



Deployment


OPENSHIFT
Kafka Bootstrap route
Openshift-kie route
Routes
Kafka Cluster Service
Openshift-kie Service
Services
Config Maps
Resources
Leader election between
Openshift -ie pods
Kafka -1
Kafka -n
Strimzi
Openshift-kie -1
Drools
Openshift-kie - n
Drools
Pods
Events topic
Control topic
Snapshot topic
Storage

Monitoring


kiali
Overview
Graph
Applications
Workloads
Services
Istio Config
Namespace: my-kafka-project
Graph
Display
Edge Labels
Graph Type
Workload
Fetching
Last 10 min
Every 15 sec
my-cluster-kafka
my-cluster-kafka-bootstrap
14.44
my-cluster-entity-operator
14.44
openshift-kie-springboot
my-cluster-zookeeper
28525.24
kubernetes
default
strimzi-cluster-operator
28525.24
router
default
Legend
Namespace: my-kafka-project
applications, services, workloads
Current Graph:
2 services
6 workloads
13 edges
HTTP - Total Request Traffic min / max:
Not enough traffic to generate chart.
TCP - Total Traffic - min / max:
Not enough traffic to generate chart.

Demo

Q & A

Reference & Contacts

https://github.com/kiegroup/openshift-drools-hacep

https://twitter.com/desmax74

http://www.slideshare.net/desmax74

https://www.linkedin.com/in/desmax74

https://github.com/desmax74/

Thanks for your attention !