Cruise Control: Effortless Management of Kafka Clusters

Adem Efe Gencer

Senior Software Engineer
LinkedIn

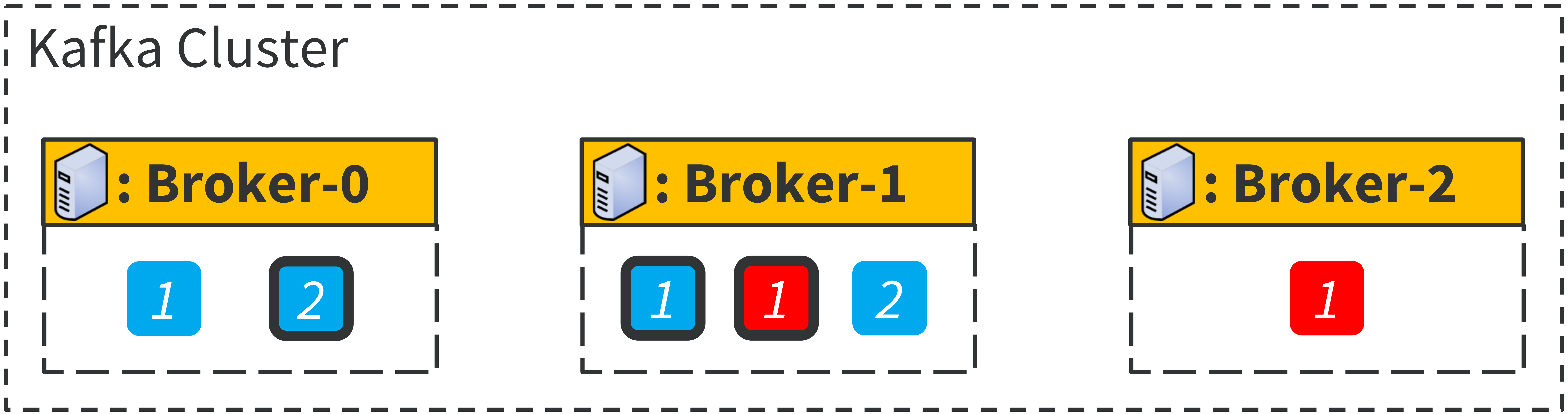
Kafka: A Distributed Stream Processing Platform

: High throughput & low latency

: Message persistence on partitioned data

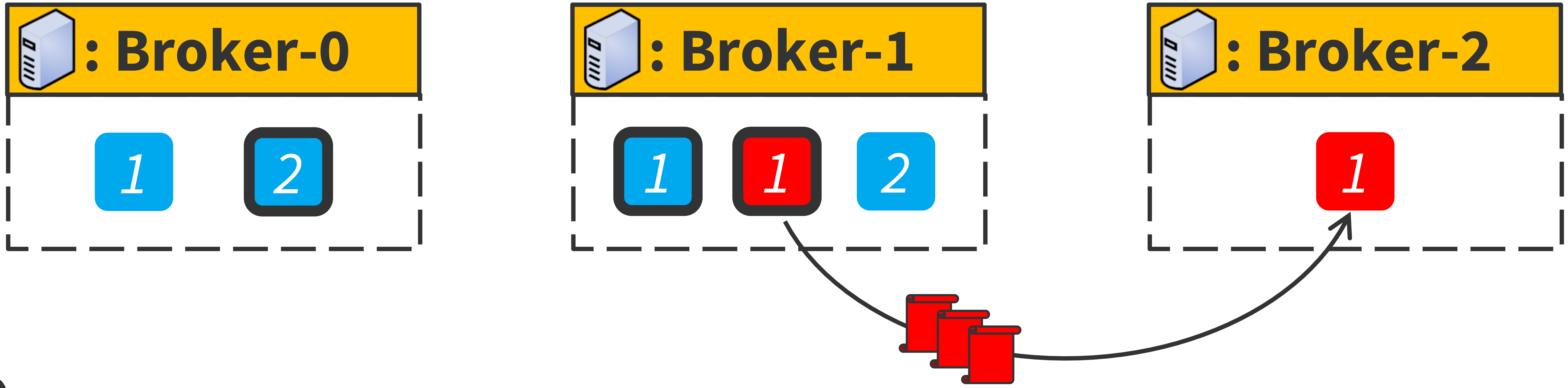
: Total ordering within each partition

Key Concepts: Brokers, Topics, Partitions, and Replicas



1 : A Replica of Partition-1 of Blue Topic

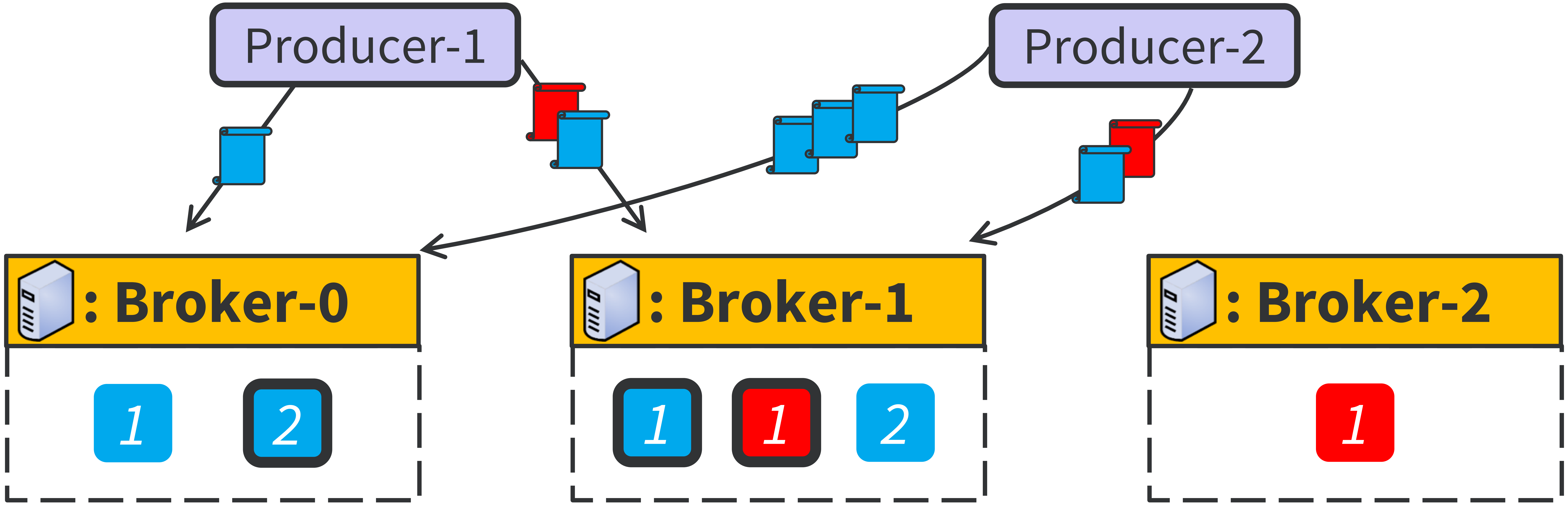
Key Concepts: Leaders and Followers



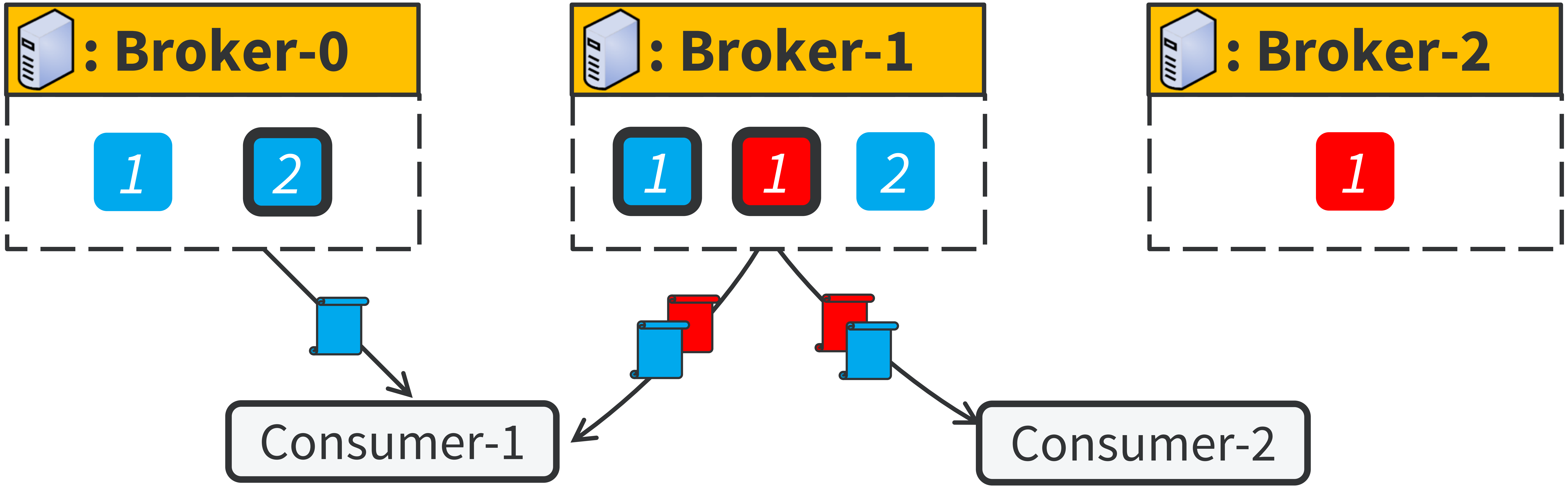
1 : The Leader Replica

1 : A Follower Replica

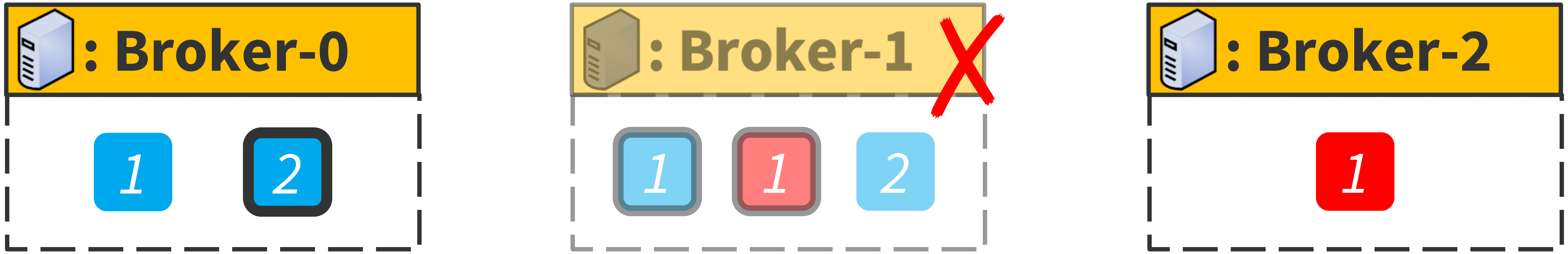
Key Concepts: Producers



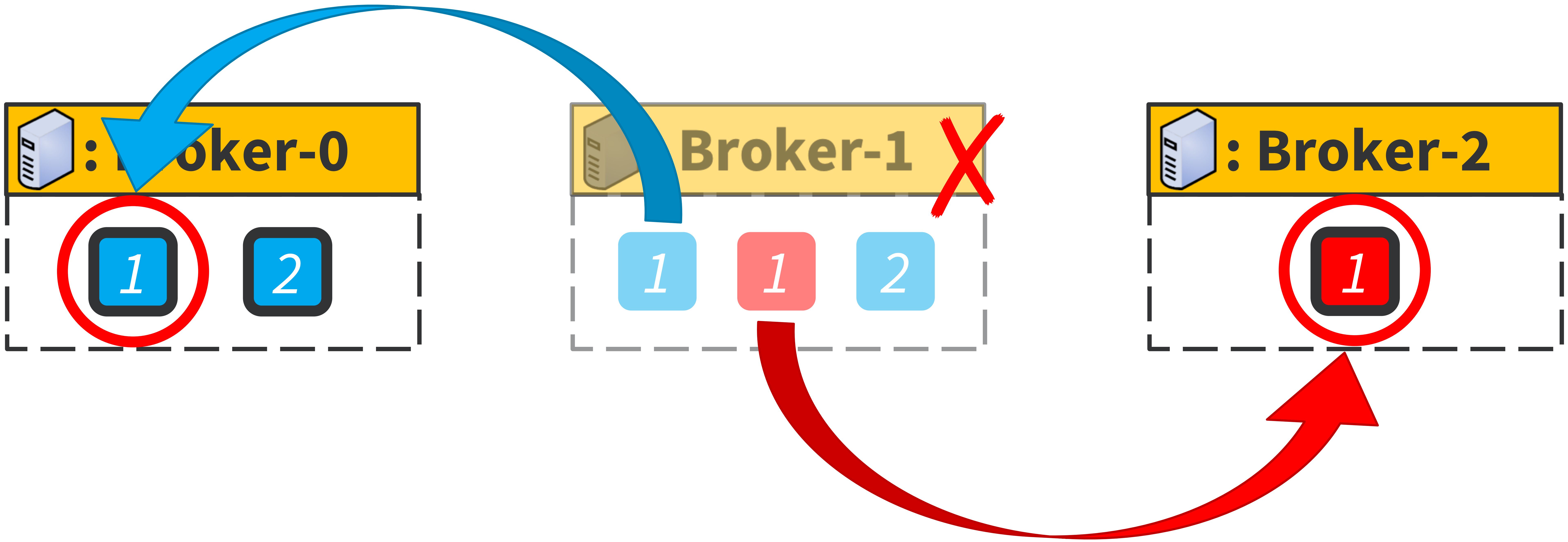
Key Concepts: Consumers



Key Concepts: Failover via Leadership Transfer

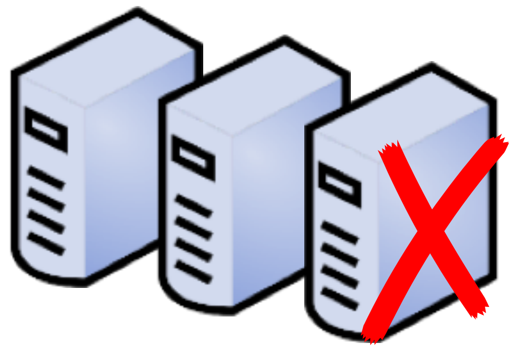


Key Concepts: Failover via Leadership Transfer



Kafka Incurs Management Overhead

: Large deployments – e.g. @LinkedIn: 2.6K+ Brokers, 44K+ Topics, 5M Partitions, 5T Messages / day



: Frequent hardware failures

: Load skew among brokers

: Kafka cluster expansion and reduction

“Elephant” (CC0): https://pixabay.com/en/elephant-safari-animal-defence-1421167, "Seesaw at Evelle" by Rachel Coleman (CC BY-SA 2.0): https://www.flickr.com/photos/rmc28/4862153119, “Inflatable Balloons” (Public Domain): https://commons.wikimedia.org/wiki/File:InflatableBalloons.jpg

Alleviating the Management Overhead

1. Admin Operations for Cluster Maintenance
2. Anomaly Detection with Self-Healing
3. Real-Time Monitoring of Kafka Clusters

1 Admin Operations for Cluster Maintenance

: Dynamically balance the cluster load

: Add / remove brokers

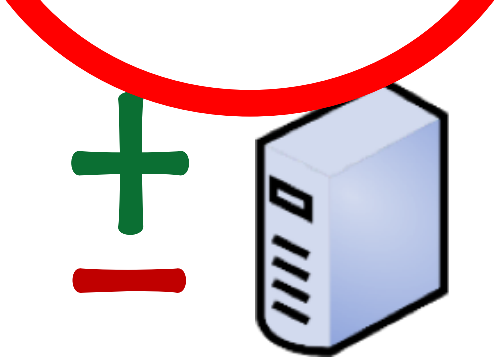
: Demote brokers – i.e. remove leadership of all replicas

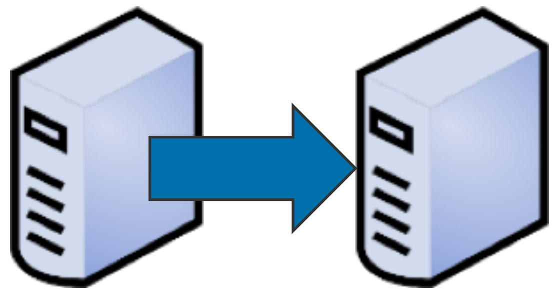
: Trigger preferred leader election

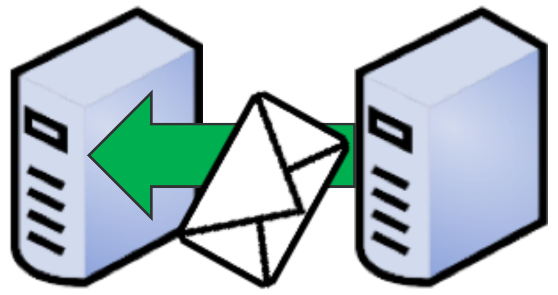
: Fix offline replicas

1 Admin Operations for Cluster Maintenance

: Dynamically balance the cluster load


: Add / remove brokers


: Demote brokers – i.e. remove leadership of all replicas


: Trigger preferred leader election

: Fix offline replicas

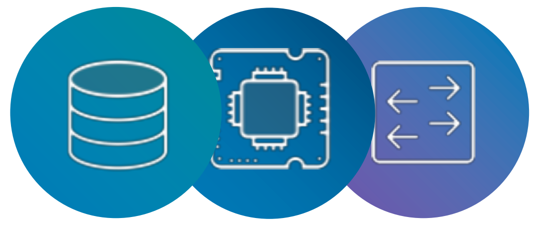
Dynamically Balance the Cluster Load

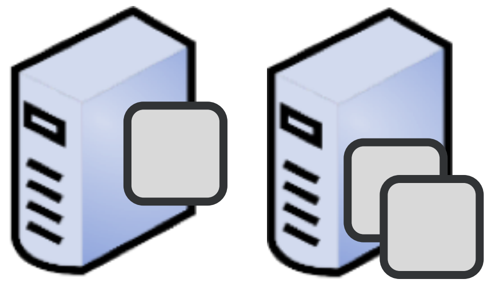
Must satisfy *hard goals*, including:

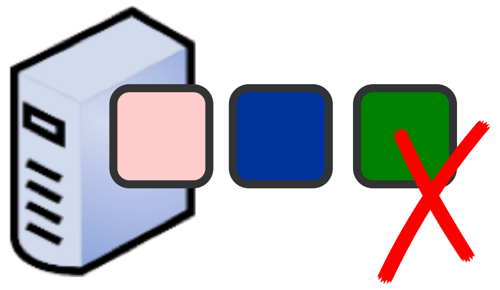
- : Guarantee rack-aware distribution of replicas
- : Never exceed the capacity of broker resources – e.g. disk, CPU, network bandwidth
- : Enforce operational requirements – e.g. maximum replica count per broker

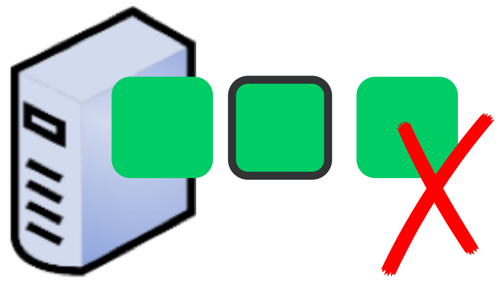
Dynamically Balance the Cluster Load

Satisfy *soft goals* as much as possible – i.e. best effort

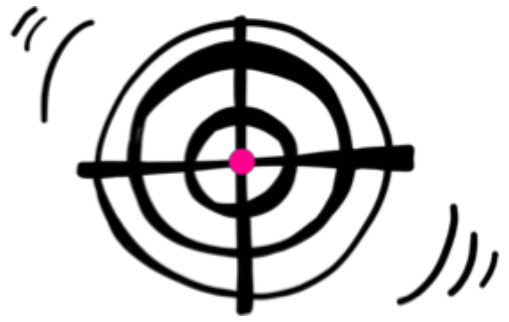

: Balance disk, CPU, inbound/outbound network traffic utilization of brokers


: Balance replica distribution

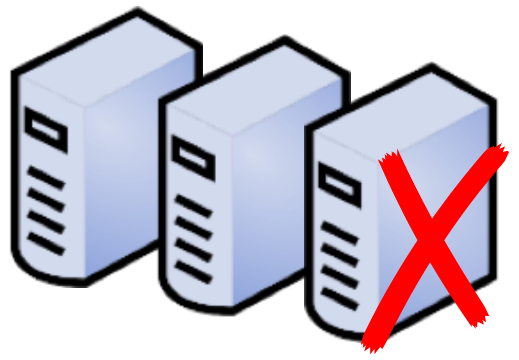

: Balance potential outbound network load


: Balance distribution of partitions from the same topic

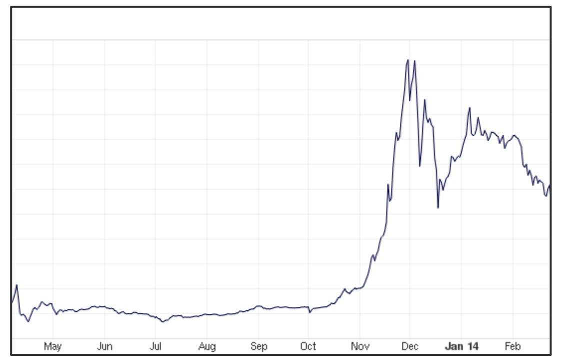
2 Anomaly Detection with Self-Healing



: Goal violation – rebalance cluster



: Broker failure – decommission broker(s)



: Metric anomaly – demote broker(s)

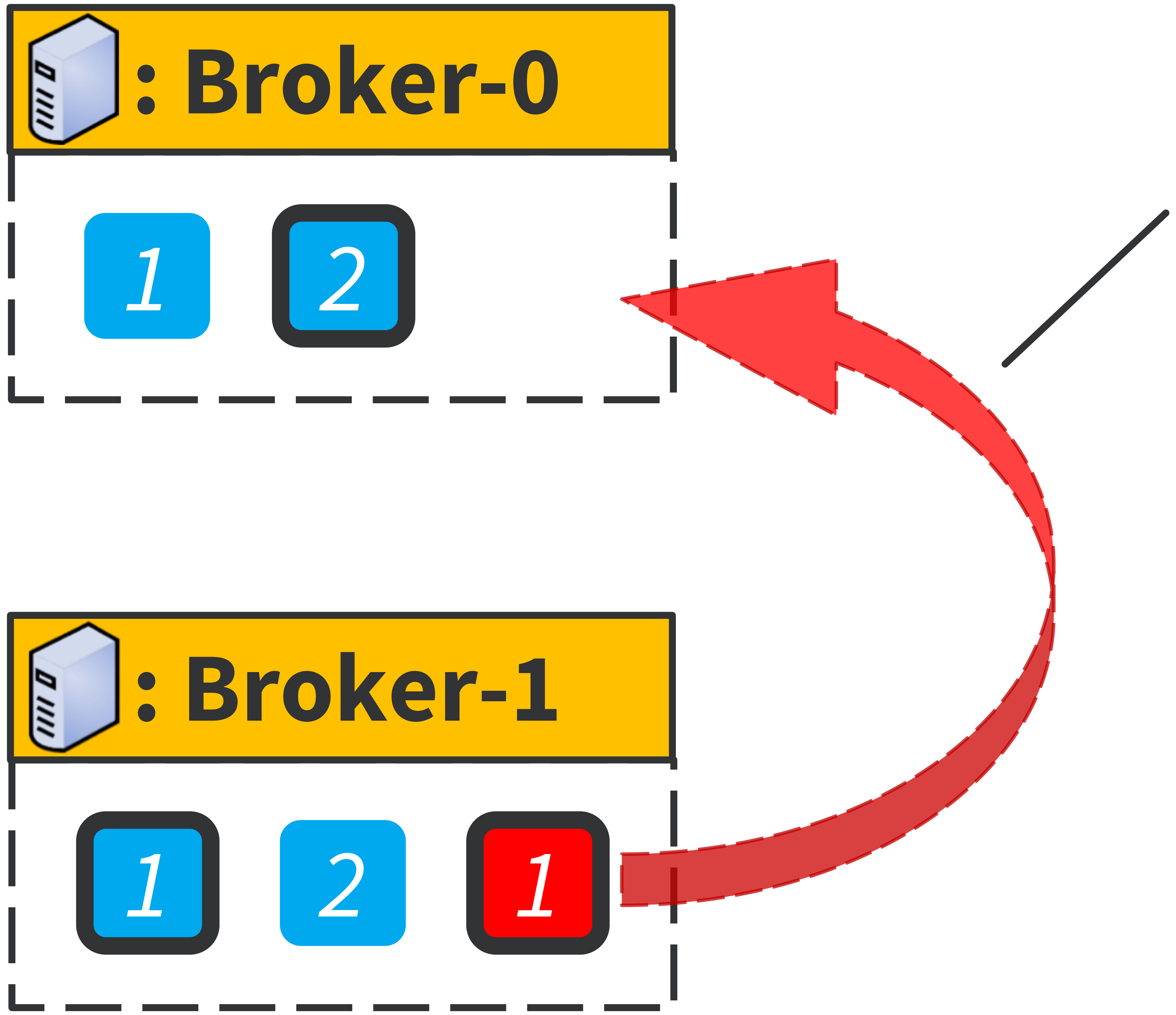
3 Real-Time Monitoring of Kafka Clusters

👁 : Examine the replica, leader, and load distribution

🌡 : Identify *under-replicated*, *under-min-ISR*, and *offline* partitions

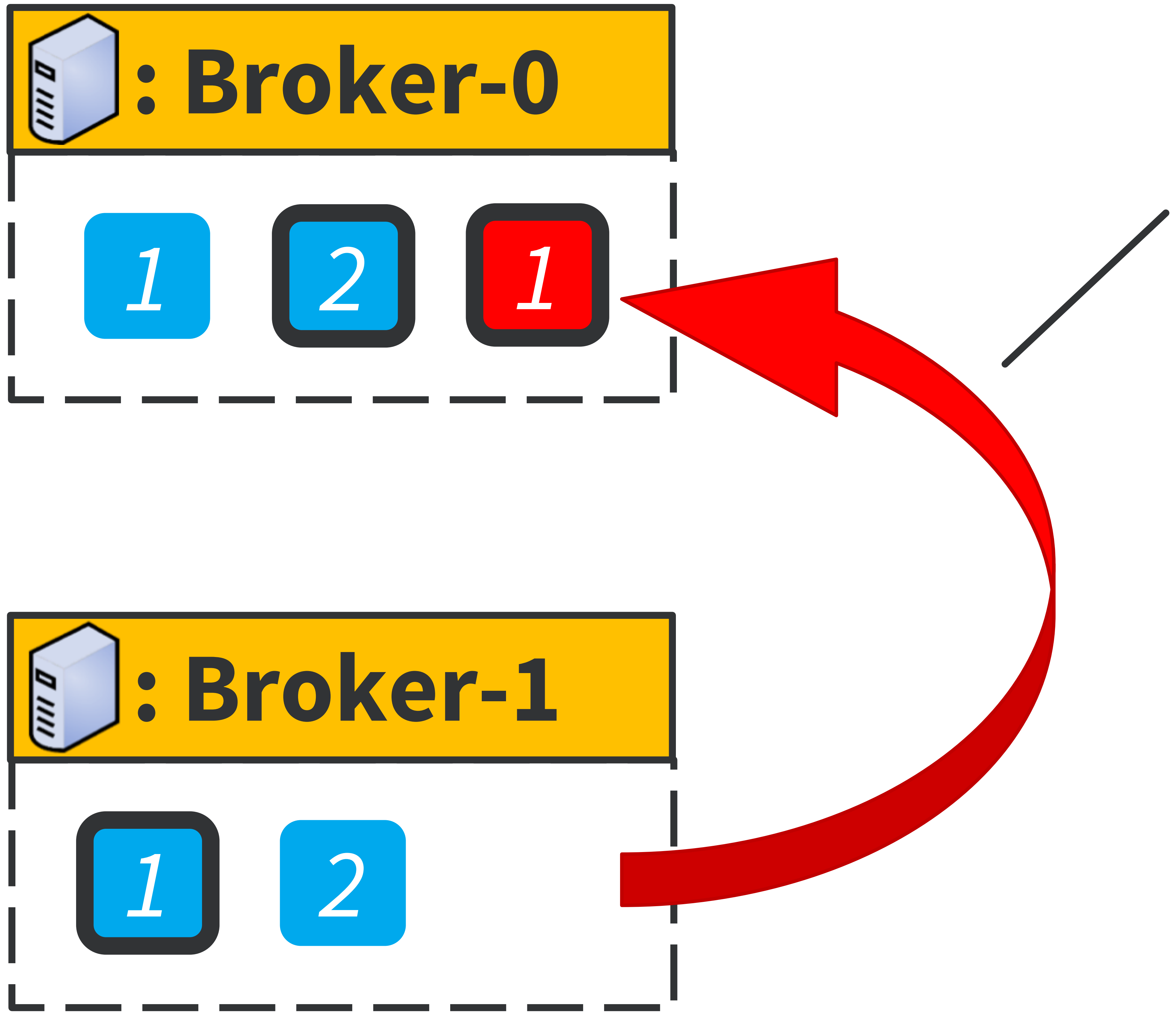
🔭 : Check the health of brokers, disks, and user tasks

Building Blocks of Management: Moving Replicas



Replica Move

Building Blocks of Management: Moving Replicas



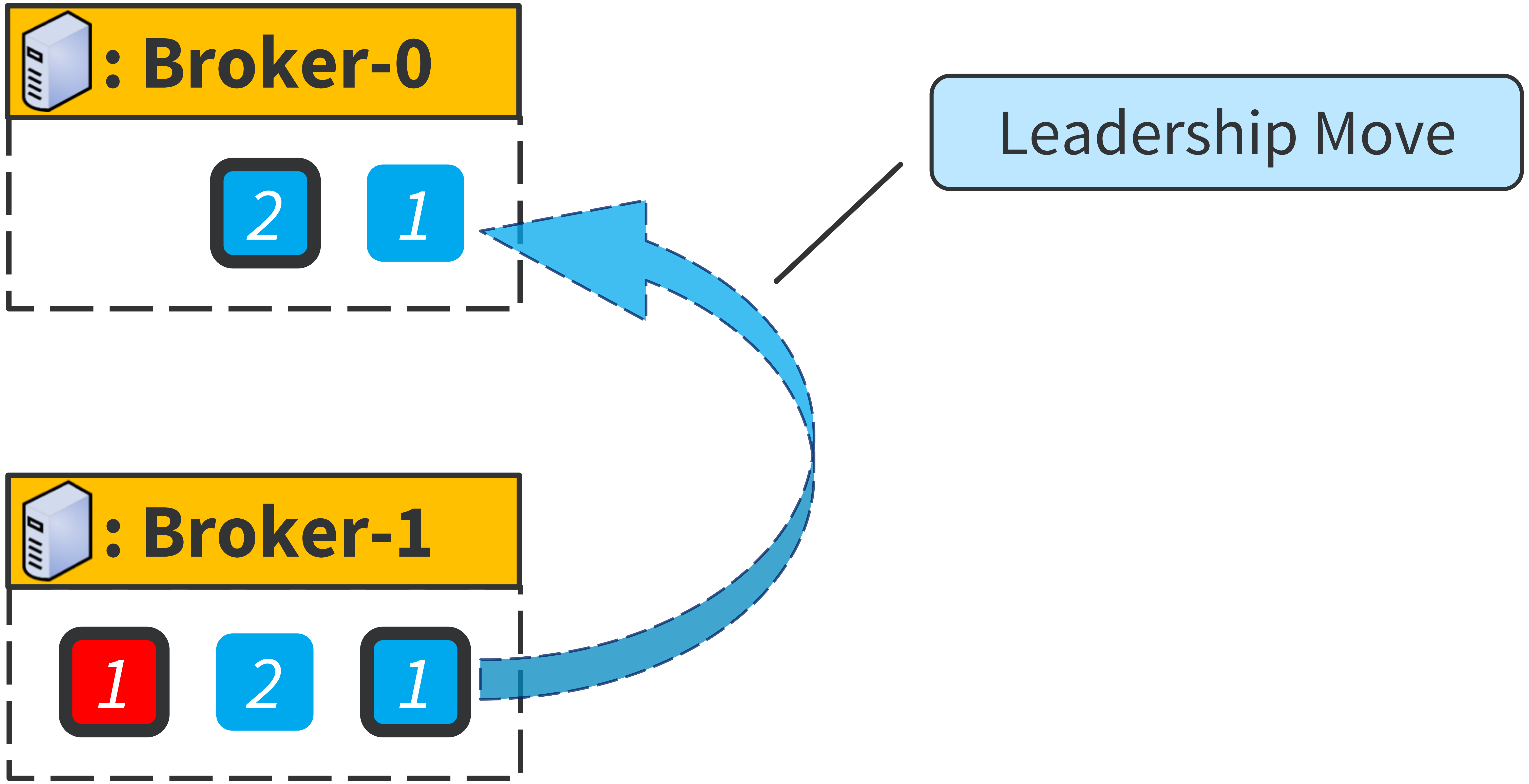
Replica Move

Broader impact, but expensive

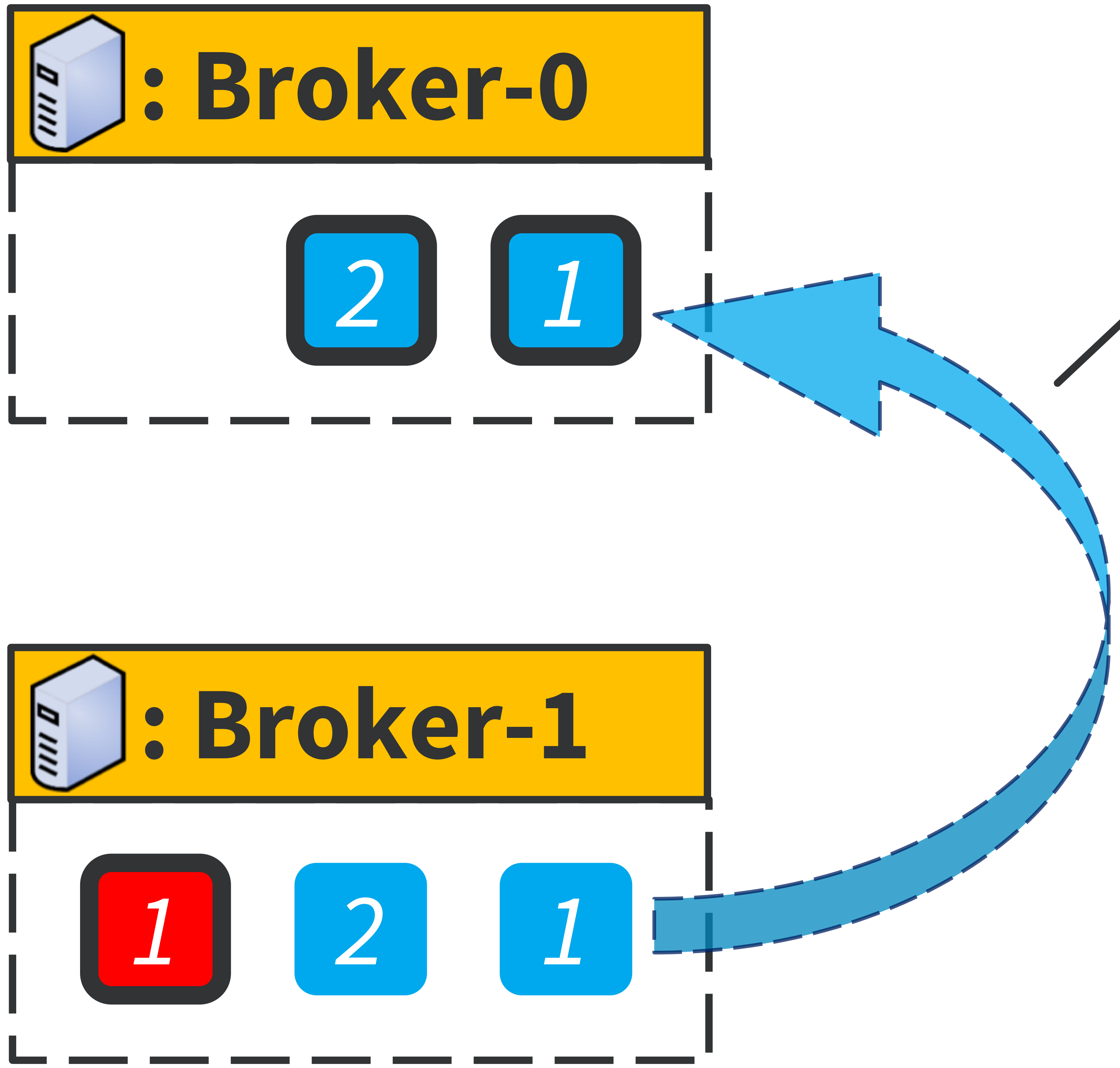
- Requires data transfer*

* Replica swap: Bidirectional reassignments of distinct partition replicas among brokers

Building Blocks of Management: Moving Leadership



Building Blocks of Management: Moving Leadership



Leadership Move

Cheap, but has limited impact

- Affects network bytes out and CPU

A Multi-Objective Optimization Problem

Achieve conflicting cluster management goals while minimizing the impact of required operations on user traffic

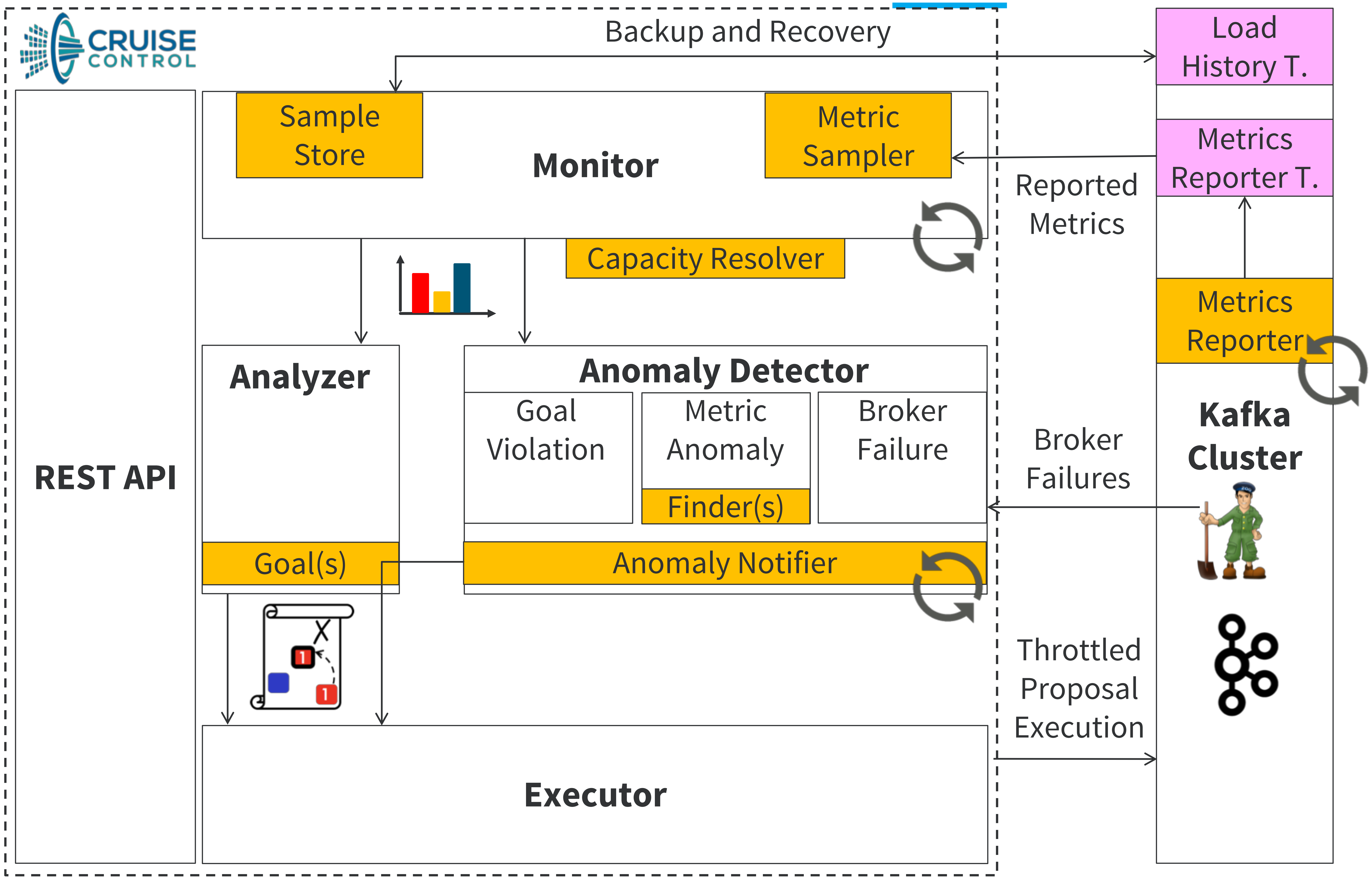
CRUISE CONTROL

ARCHITECTURE

Wrought Iron Vane
Shimming
Brick Diaper
APPROVED GAL. METAL TIES EVERY THIRD COURSE
CENTRE FLUE LINED & REINFORCED SEE PLAN.
ON 14 # ASBESTOS.
G.I. Gutter
Brick Corbels
Barrett Co. Asphalt Lanark Blend Shingles
Flashing
Ceiling
G.I. Gutter
G.I. Gutter
Brick Quoins ½ proj.
G.I. Gutter
Coal Door
P.G.I. Hopes Hopkin Window 3'6" × 5'6"
P.G.I. D.O.
Plate Glass
See Bay Window Details Sh#4
MEN
Shutter
LADIES
Steel Sash
Brick in Cement
G.I. Gutter
TOWER CORNICE 1½" Scale
BEAMFILL TO UNDERSIDE OF ROOF BOARDS.
STEEL LINTELS SEE PLAN.
3/4"
3/4"
Floor
9'-0"
10'-9"
4'-0"
8"
FRONT ELEVATION ¼" Scale
Approved as Approved Nov 17 1936
Steel Sash 1'9" × 5'-6" Approx.
Plate Glass
G.I. Gutter
Brick Corbels
Vent
Vent
Vent
STEEL LINTELS SEE PLAN
Wood Lintel. FACING
STEEL LINTELS SEE PLAN
Maze Glass
Steel Sash
Steel Sash
Shutters
Brick Sill
RIGHT SIDE ELEVATION ¼" Scale
ONTARIO
Revised Aug. 20, 1936
JOB #834
SHEET #2
G.I. = Galvanized Iron
DCB

"Joy Oil Gas Station Blueprints" (Public Domain) https://commons.wikimedia.org/wiki/File:Joy_Oil_gas_station_blueprints.jpg

Cruise Control Architecture



Pluggable Component

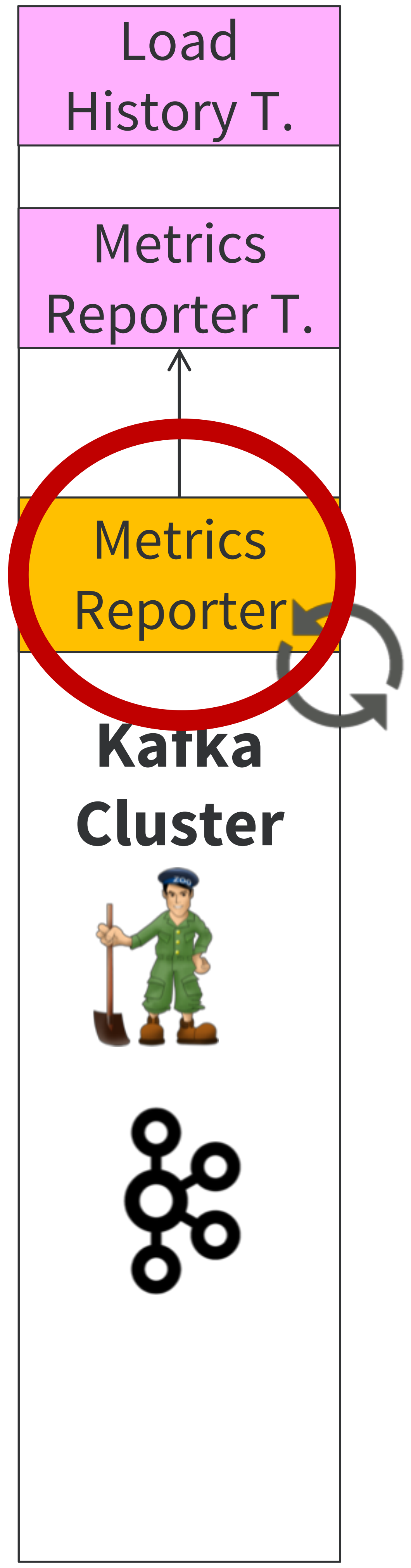
- Implements a public interface
- Accepts custom user code

Internal Topic

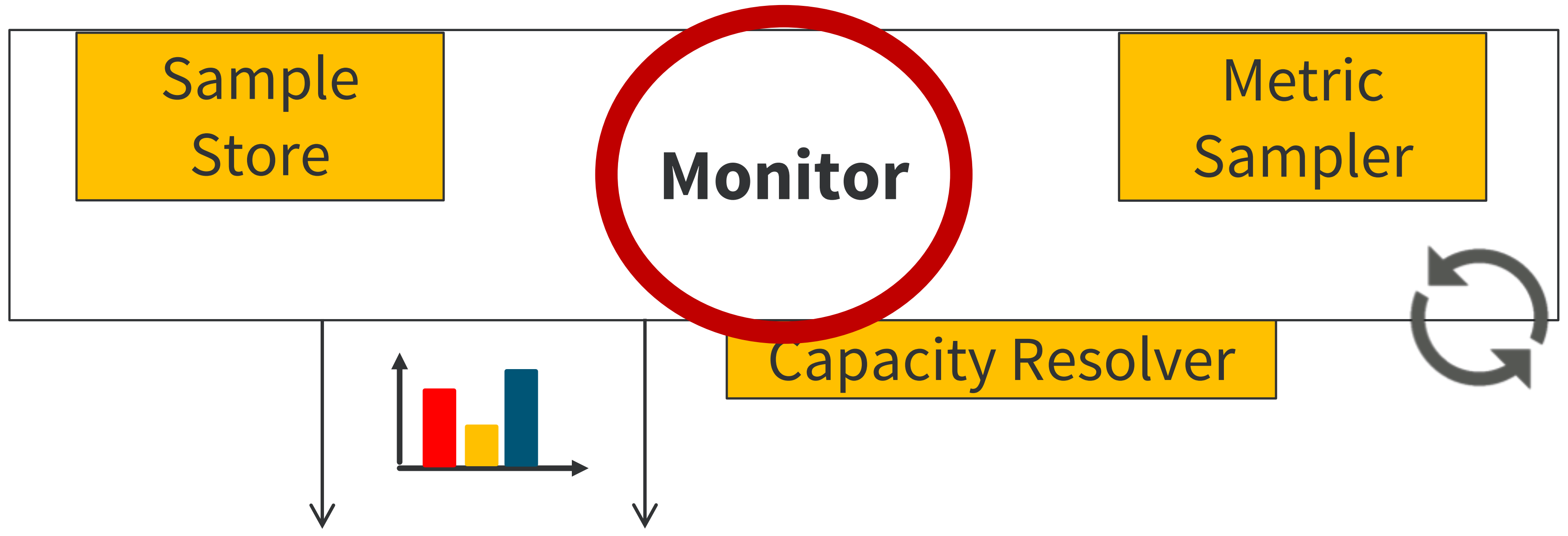
- Created and used by Cruise Control and its metrics reporter

Metrics Reporter

Produces selected Kafka cluster metrics to the configured metrics reporter topic with the configured frequency



Monitor



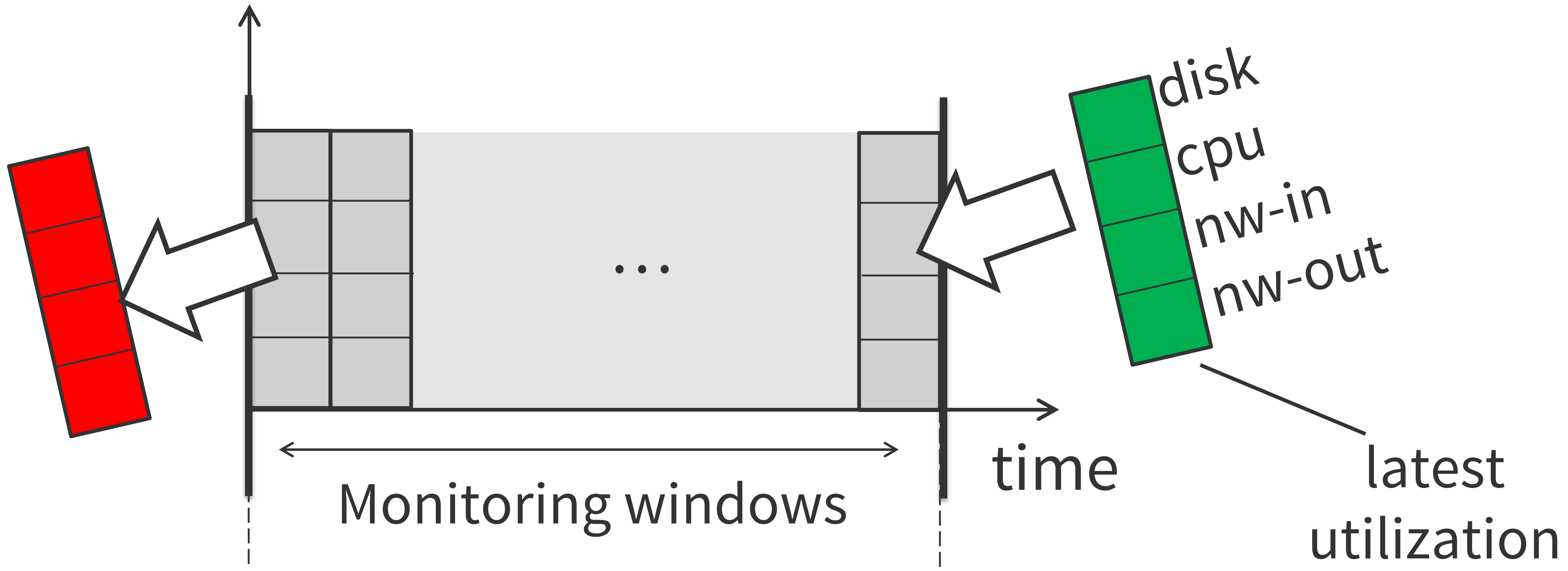
Generates a model (📊) to describe the cluster

Monitor: Cluster Model (📊)

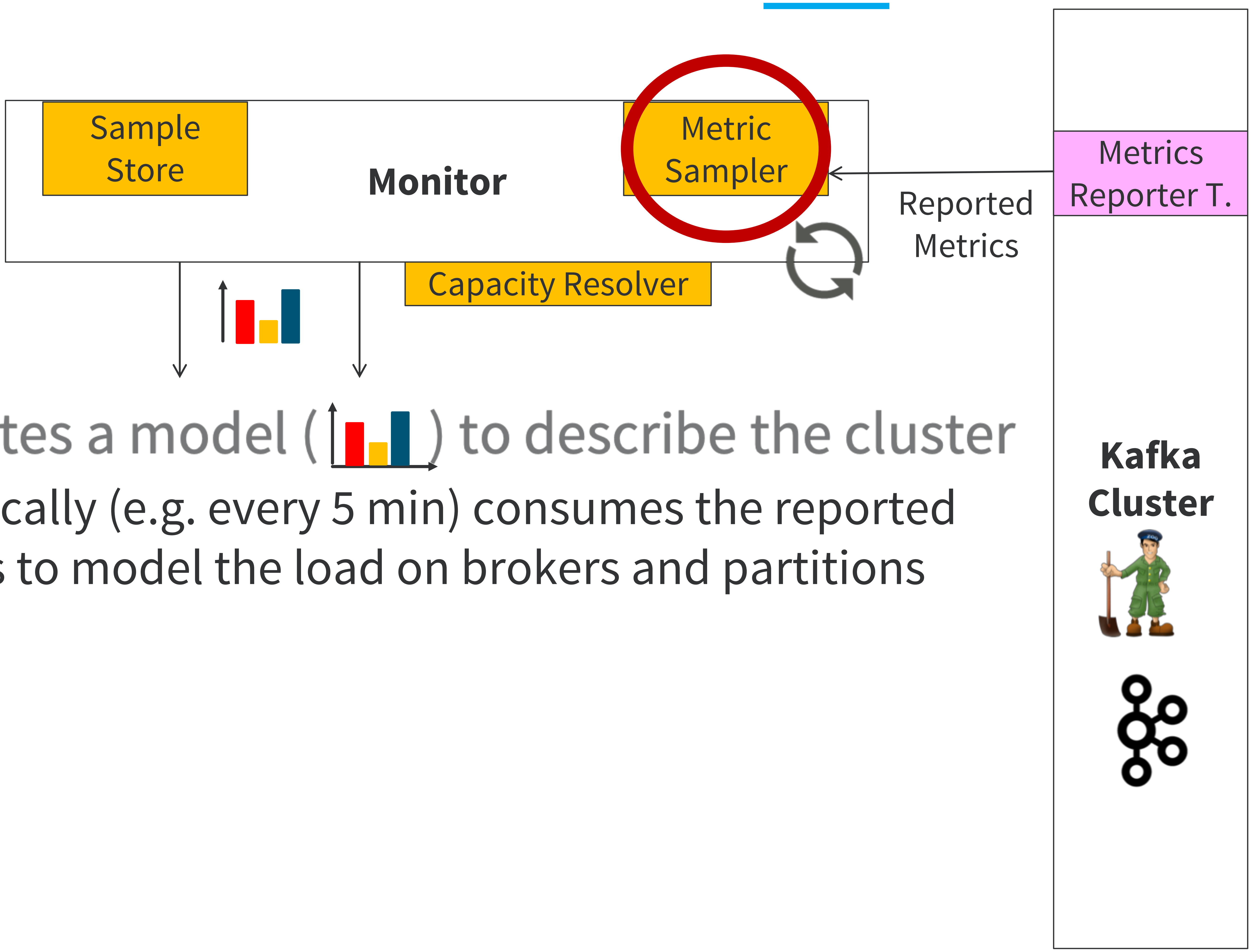
: *Topology* – rack, host, and broker distribution

: *Placement* – replica, leadership, and partition distribution

: *Load* – current and historical utilization of brokers and replicas



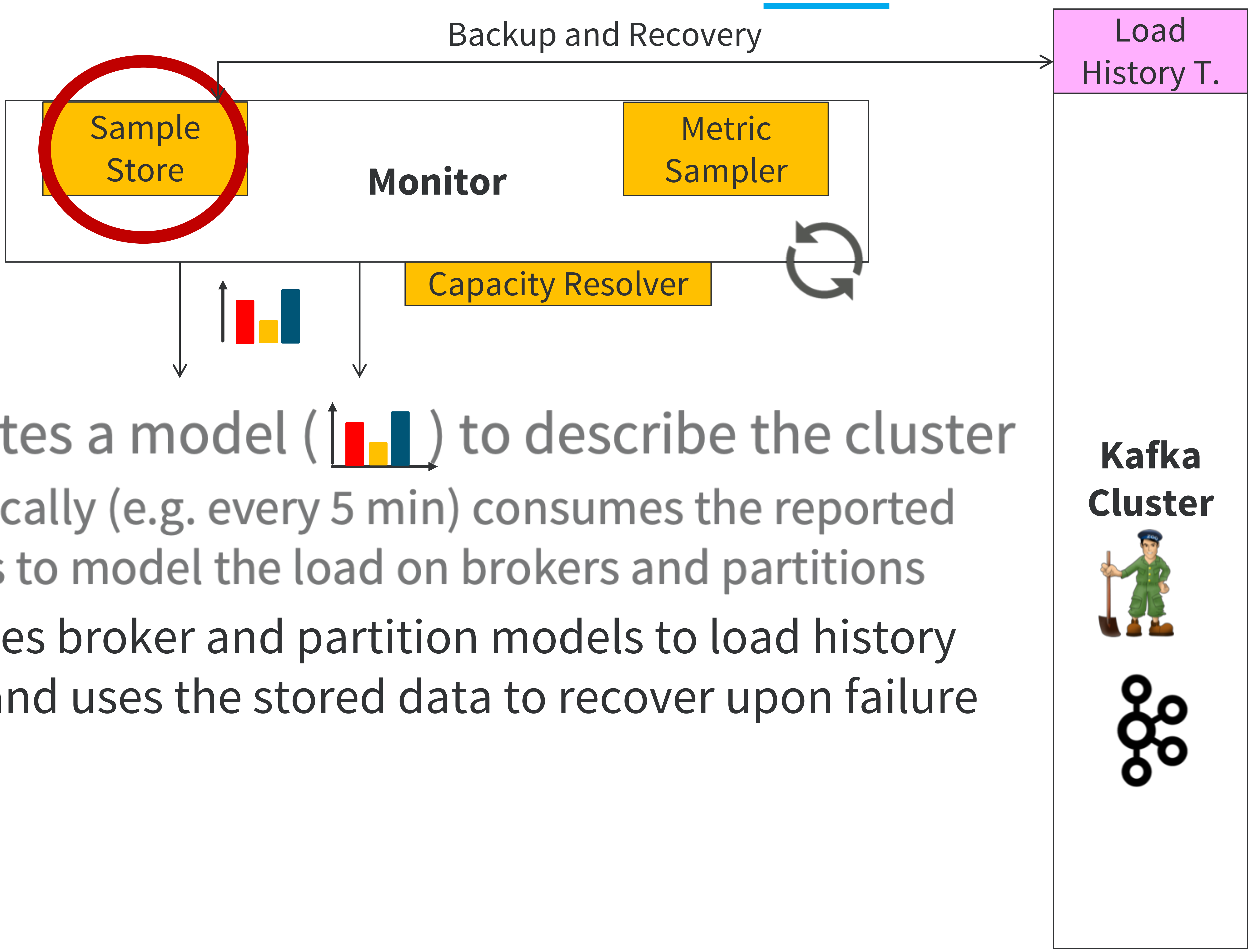
Monitor: Metric Sampler



Generates a model ( ) to describe the cluster

- Periodically (e.g. every 5 min) consumes the reported metrics to model the load on brokers and partitions

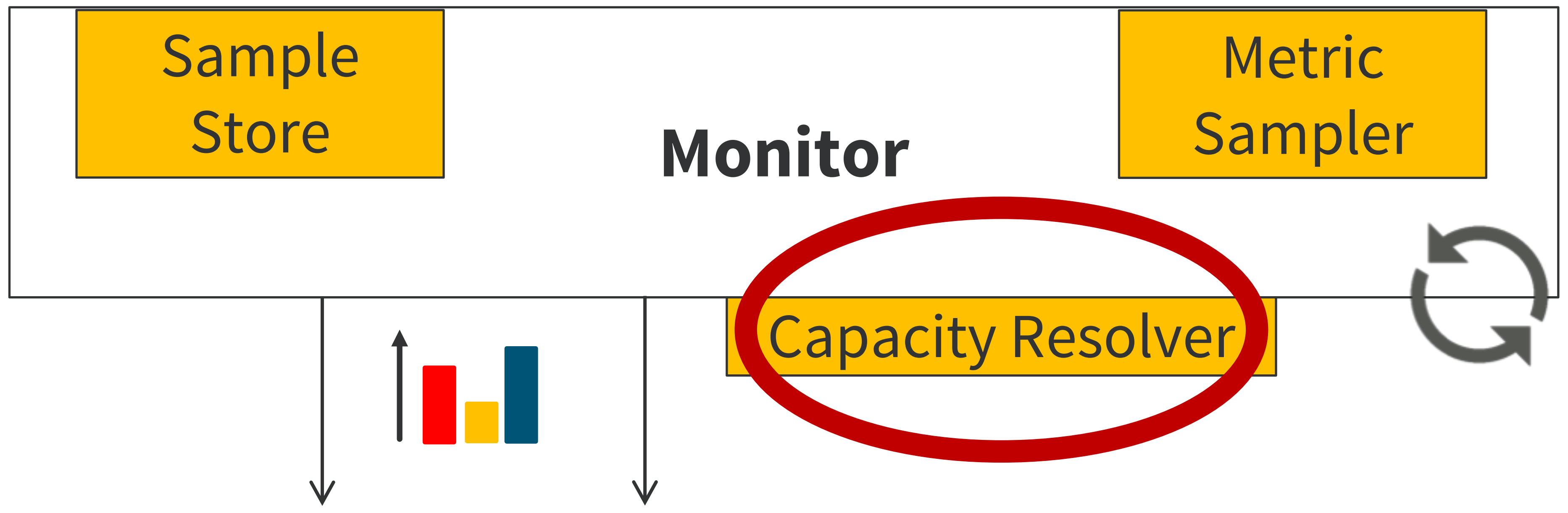
Monitor: Sample Store



Generates a model (📊) to describe the cluster

- Periodically (e.g. every 5 min) consumes the reported metrics to model the load on brokers and partitions
- Produces broker and partition models to load history topic, and uses the stored data to recover upon failure

Monitor: Capacity Resolver



Generates a model ( ) to describe the cluster

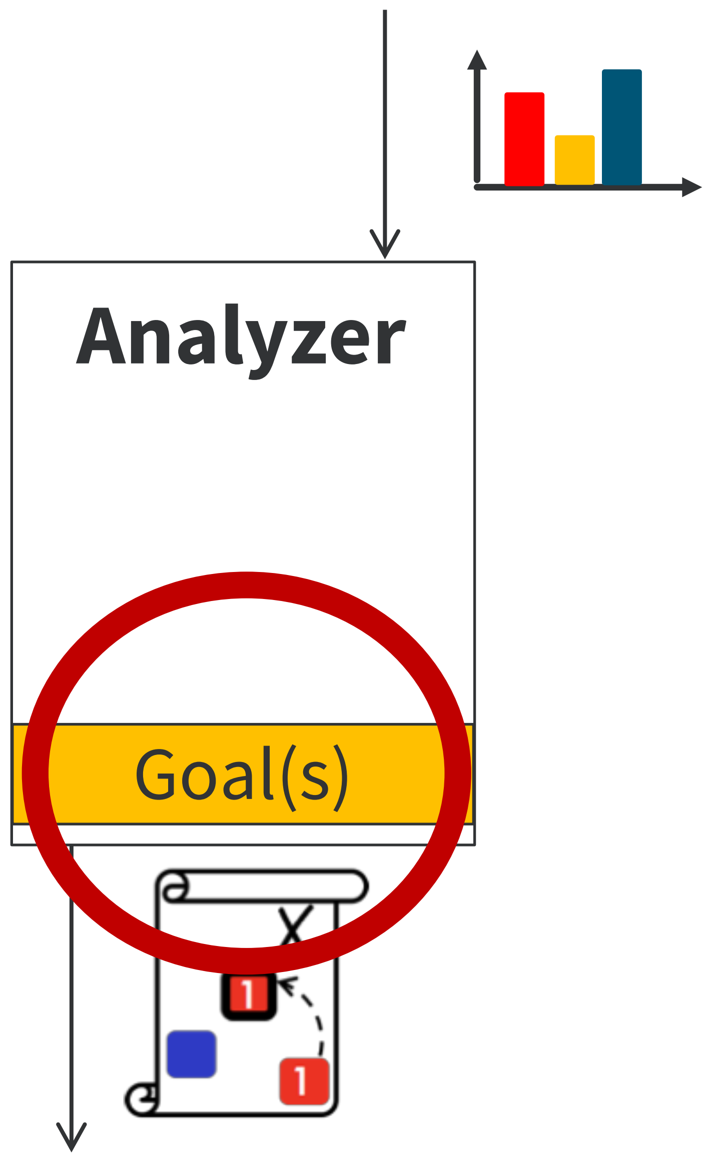
- Periodically (e.g. every 5 min) consumes the reported metrics to model the load on brokers and partitions
- Produces broker and partition models to load history topic, and uses the stored data to recover upon failure
- Gathers the broker capacities from a pluggable resolver

Analyzer

Analyzer
Goal(s)
1
1

Generates proposals to achieve goals via a fast and near-optimal heuristic solution

Analyzer: Goals



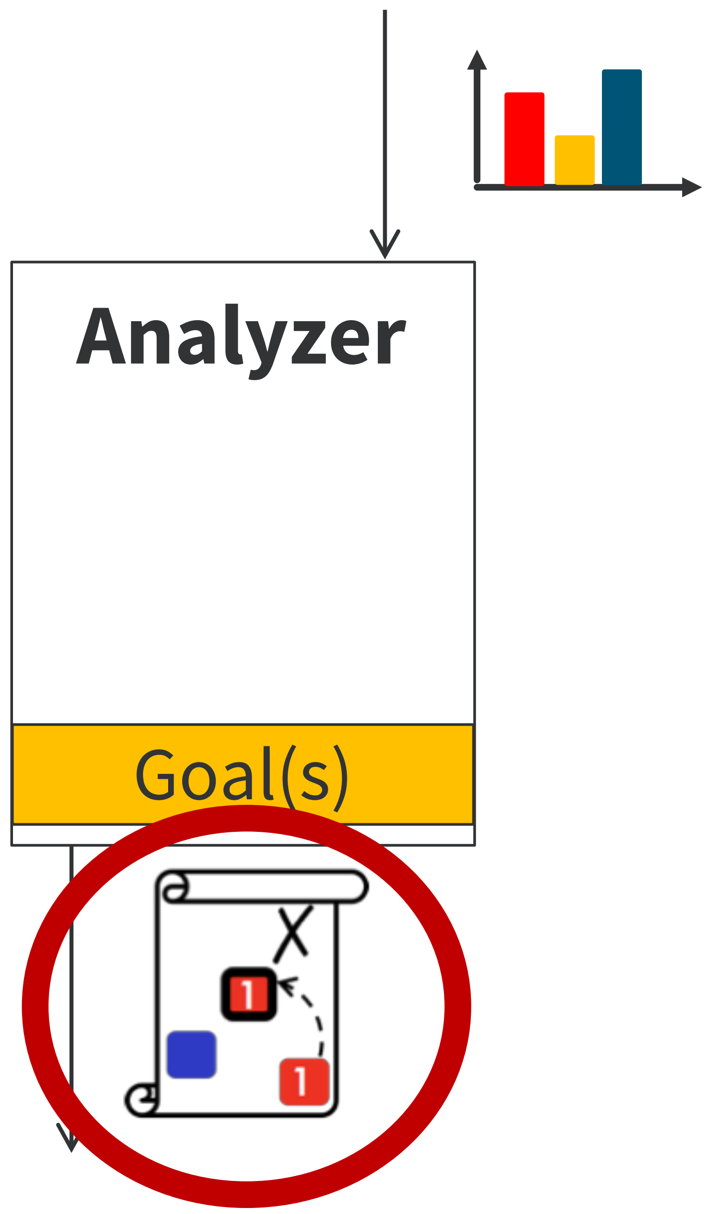
Generates proposals to achieve goals via a fast and near-optimal heuristic solution

: *Priorities* – custom order of optimization

: *Strictness* – hard (e.g. rack awareness) or soft (e.g. resource utilization balance) optimization demands

: *Modes* – e.g. kafka-assigner (https://github.com/linkedin/kafka-tools)

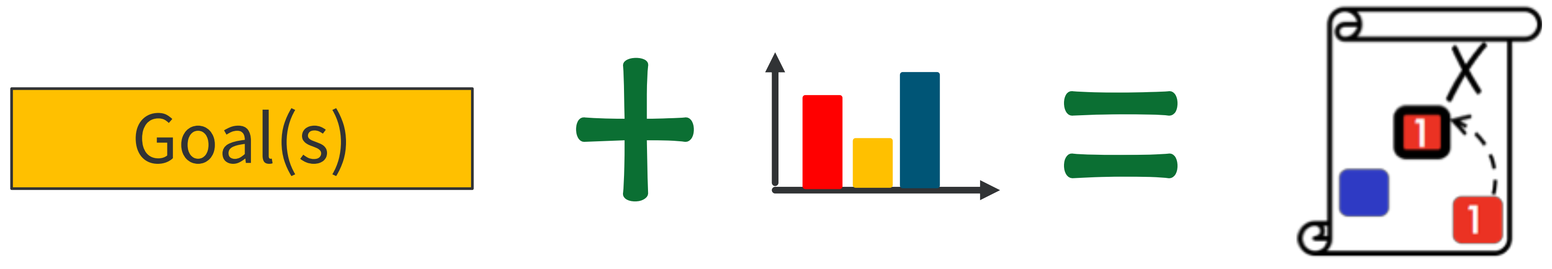
Analyzer: Proposals



Analyzer

Goal(s)

Generates proposals to achieve goals via a fast and near-optimal heuristic solution



Goal(s) + =

Proposals – in order of priority:

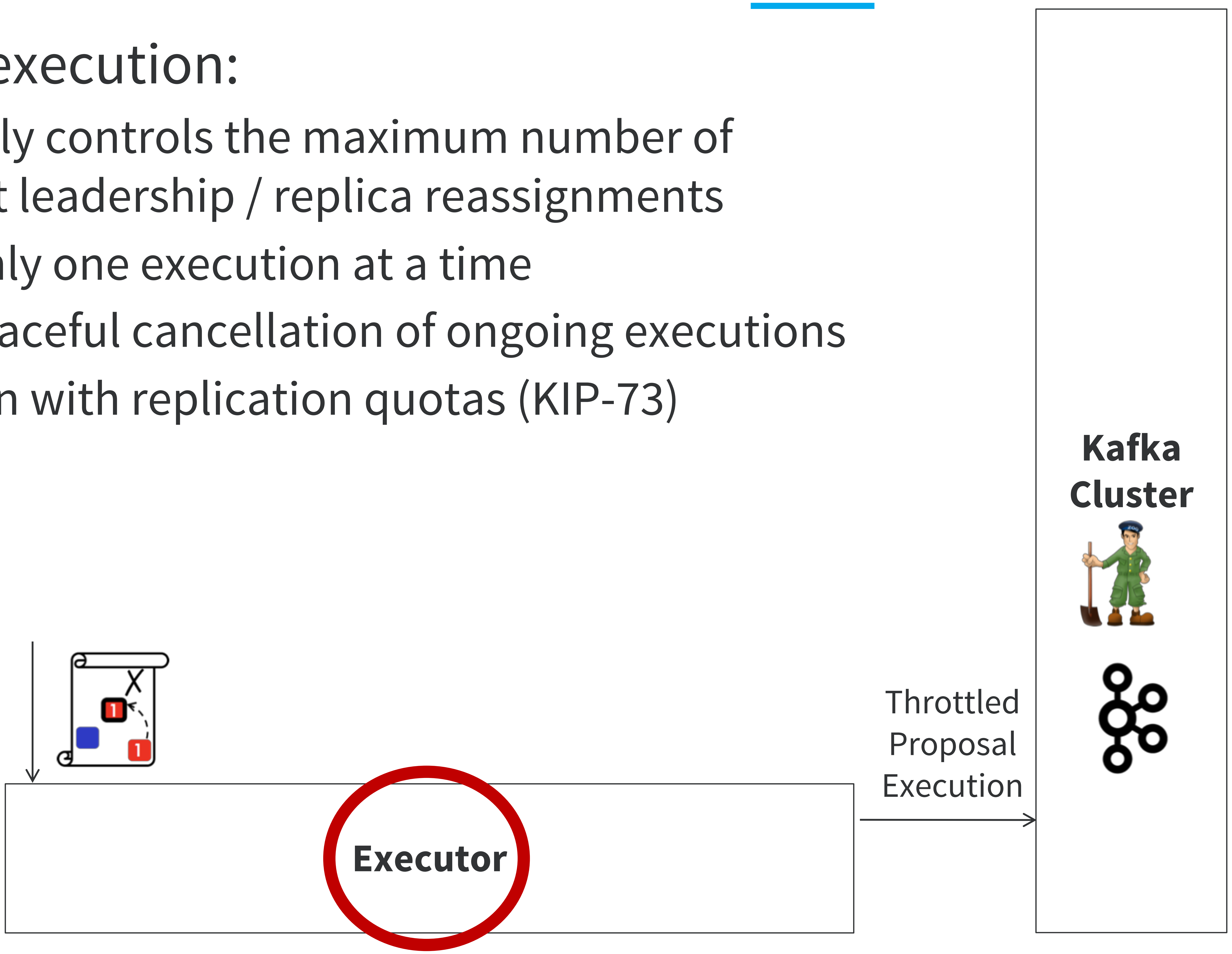
- Leadership move > Replica move > Replica swap

Executor

Proposal execution:

- Dynamically controls the maximum number of concurrent leadership / replica reassignments
- Ensures only one execution at a time
- Enables graceful cancellation of ongoing executions

⌛ Integration with replication quotas (KIP-73)



Kafka Cluster

Throttled Proposal Execution

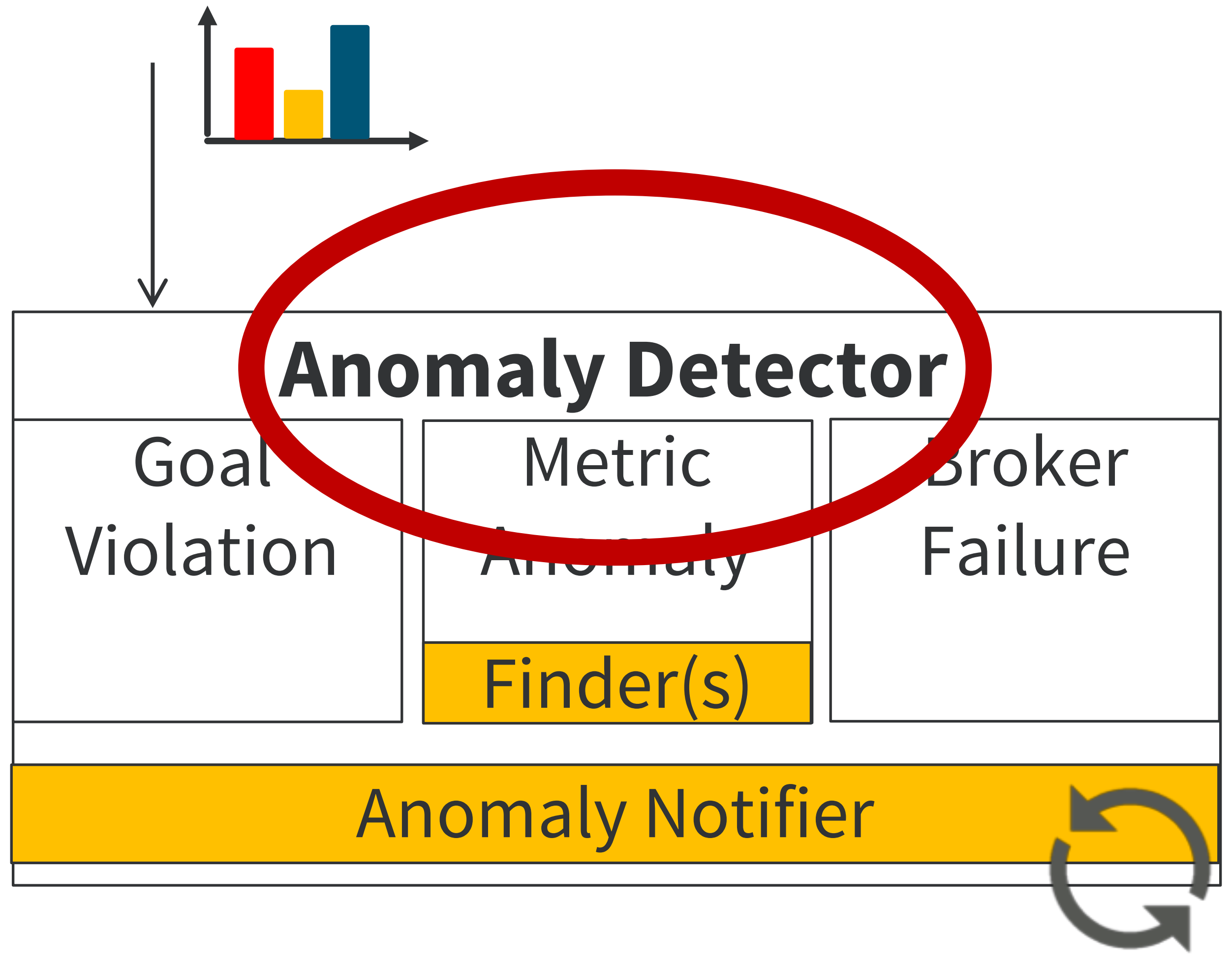
Executor

Anomaly Detector

Identifies, notifies, and fixes (self-healing):

- Violation of anomaly detection goals
- Broker failures
- Metric anomalies

⌛ Disk failures (JBOD)



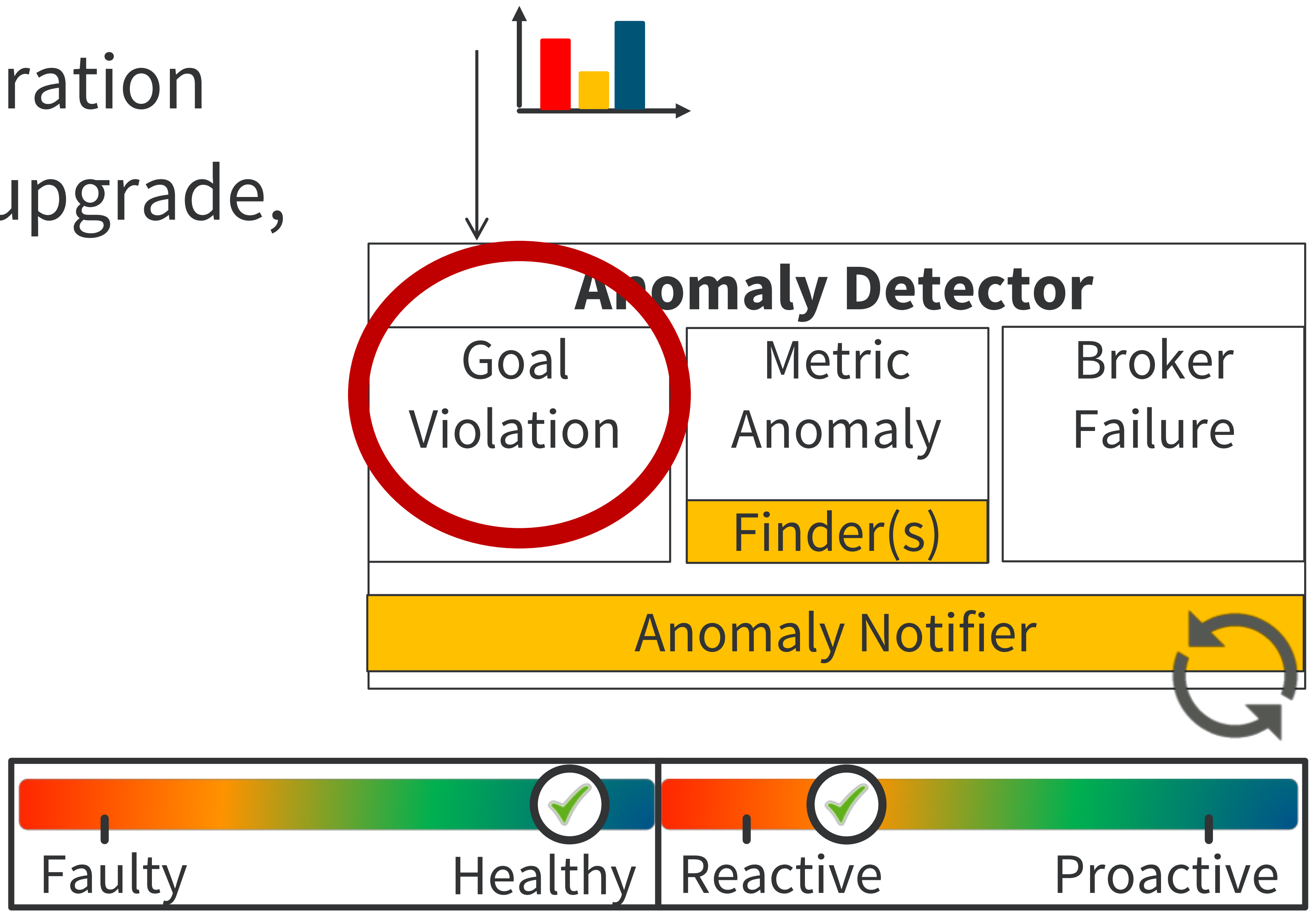
👎👍 : Faulty vs Healthy Cluster

⚡🛡 : Reactive vs. Proactive Mitigation

Anomaly Detector: Goal Violations and Self-Healing

Checks for the violation of the anomaly detection goals

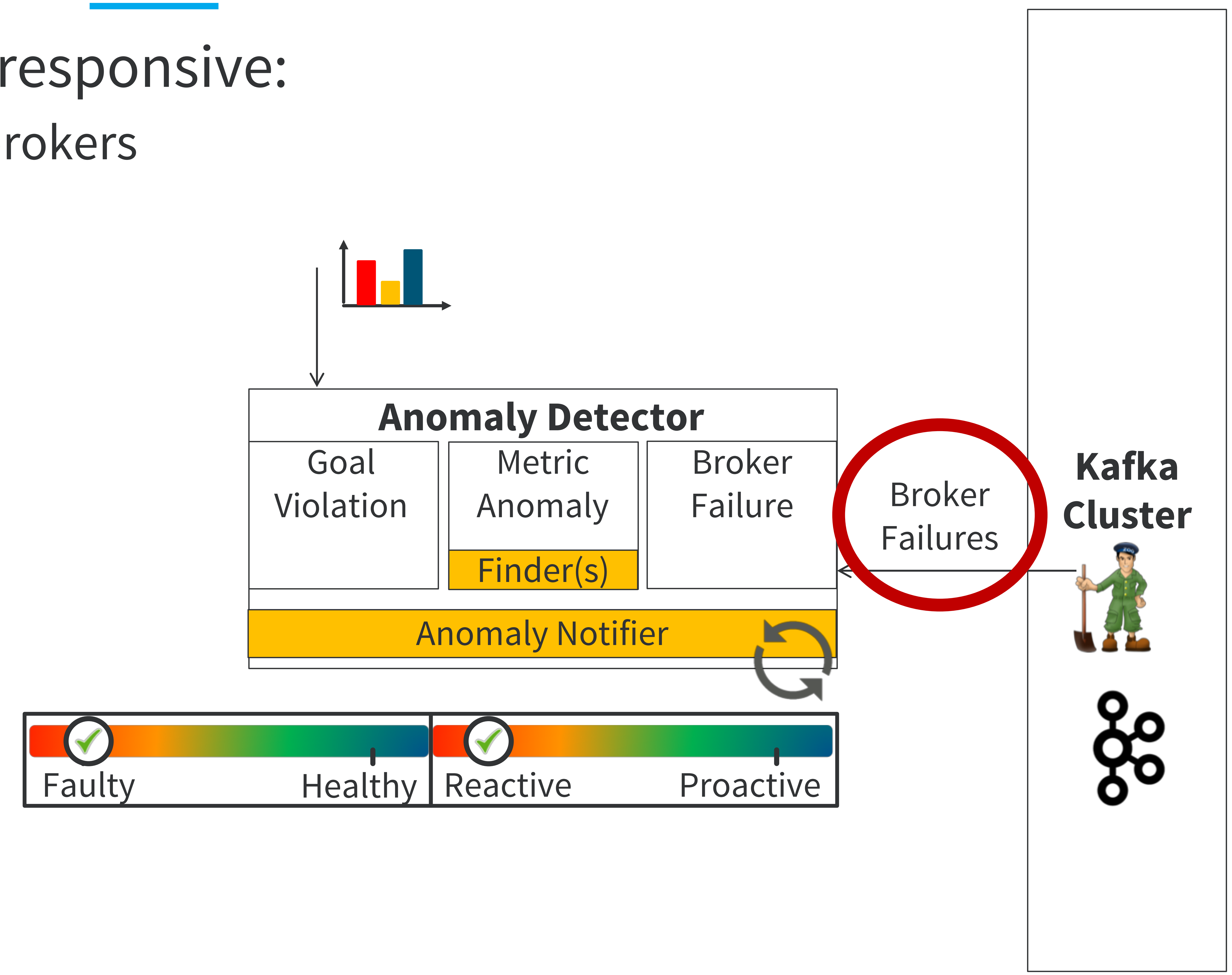
- Identifies *fixable* and *unfixable* goal violations
- Self-healing triggers a cluster rebalance operation
- Avoids false positives due to broker failure, upgrade, restart, or release certification



Anomaly Detector: Broker Failures

Concerned with whether brokers are responsive:

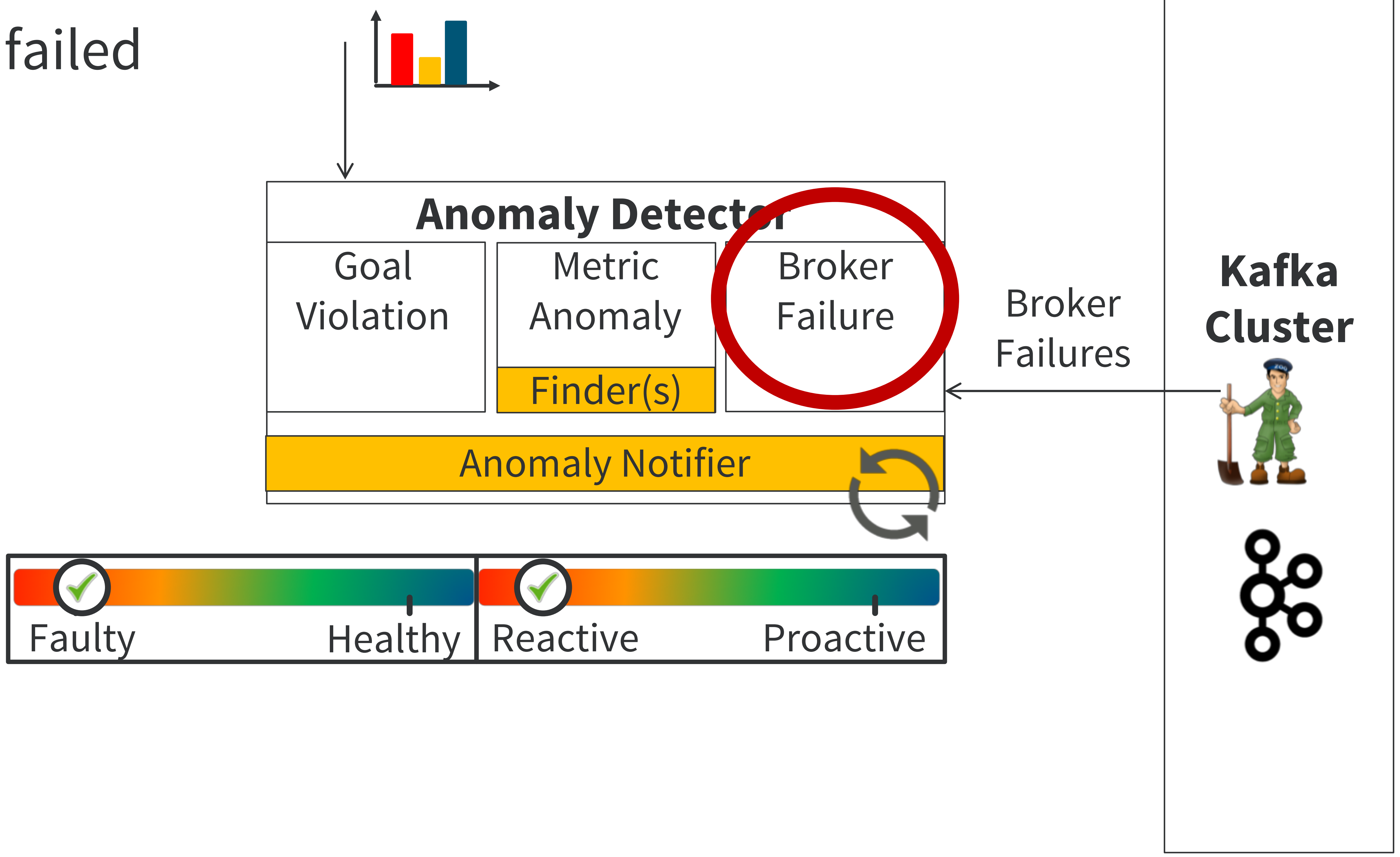
- Ignores the internal state deterioration of brokers
- Identifies fail-stop failures



Anomaly Detector: Broker Failures and Self-Healing

Checks for broker failures:

- Enables a grace period to lower false positives – e.g. due to upgrade, restart, or release certification
- Self-healing triggers a remove operation for failed brokers

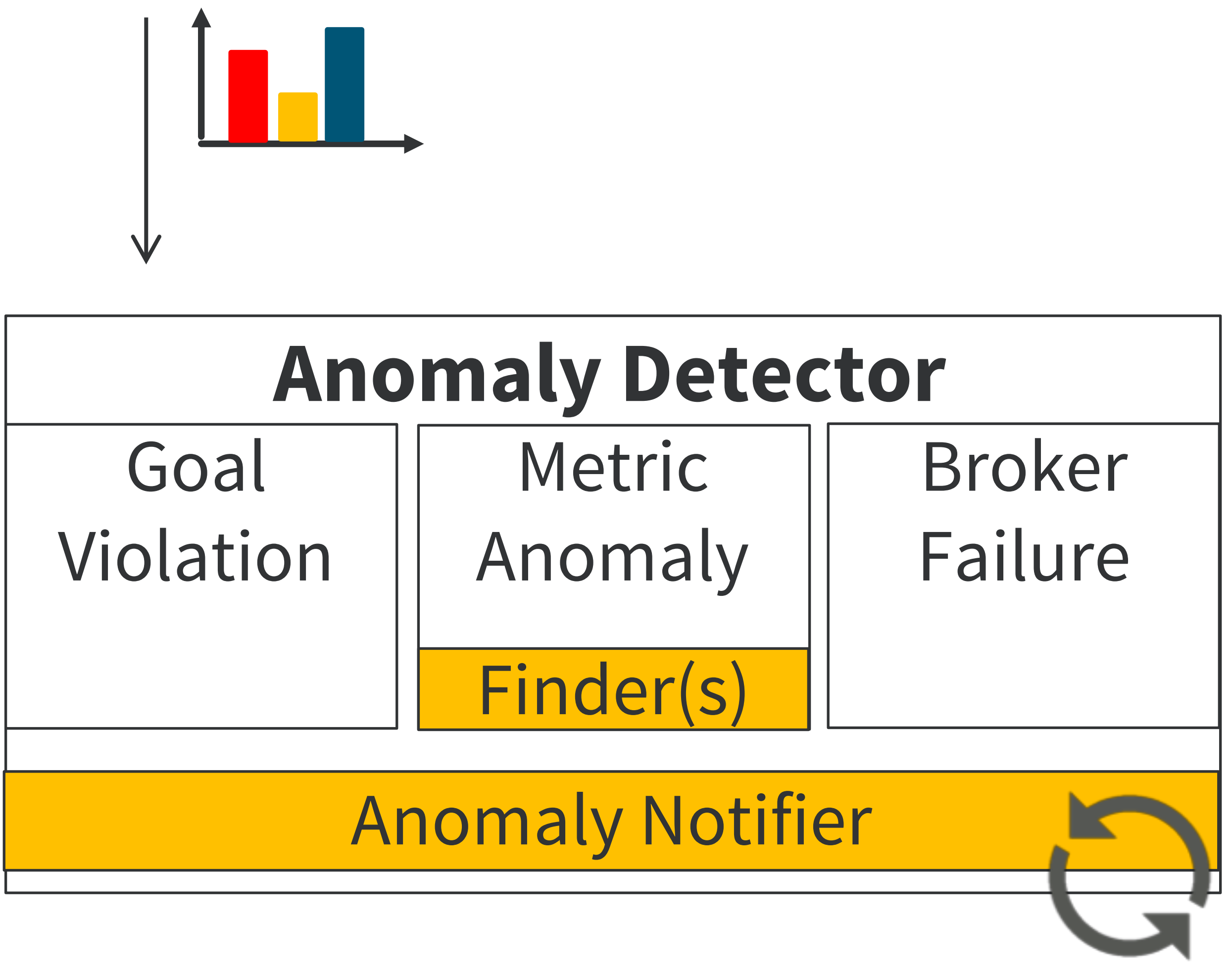


Anomaly Detector: Reactive Mitigation

- Cluster maintenance becomes costly
- Requires immediate attention of affected services
- Poor user experience due to frequent service interruptions

Server & network failures

~
- Size of clusters
- Volume of user traffic
- Hardware degradation


Anomaly Detector

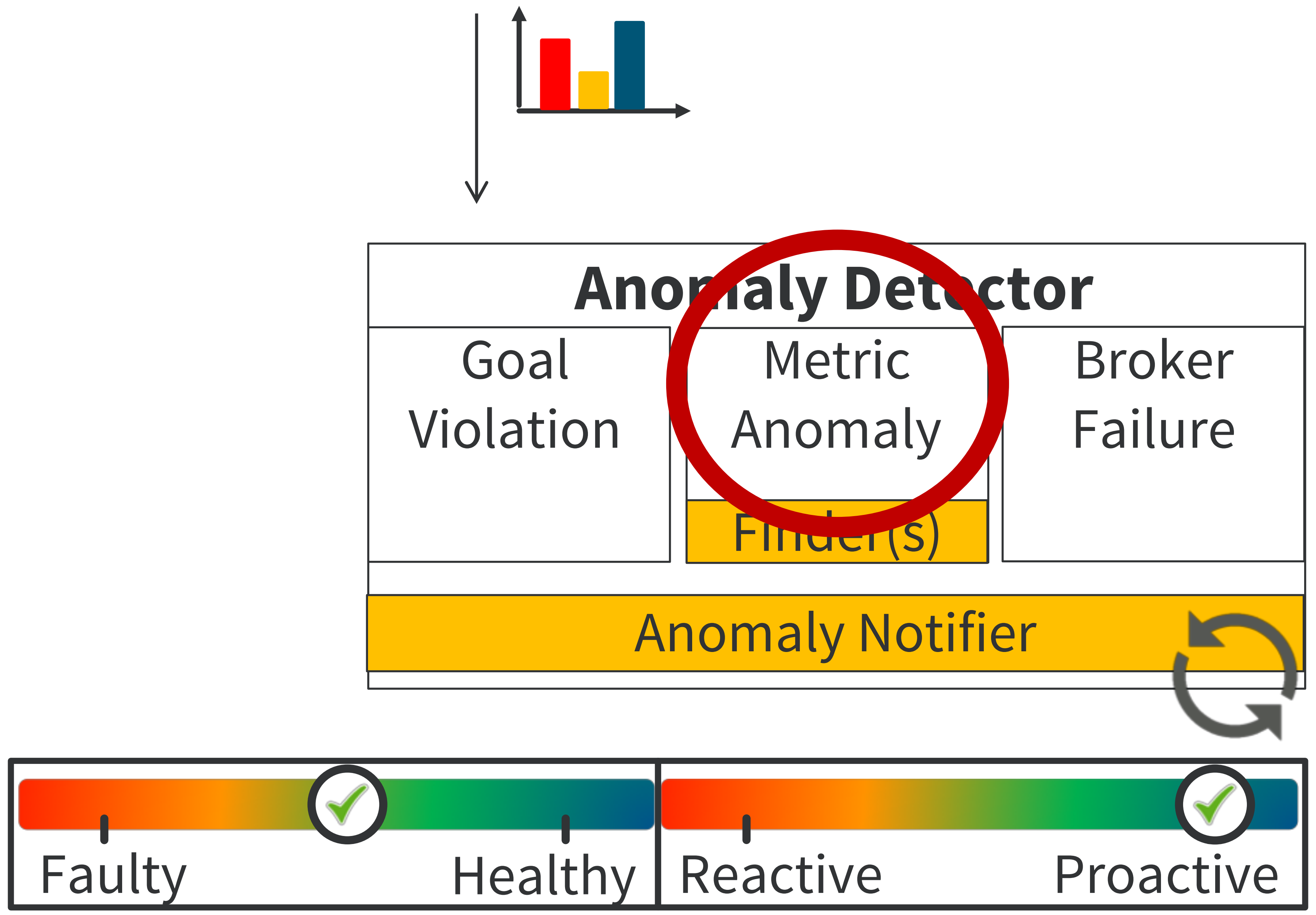
Goal Violation	Metric Anomaly Finder(s)	Broker Failure

Anomaly Notifier

Anomaly Detector: Metric Anomaly

Checks for abnormal changes in broker metrics – e.g. a recent spike in log flush time:

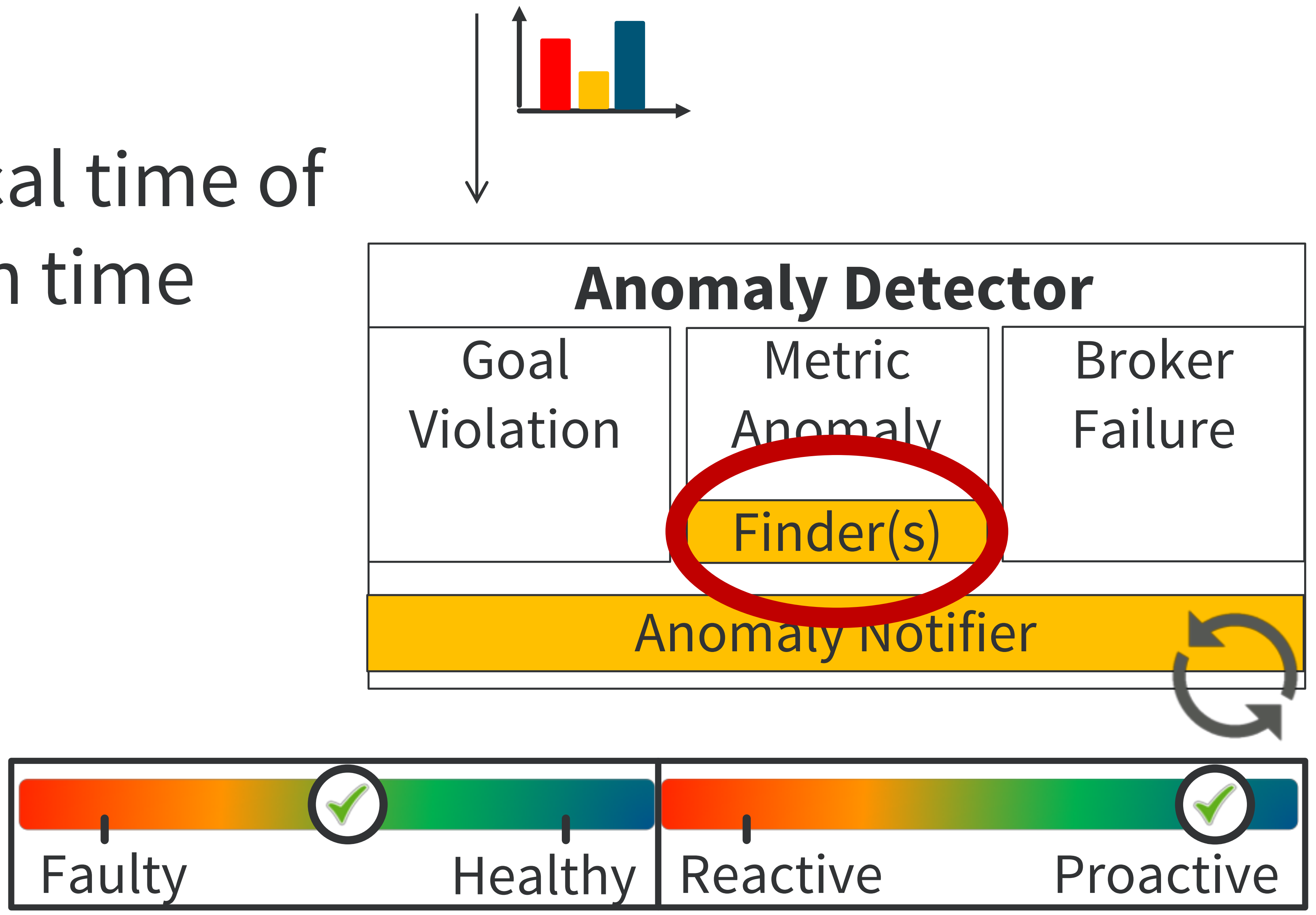
- Self-healing triggers a demote operation for *slow* brokers



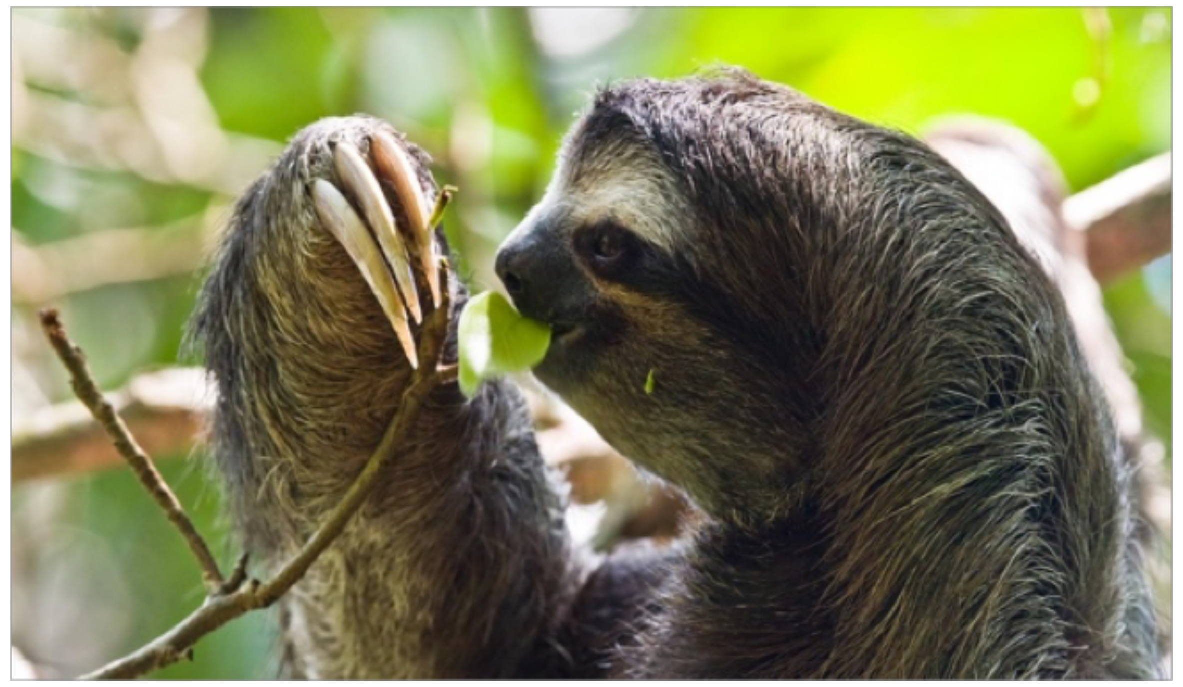
Anomaly Detector: Metric Anomaly

Compares current and historical metrics to detect slow brokers:

- The comparison in the default finder is based on the percentile rank of the latest metric value
- Metrics of interest are configurable – e.g. local time of produce / consume / follower fetch, log flush time
- Supports multiple active finders

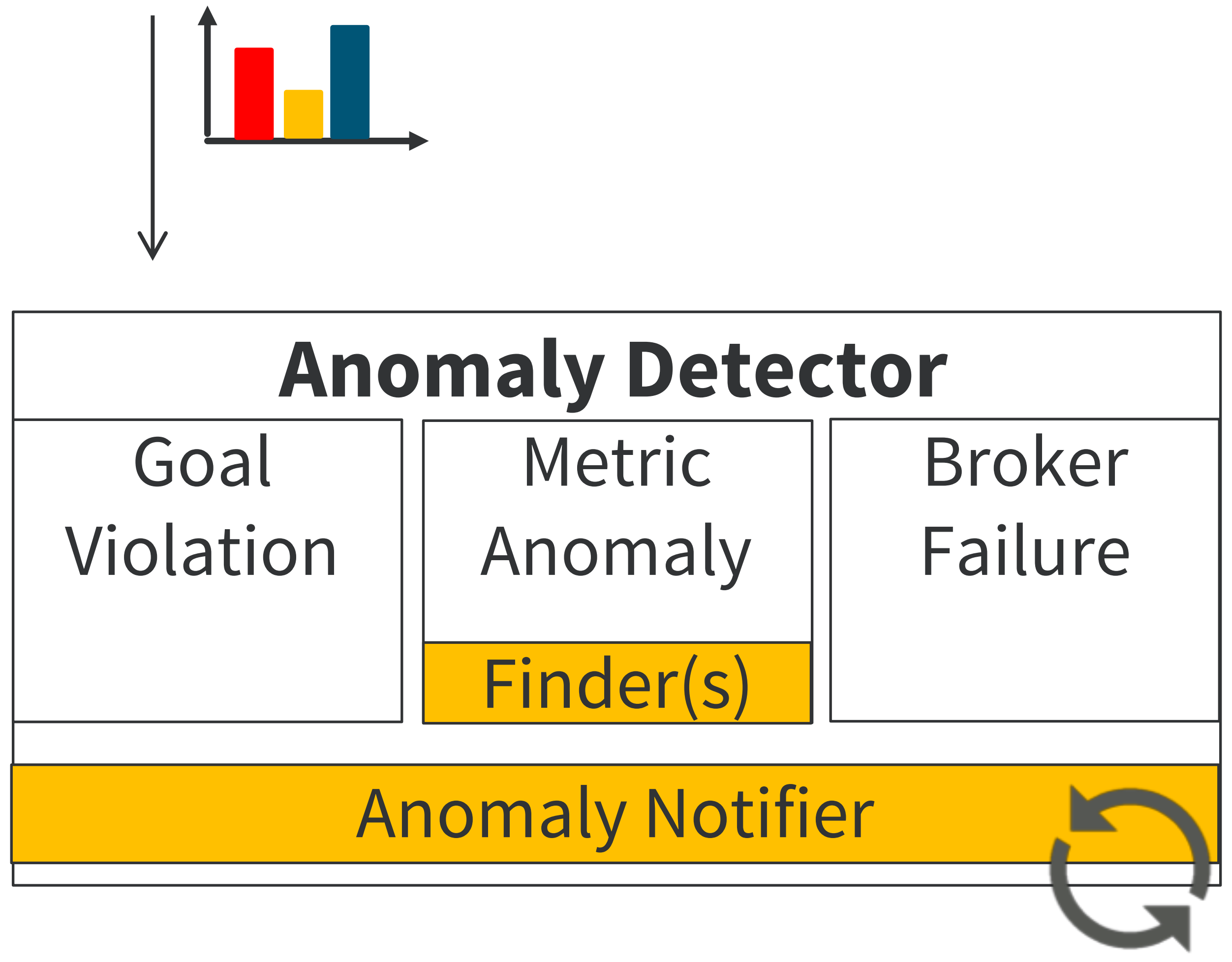


Anomaly Detector: Proactive Mitigation



In-place fix of slow / faulty brokers is non-trivial

- The root cause could be a hardware issue (e.g. a misbehaving disk), a software glitch, or a traffic shift
- Hence, the mitigation strategies are agnostic of the particular issue with the broker



Anomaly Detector

Goal Violation

Metric Anomaly

Finder(s)

Broker Failure

Anomaly Notifier

"Three-toed-sloth" (CC BY 2.5): https://en.wikipedia.org/wiki/File:MC_Drei-Finger-Faultier.jpg

REST API

Supports sync and async endpoints including:

GET
- Cluster Load
- Partition Load
- Proposals
- Kafka Cluster State
- Cruise Control State
- User Tasks

POST
- Add / Remove / Demote Broker
- Rebalance Cluster
- Fix Offline Replicas (JBOD)
- Stop Ongoing Execution
- Pause / Resume Sampling
- Admin – ongoing behavior changes

GUI & multi-cluster management

Managing the Manager – *Monitoring Cruise Control*

Reported *JMX* metrics include:

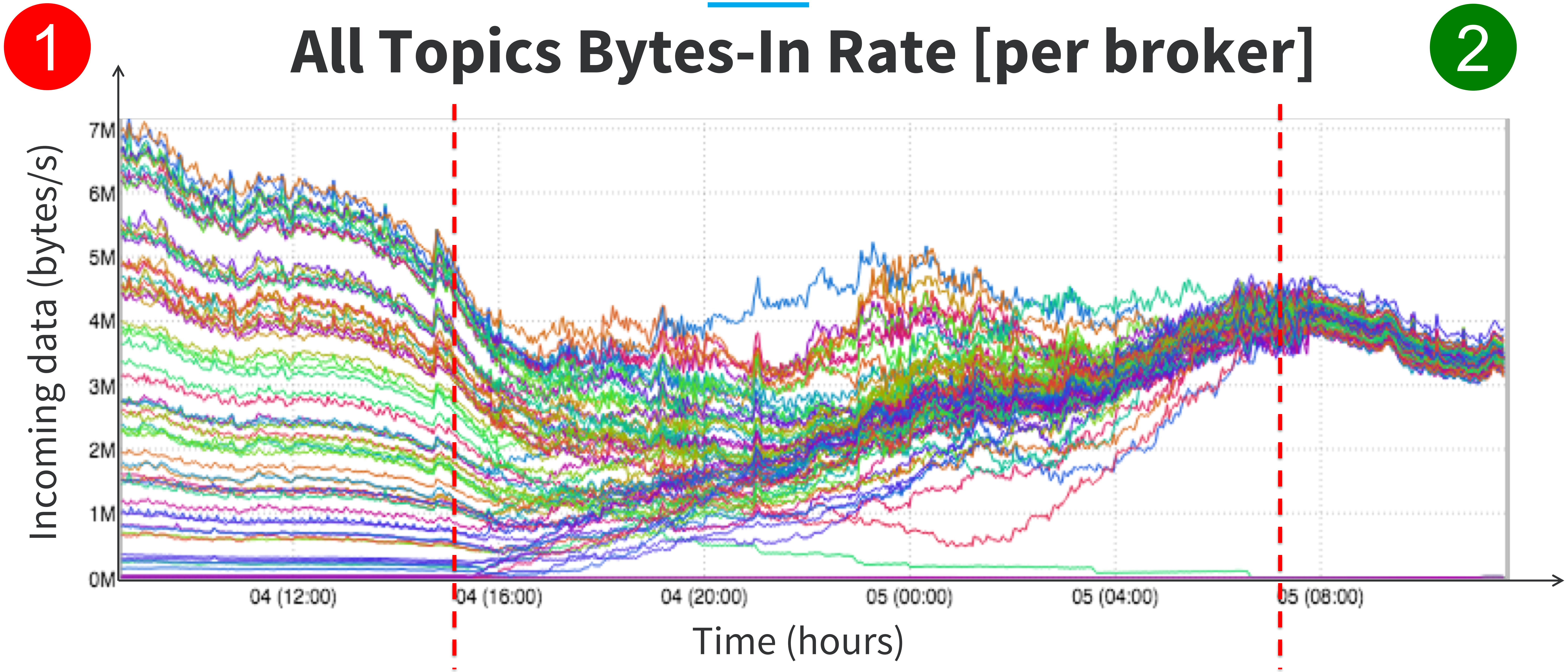
Executor : Started, stopped, and ongoing executions in different modes, and the status of balancing tasks

Anomaly Detector : Broker failure, goal violation, and metric anomaly rate

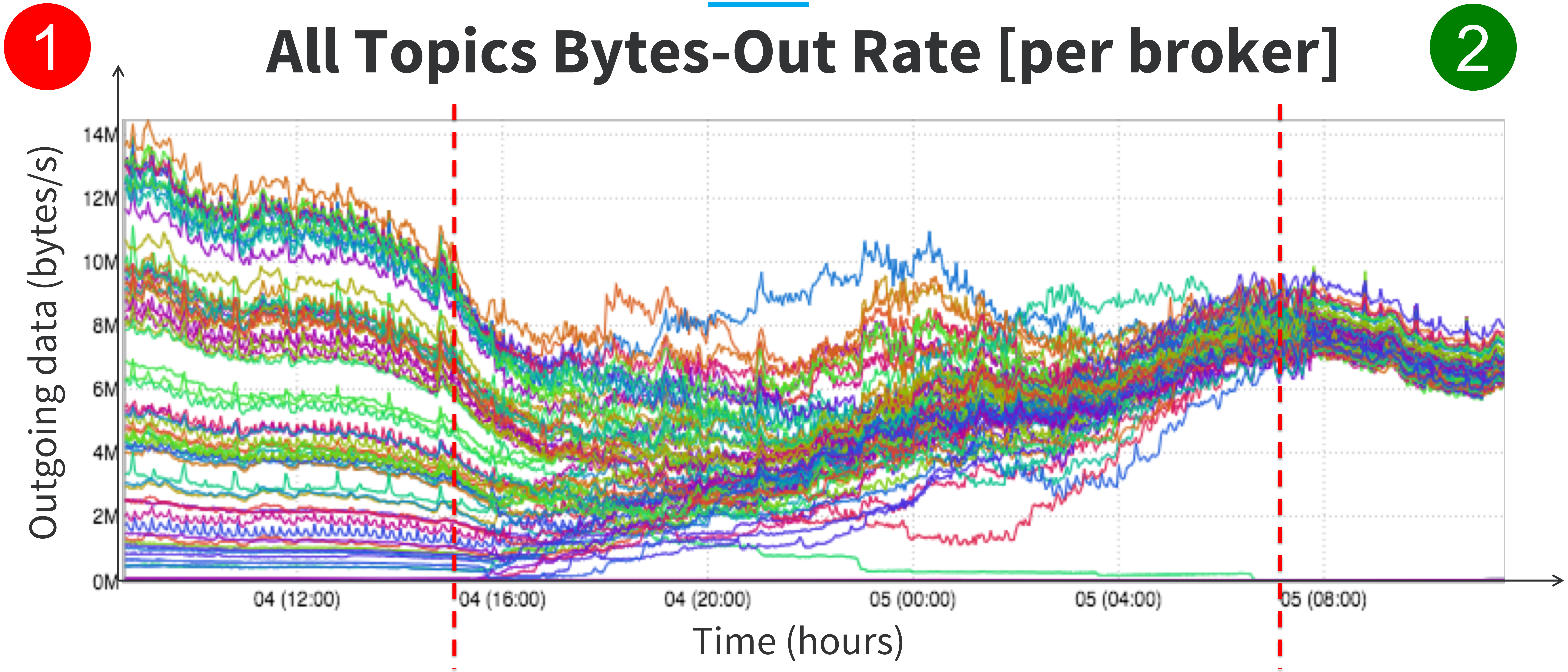
Monitor : Cluster model and sampling performance

Analyzer : Stats on proposal generation

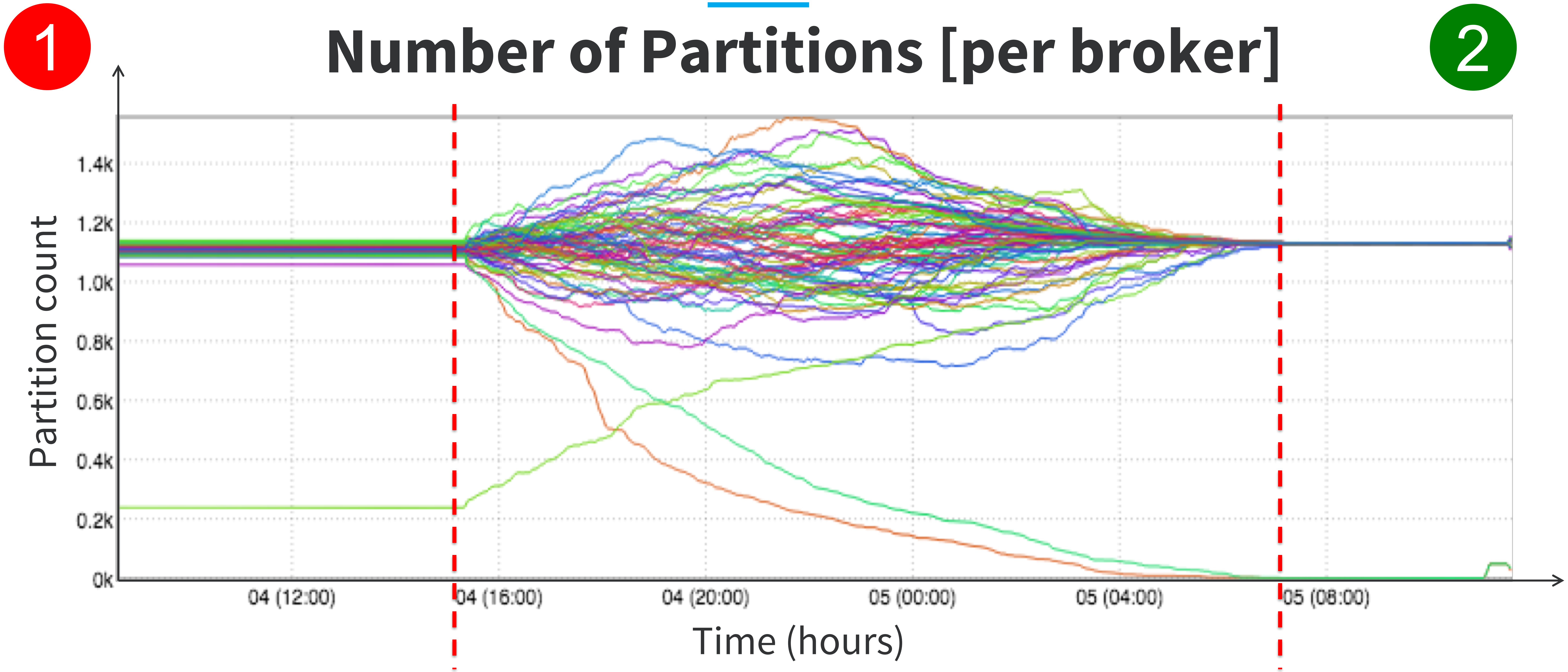
Evaluation: Remove Brokers and Rebalance



Evaluation: Remove Brokers and Rebalance



Evaluation: Remove Brokers and Rebalance

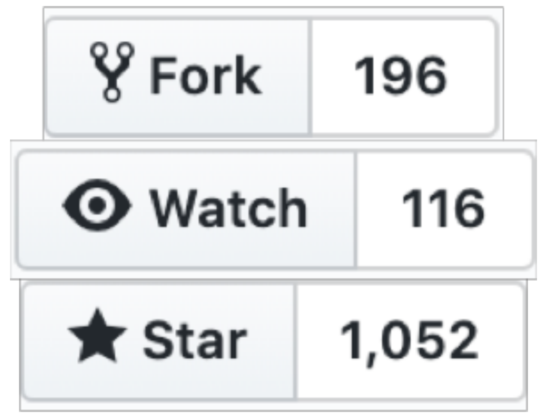


Summary

A system that provides effortless management of Kafka clusters

- ✓ Admin Operations for Cluster Maintenance
- ✓ Anomaly Detection with Self-healing
- ✓ Real-Time Monitoring of Kafka Clusters
- ⌛ Integration with Other Systems – e.g. Apache Helix

More…

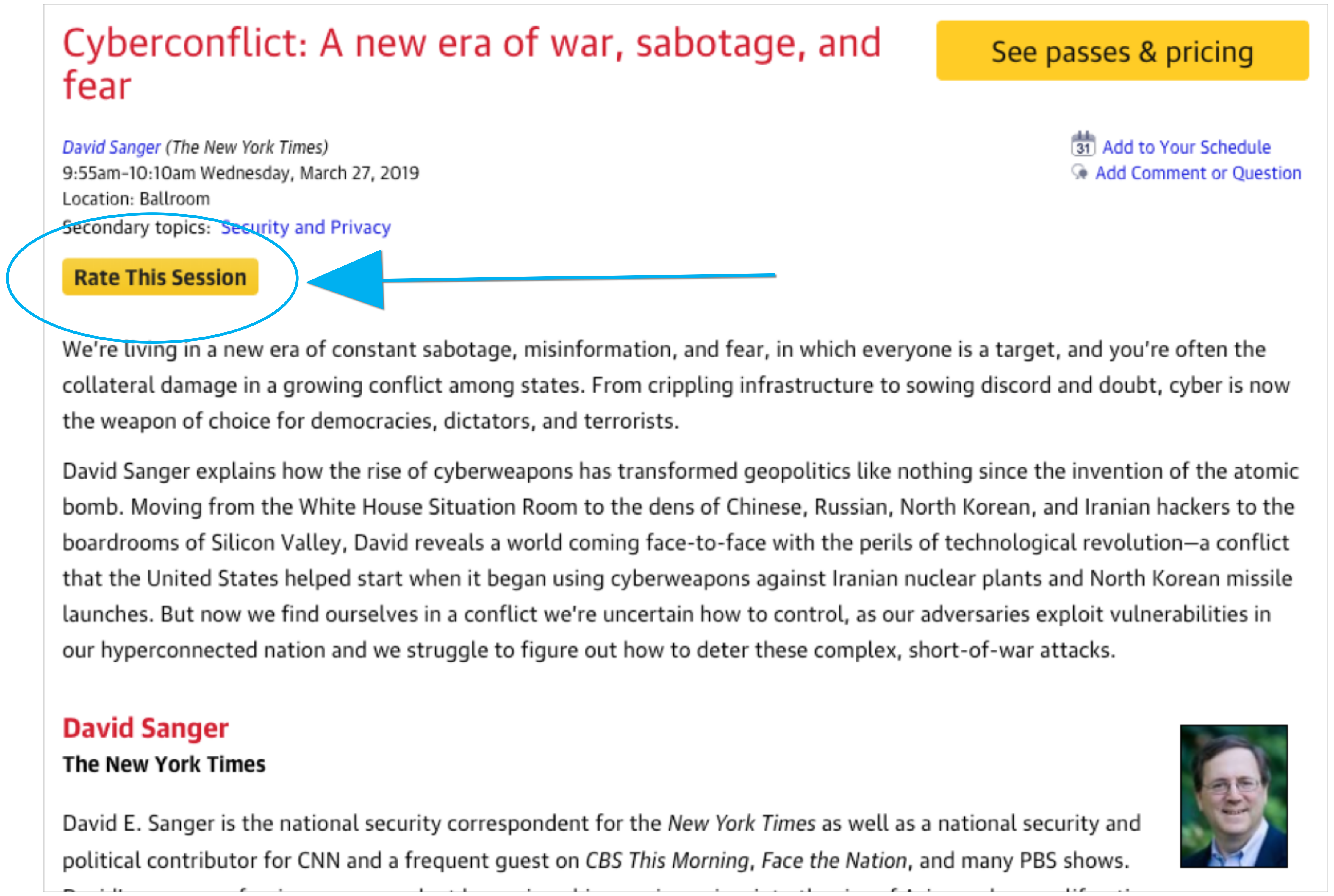


: Open source repository (https://github.com/linkedin/cruise-control)

: Gitter room (https://gitter.im/kafka-cruise-control)

: UI (https://github.com/linkedin/cruise-control-ui)

Rate today ’s session



Cyberconflict: A new era of war, sabotage, and fear

See passes & pricing

David Sanger (The New York Times)
9:55am–10:10am Wednesday, March 27, 2019
Location: Ballroom
Secondary topics: Security and Privacy

Add to Your Schedule
Add Comment or Question

Rate This Session

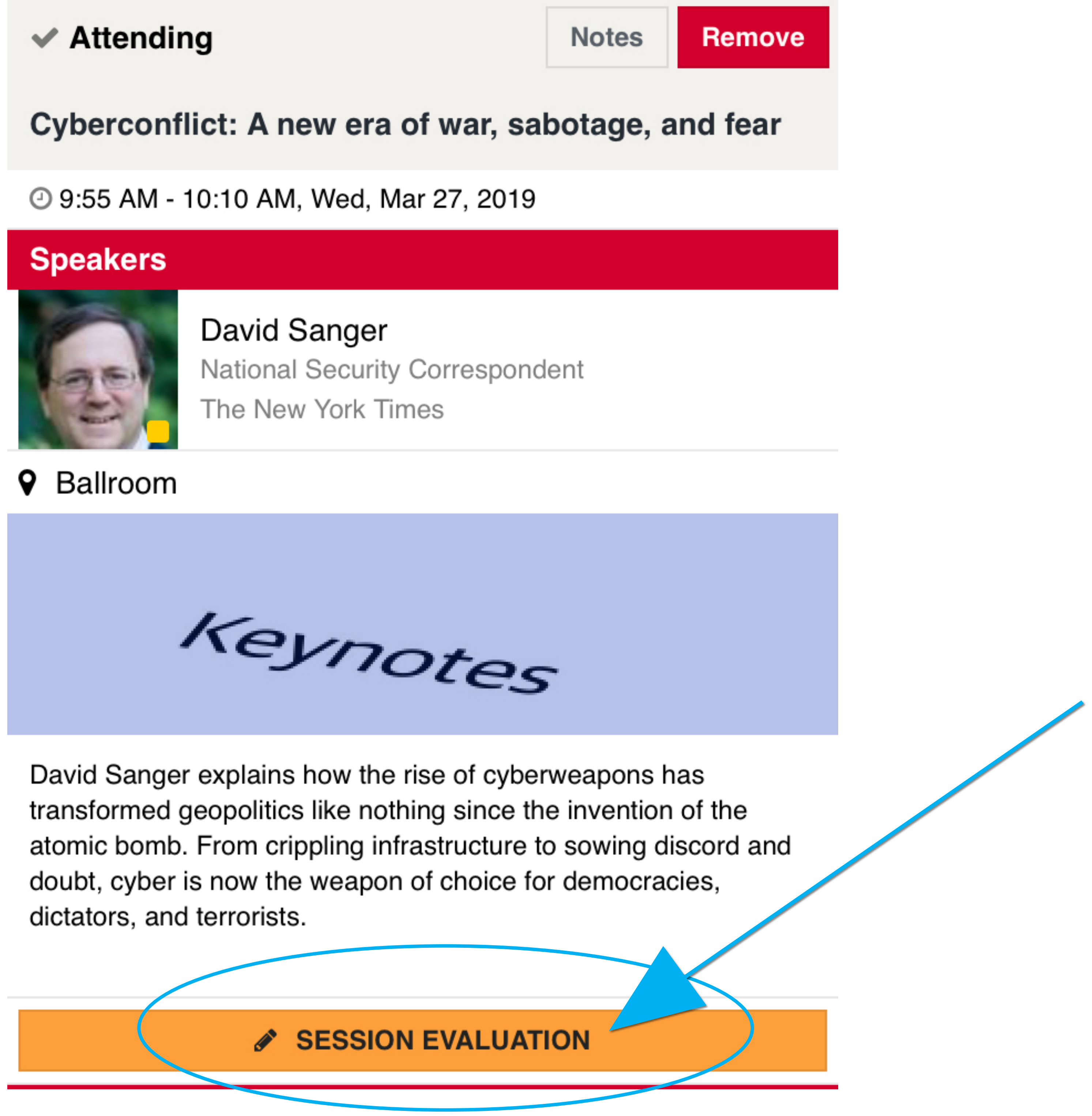
We're living in a new era of constant sabotage, misinformation, and fear, in which everyone is a target, and you're often the collateral damage in a growing conflict among states. From crippling infrastructure to sowing discord and doubt, cyber is now the weapon of choice for democracies, dictators, and terrorists.

David Sanger explains how the rise of cyberweapons has transformed geopolitics like nothing since the invention of the atomic bomb. Moving from the White House Situation Room to the dens of Chinese, Russian, North Korean, and Iranian hackers to the boardrooms of Silicon Valley, David reveals a world coming face-to-face with the perils of technological revolution—a conflict that the United States helped start when it began using cyberweapons against Iranian nuclear plants and North Korean missile launches. But now we find ourselves in a conflict we're uncertain how to control, as our adversaries exploit vulnerabilities in our hyperconnected nation and we struggle to figure out how to deter these complex, short-of-war attacks.

David Sanger
The New York Times

David E. Sanger is the national security correspondent for the *New York Times* as well as a national security and political contributor for CNN and a frequent guest on *CBS This Morning*, *Face the Nation*, and many PBS shows.

Session page on conference website



✓ **Attending** Notes Remove

Cyberconflict: A new era of war, sabotage, and fear

9:55 AM - 10:10 AM, Wed, Mar 27, 2019

Speakers

David Sanger
National Security Correspondent
The New York Times

Ballroom

David Sanger explains how the rise of cyberweapons has transformed geopolitics like nothing since the invention of the atomic bomb. From crippling infrastructure to sowing discord and doubt, cyber is now the weapon of choice for democracies, dictators, and terrorists.

SESSION EVALUATION

O’Reilly Events App