Finding a work-load balance

Cruise Control for Kafka on Kubernetes

Kyle Liberti
Software Engineer

Paolo Patierno
Principal Software Engineer

Red Hat

Principal Software Engineer @Red Hat
Working on Apache Kafka and Strimzi

@ppatierno

Software Engineer @Red Hat
Working on Apache Kafka and Strimzi

Kyle Liberti

Overview

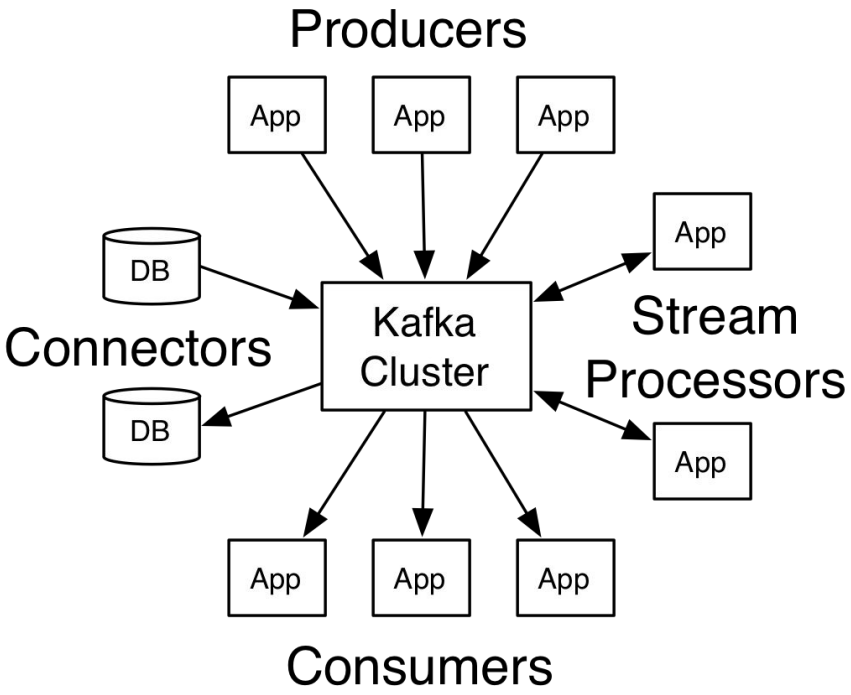
1) Intro to Kafka
2) Managing Kafka clusters using Strimzi
3) Intro to Kafka Cruise Control
4) Balancing Kafka clusters using Strimzi

What is Kafka?

Red Hat

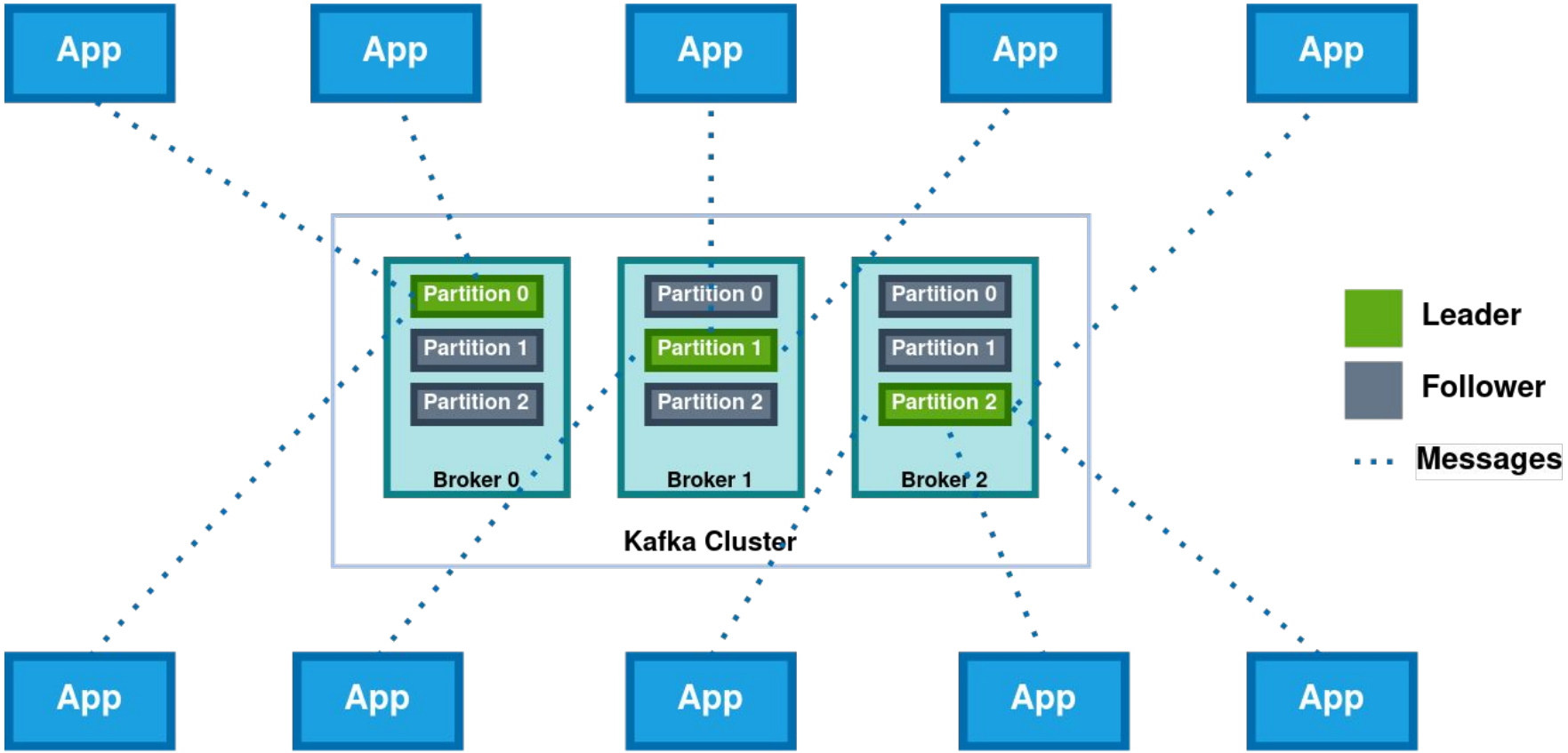
Kafka

- Distributed messaging system
- Scalable and fault tolerant
- Integrated ecosystem



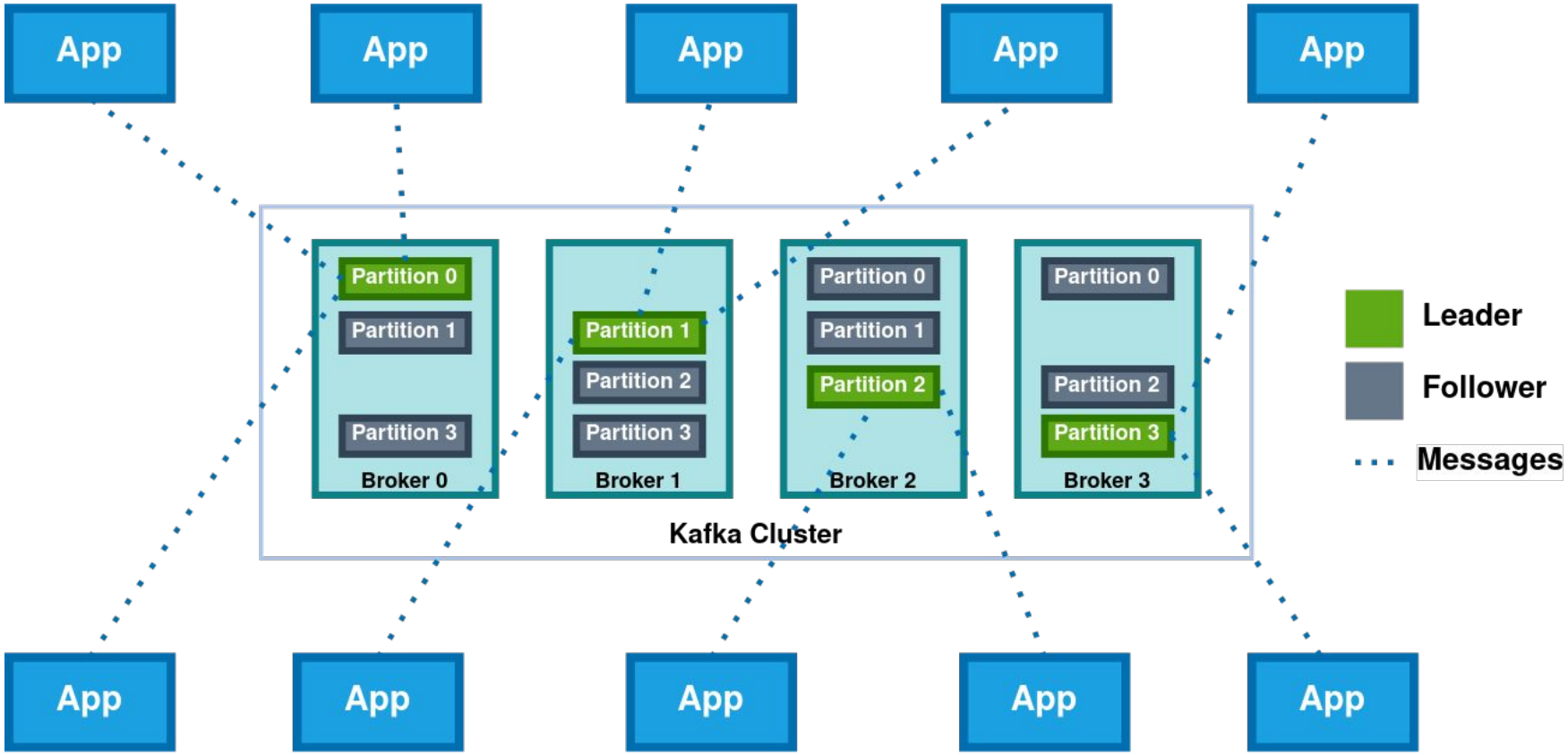
Red Hat

Simplified Kafka Architecture



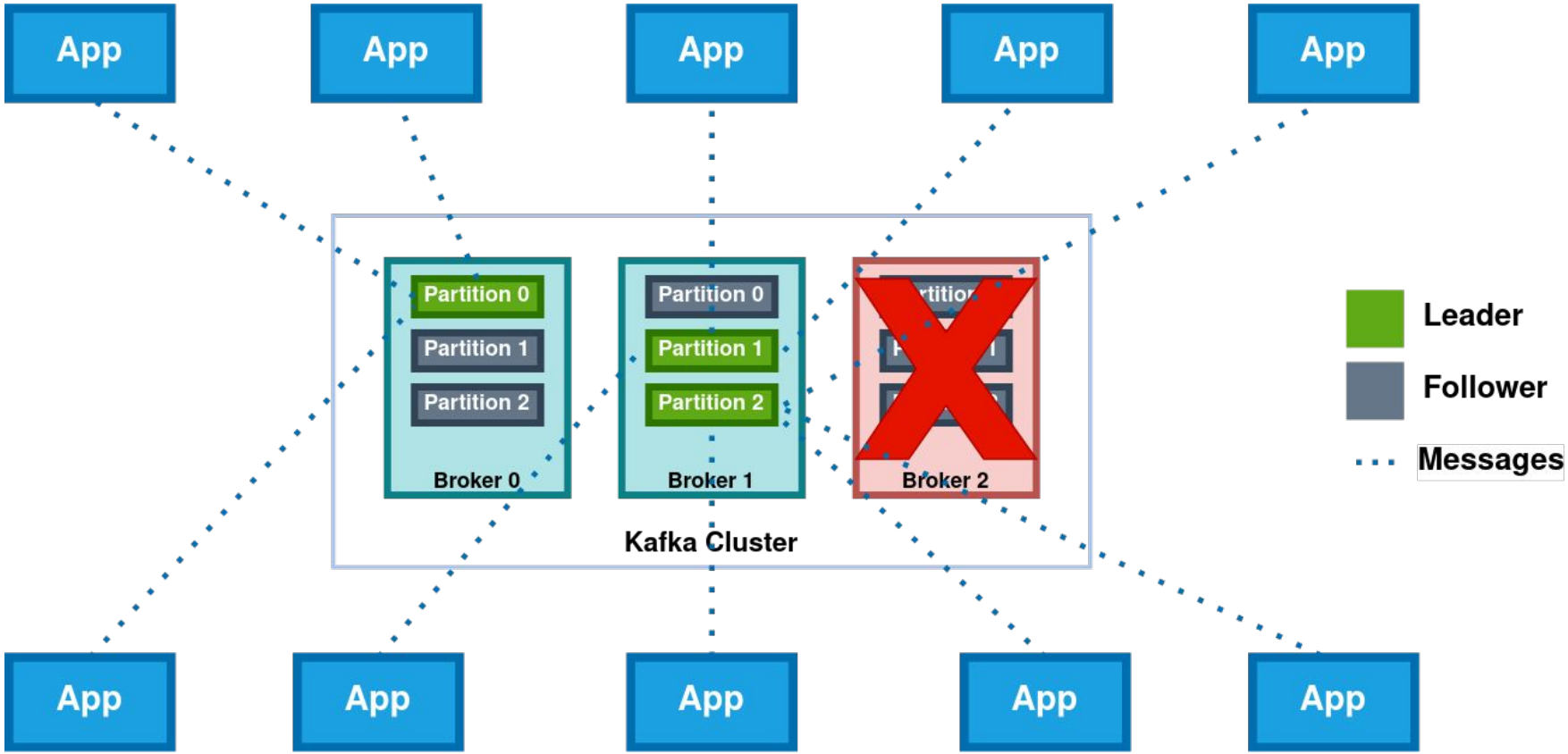
Red Hat

Kafka scales horizontally



Red Hat

Kafka is fault tolerant



Red Hat

BUT...

Kafka is **HARD**

- Configuration
- Operation
- Development

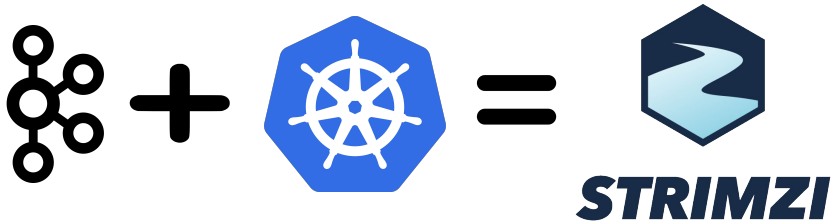
Red Hat

IS THERE AN EASIER WAY?

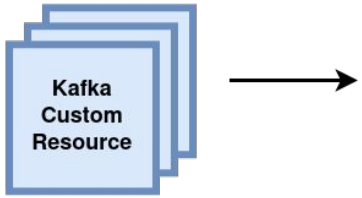
Strimzi

Operator for running Kafka on Kubernetes

- Automated configuration and deployment
- Built-in security
- Simple user interface



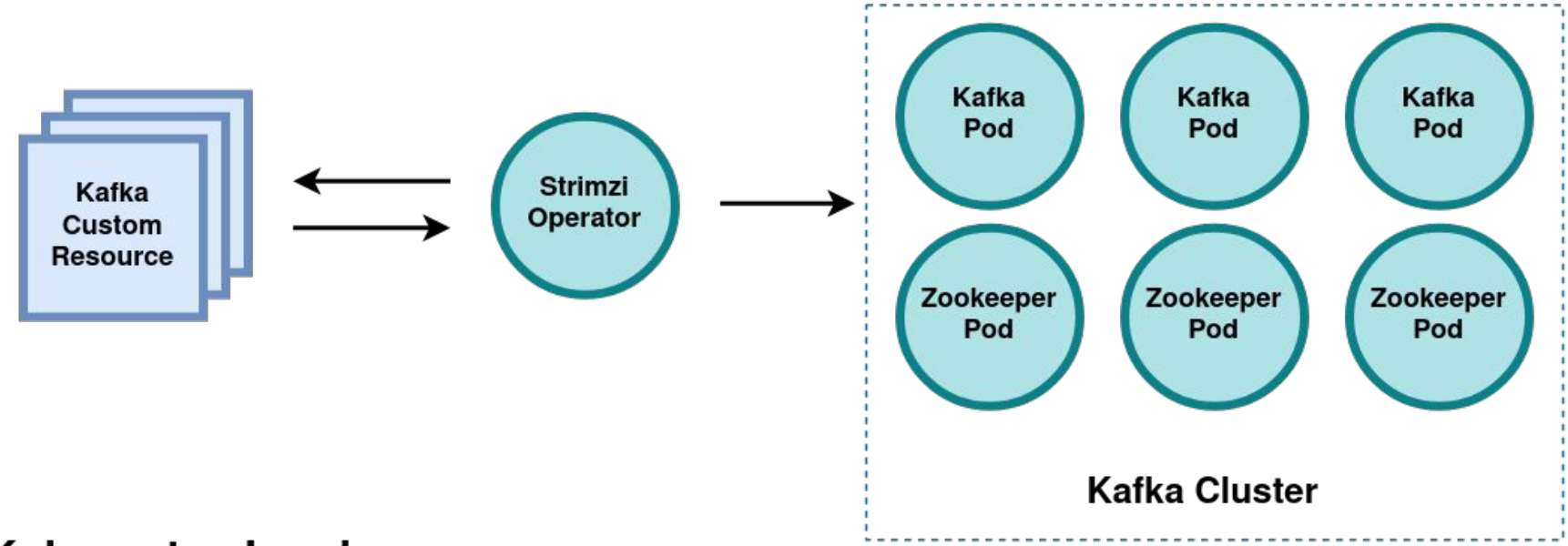
Strimzi Custom Resources




```
apiVersion: kafka.strimzi.io/v1beta2
kind: Kafka
metadata:
    name: my-cluster
spec:
    kafka:
        version: 2.8.0
        replicas: 3
        config:
            offsets.topic.replication.factor: 3
            log.message.format.version: "2.8"
            inter.broker.protocol.version: "2.8"
        storage:
            type: ephemeral
    zookeeper:
        replicas: 3
        storage:
            type: ephemeral
```

Red Hat

Simplified Strimzi Architecture



NOT OUT OF THE WOODS YET...

Unbalanced Kafka brokers

- Uneven distribution of load across brokers

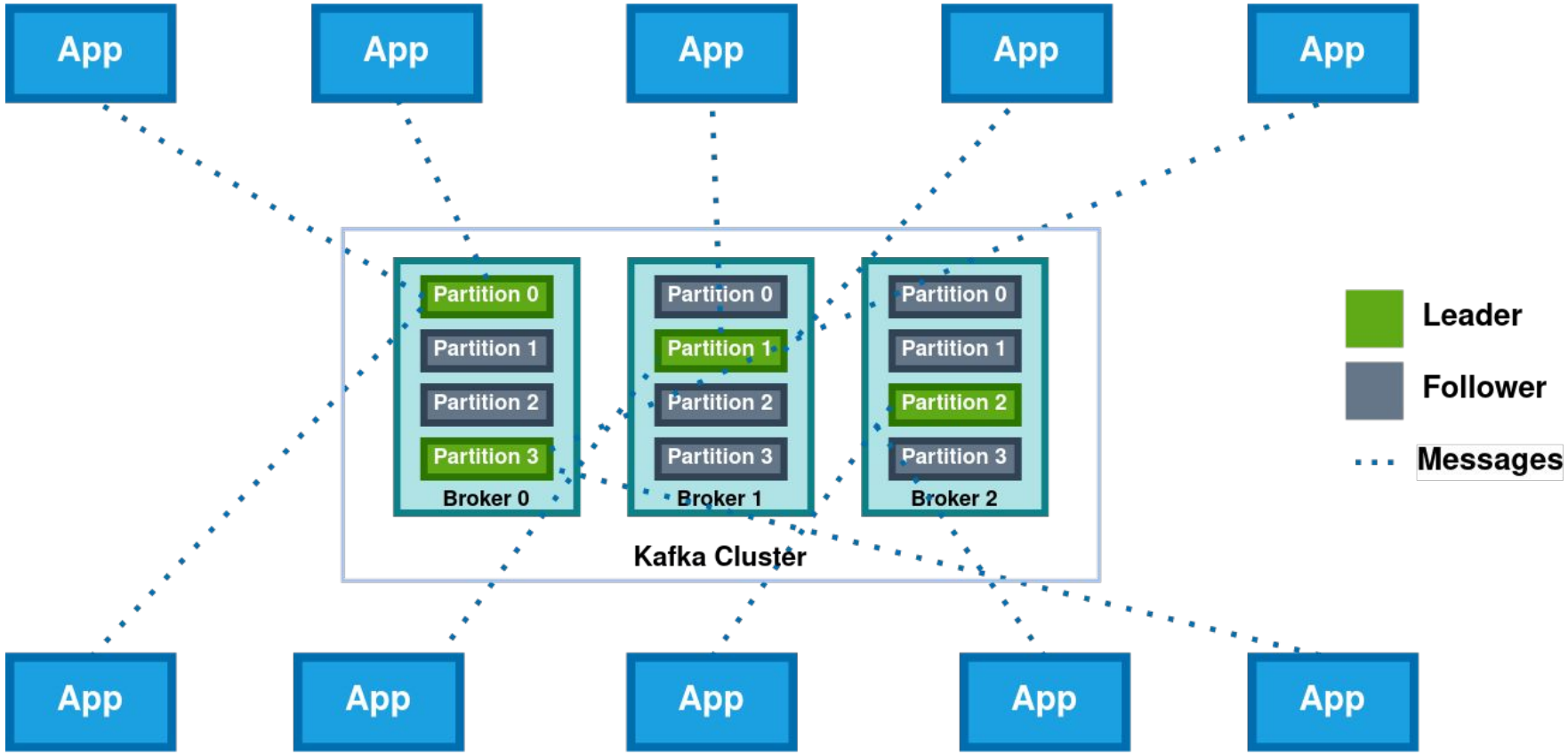
---> **Poor performance**

---> **Storage and Network are badly utilized**

---> **More load on some brokers**

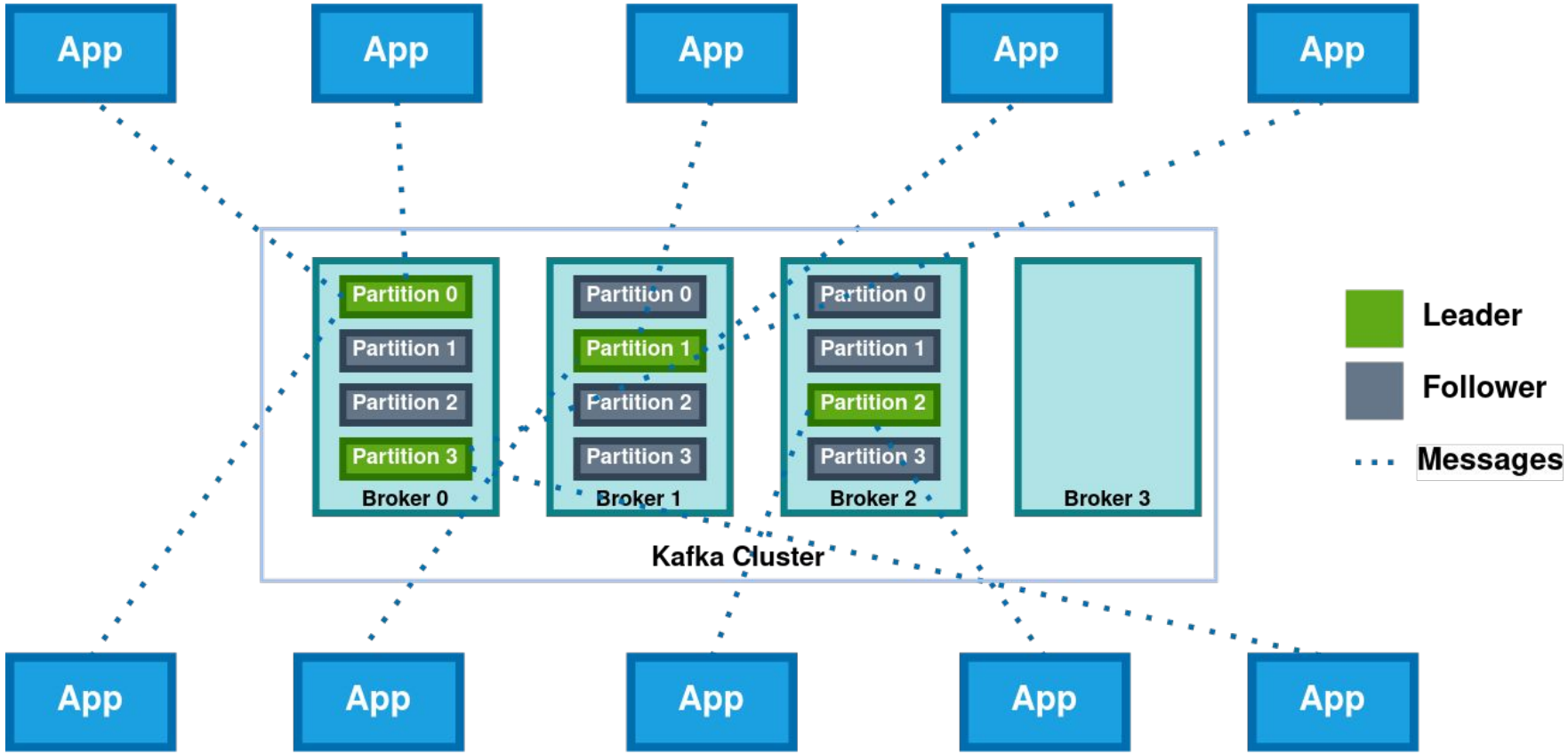
Red Hat

Simplified Kafka Architecture



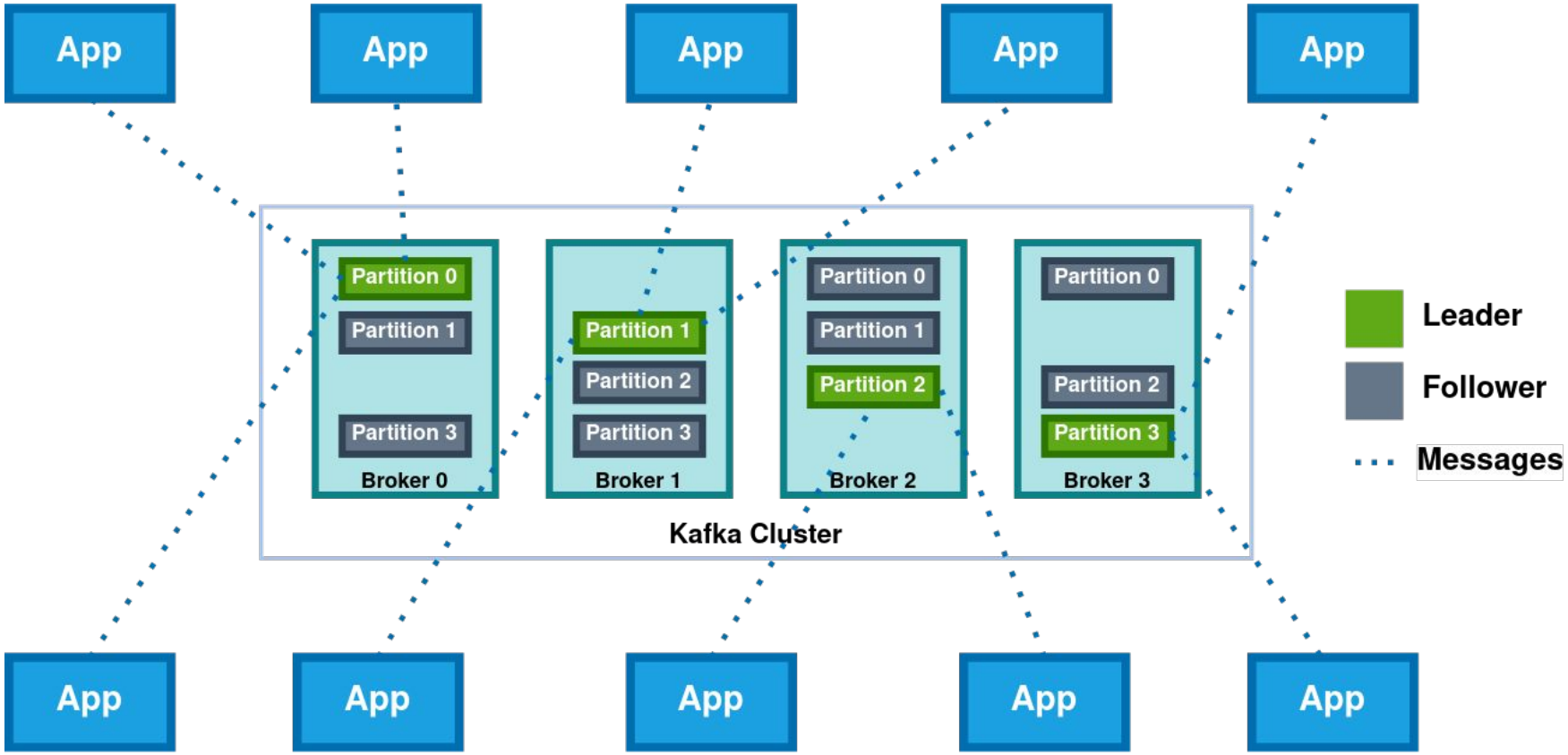
Red Hat

Unbalanced Kafka Cluster



Red Hat

Balanced Partition Distribution



Red Hat

WHAT CAN WE DO ABOUT IT?

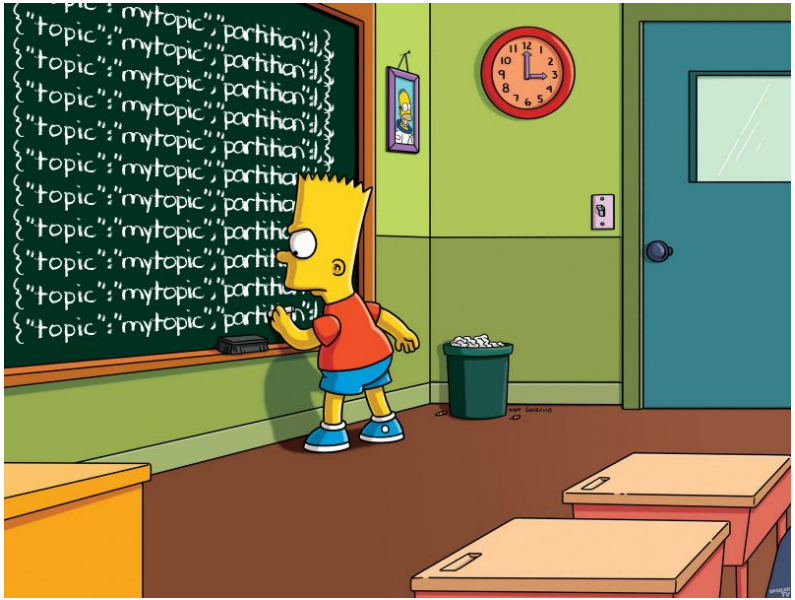
Balancing Partitions is **HARD**

- Moving target
- Optimization

Red Hat

Old School Rebalancing

- kafka-reassign-partitions.sh
- Manual process
- One dimensional balancing



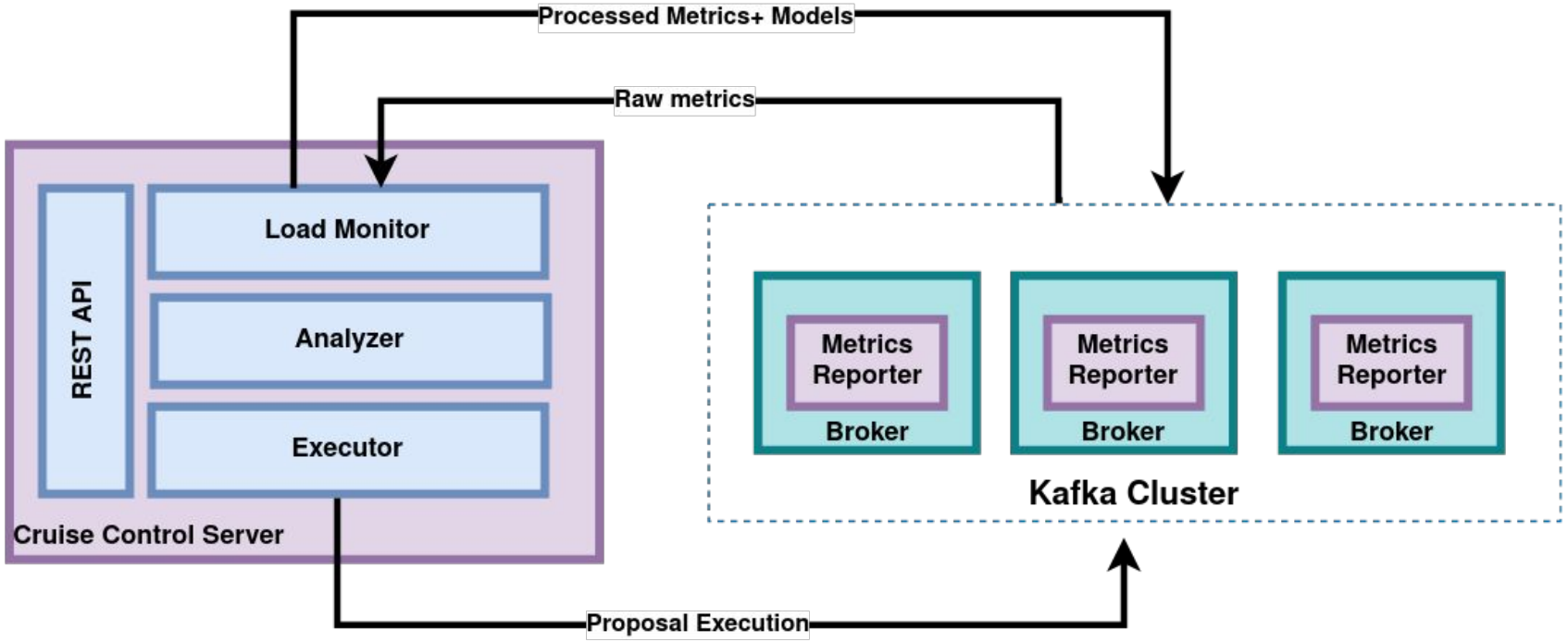
Red Hat

CAN WE DO BETTER?

Cruise Control

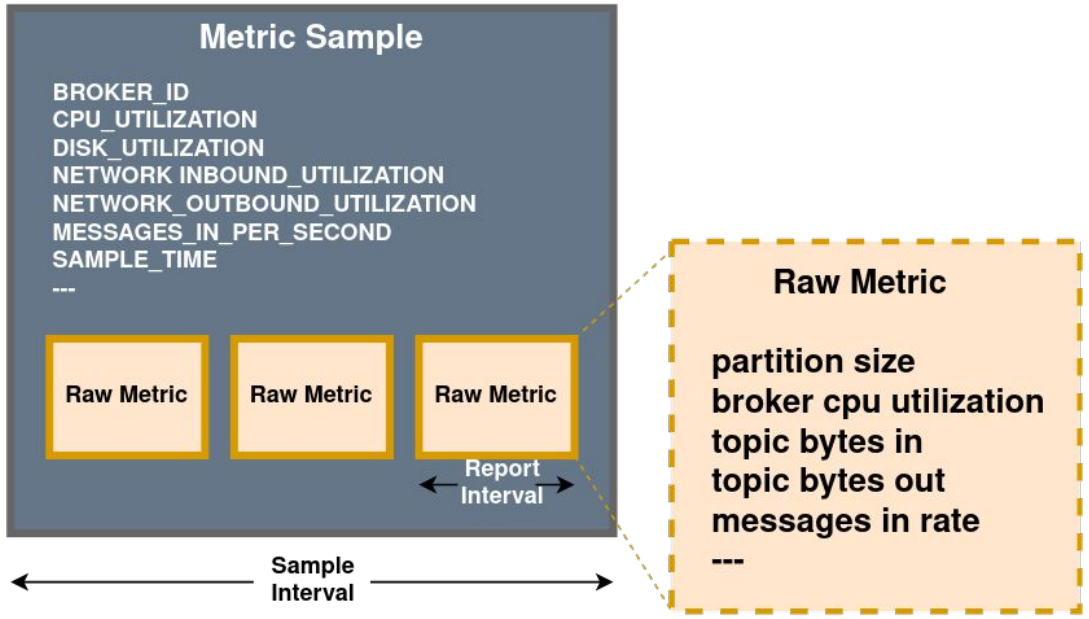
- Fully automated rebalancing
- Fine-grained resource tracking
- Multidimensional balancing

Simplified Cruise Control Architecture



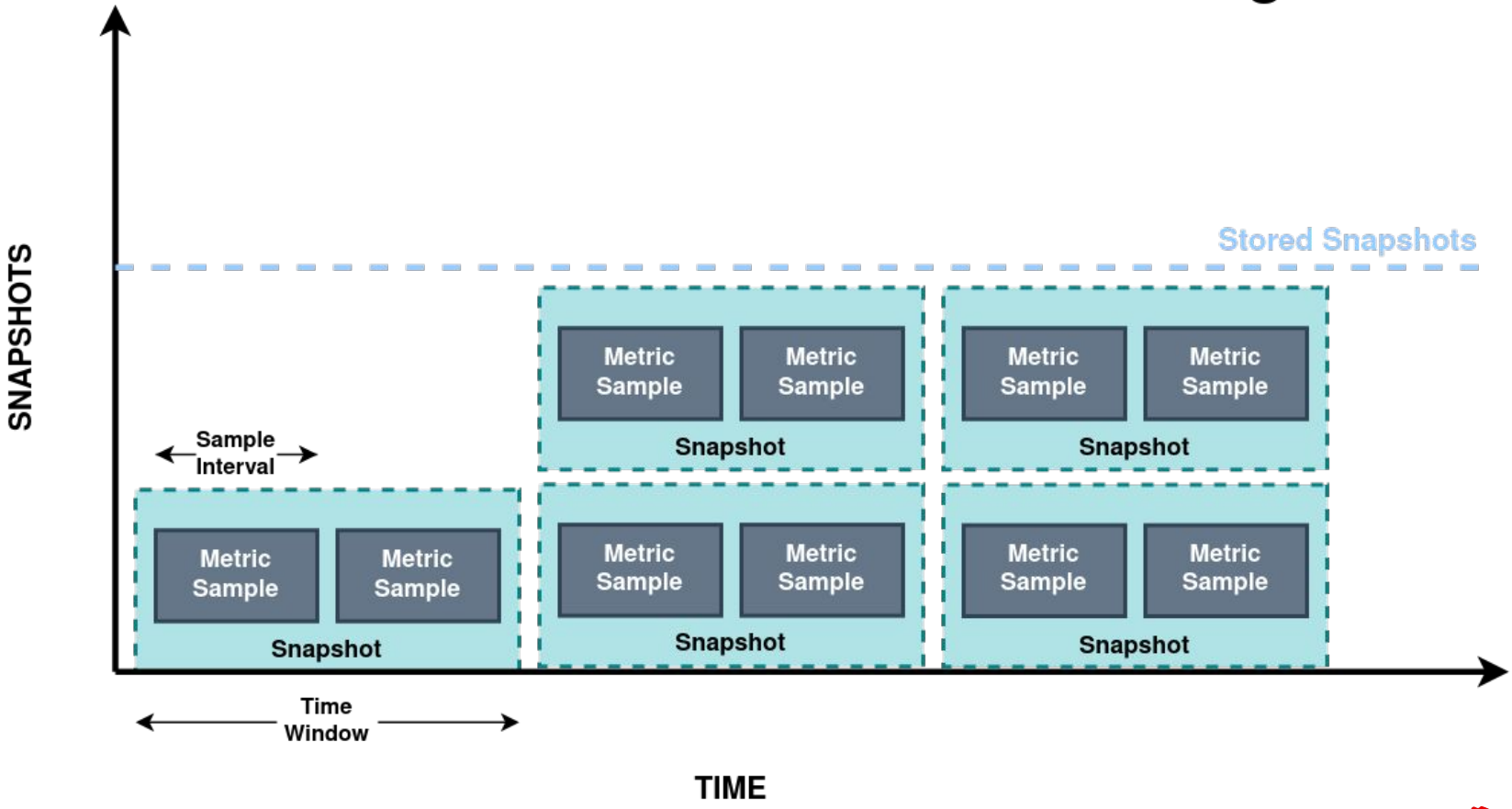
Metric Samples

- Processed Kafka broker metrics
- Configurable sampling intervals
- Stored back into Kafka



Red Hat

Load Monitor Metrics Processing



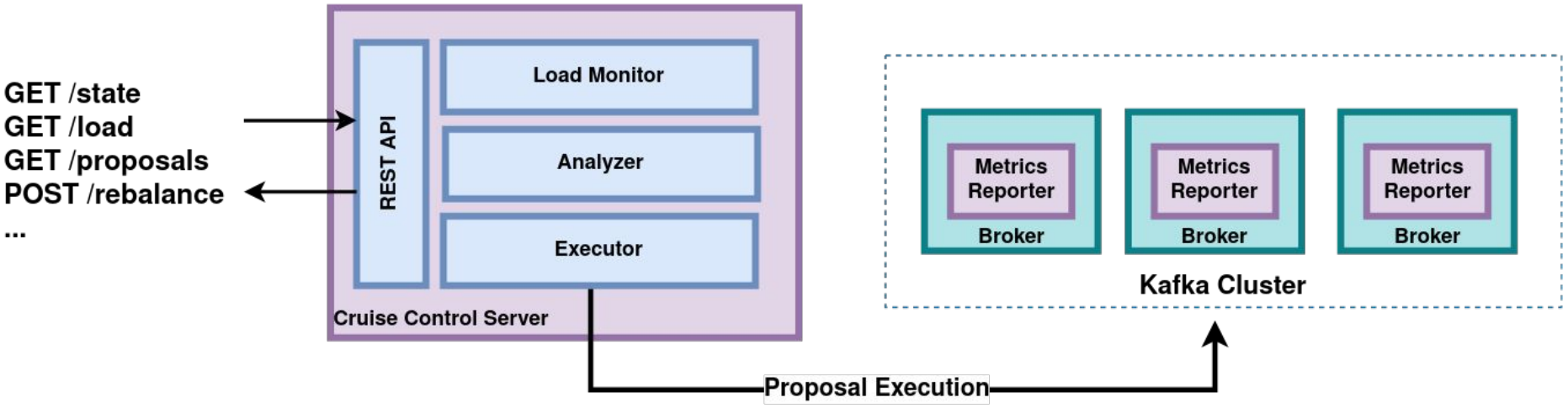
Red Hat

Cluster Workload Model

- Workload data of cluster resources
- Used to simulate partition movements

Broker	Topic/Partition		Disk/CPU		Network Rate	
Host	#Replicas	#Leaders	Disk	CPU	IN	OUT
my-cluster-broker-0	24	24	100 GB	8.05%	5 MBps	8 MBps
my-cluster-broker-1	22	22	56 GB	9.60%	6 MBps	12 MBps
my-cluster-broker-2	19	19	76 GB	8.52%	2 MBps	9 MBps

Cruise Control REST API

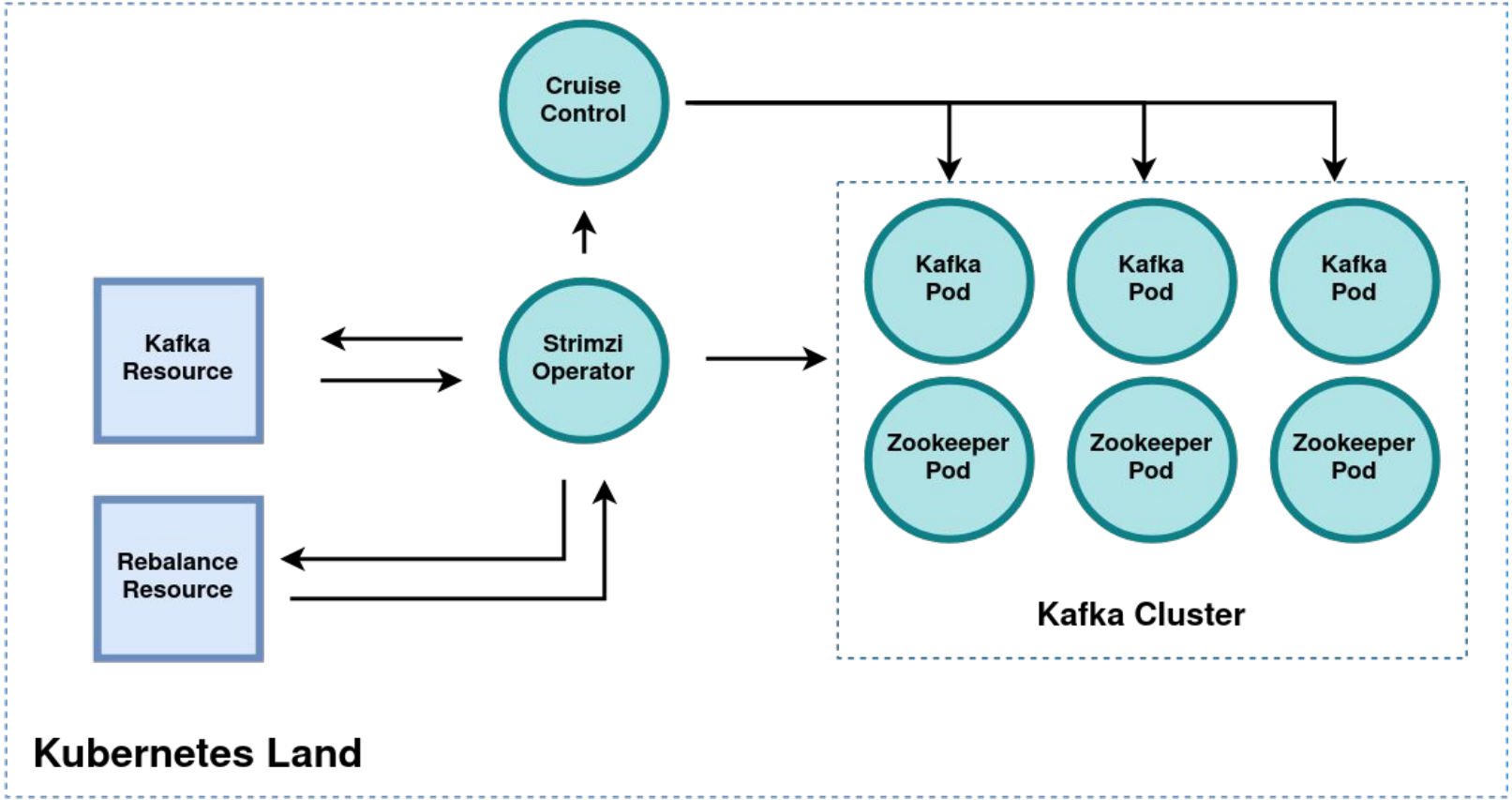


Cruise Control

Integration with Strimzi

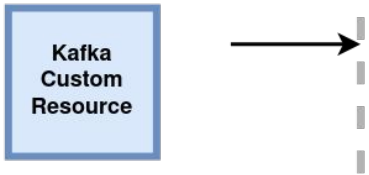
- Automated deployment
- Simple user interface

Strimzi Architecture with Cruise Control



Red Hat

Kafka Resource with Cruise Control




```
apiVersion: kafka.strimzi.io/v1beta2
kind: Kafka
metadata:
    name: my-cluster
spec:
    kafka:
     ...
    cruiseControl:
        config:
            default.goals: >
              <path>.RackAwareGoal,
              <path>.ReplicaCapacityGoal
               ...
            cpu.balance.threshold: 1.1
            metadata.max.age.ms: 300000
            send.buffer.bytes: 131072
        brokerCapacity:
            disk: 100Gi
            cpuUtilization: 100
            inboundNetwork: 10000KiB/s
            outboundNetwork: 10000KiB/s
```

Red Hat

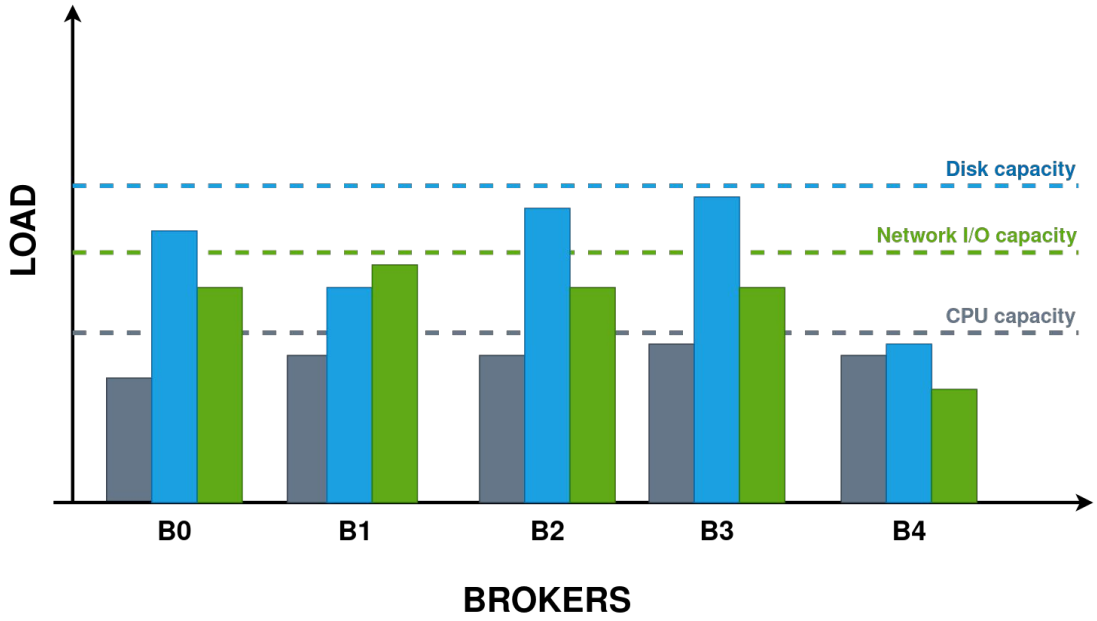
Cruise Control Goals

- Analyzer proposes load optimizations using selected goals
- Goals cover different dimensions of a rebalance
  - Rack awareness
  - Replica capacity
  - Resource capacity/utilization
- The combination and priorities of goals is customizable

Cruise Control Capacity Configurations

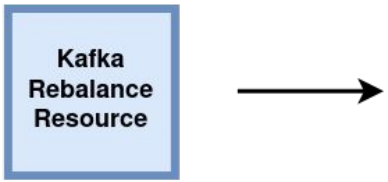
Configurable capacity limits for:

- Disk usage
- CPU utilization
- Network throughput



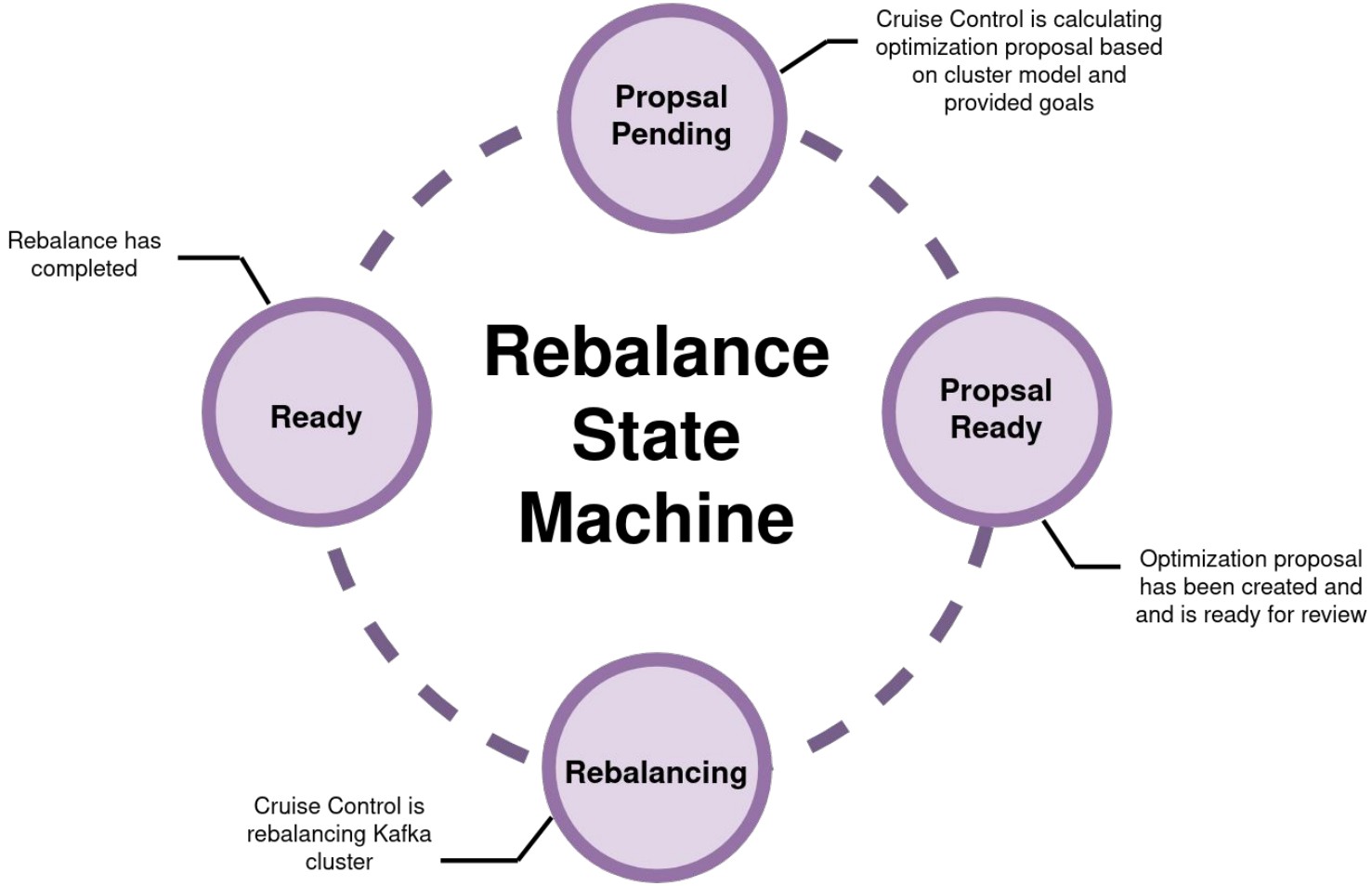
Red Hat

Kafka Rebalance Resource



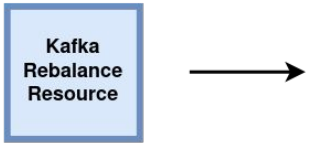

```
apiVersion: kafka.strimzi.io/v1beta2
kind: KafkaRebalance
metadata:
    name: my-rebalance
    labels:
        strimzi.io/cluster: my-cluster
spec:
     goals:
         - CpuCapacityGoal
         - NetworkInboundCapacityGoal
         - DiskCapacityGoal
         - RackAwareGoal
         - MinTopicLeadersPerBrokerGoal
         - NetworkOutboundCapacityGoal
         - ReplicaCapacityGoal
```

Red Hat


Rebalance State Machine
Propsal Pending
Cruise Control is calculating optimization proposal based on cluster model and provided goals
Propsal Ready
Optimization proposal has been created and and is ready for review
Rebalancing
Cruise Control is rebalancing Kafka cluster
Ready
Rebalance has completed

Red Hat

Kafka Rebalance Status




```
Status:
  Conditions:
    Last Transition Time: 2020-05-19T13:50:12.533Z
    Status: ProposalReady
    Type: State
  Observed Generation: 1
  Optimization Result:
    Data To Move MB: 0
    Excluded Brokers For Leadership:
    Excluded Brokers For Replica Move:
    Excluded Topics:
    Intra Broker Data To Move MB: 0
    Monitored Partitions Percentage: 100
    Num Intra Broker Replica Movements: 0
    Num Leader Movements: 0
    Num Replica Movements: 26
    On Demand Balancedness Score After: 81.8666802863978
    On Demand Balancedness Score Before: 78.01176356230222
    Recent Windows: 1
    Session Id: 05539377-ca7b-45ef-b359-e13564f1458c
```

Red Hat

Red Hat

DEMO

What's Next?

Cruise Control Integration Roadmap

- API authentication + authorization
- Intra broker data balancing
- Changing topic replication factor

Red Hat

Questions?

Thank You!


linkedin.com/company/red-hat


youtube.com/user/RedHatVideos


facebook.com/redhatinc


twitter.com/RedHat

Red Hat