THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
EUROPE

A day in the life of a log message

Kyle Liberti, Josef Karasek

@Pepe_CZ

THE LINUX FOUNDATION

- Order is vital for scale
- Abstractions make systems manageable

Problems of Distributed Systems

- Reliability
- Data throughput
- Latency

Abstracted Tools

- Allow us to leverage complex systems with little work
- When things get too complicated, we add another layer of abstraction
- ....and we repeat this process

OKD

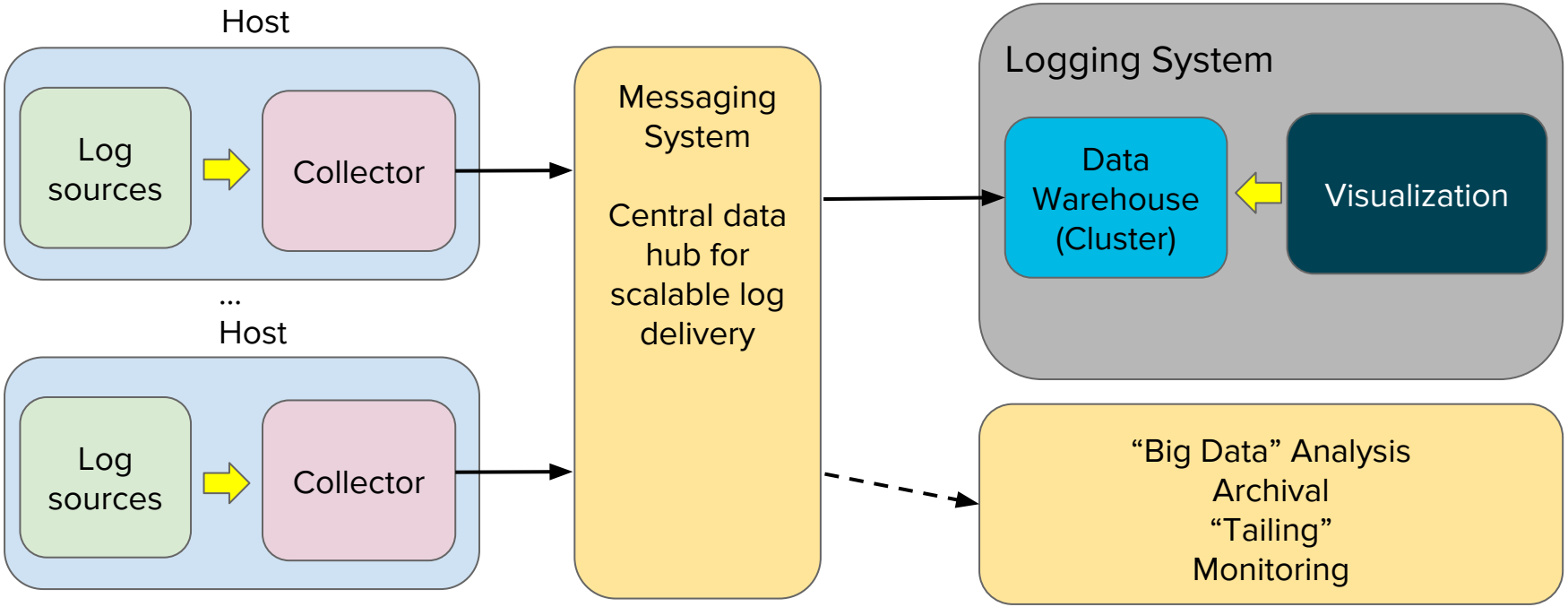
- The Origin Community Distribution of Kubernetes that powers Red Hat OpenShift
- Built around Kubernetes container cluster management, carries all k8s features, e.g.
  - Services, Pods, Controllers
  - Readiness & liveness probes
  - Persistent Volumes and Persistent Volume Claims
  - Release versions correspond to k8s releases
- Complete open source container application platform
  - Security and multi-tenancy
    - Access to namespaces per users/groups
  - Container image registry and source-to-image builds
  - CI/CD and devops workflows

Kafka

- A publish/subscribe messaging system
- Scalable and reliable data delivery
- Integrates well with other systems

Use Case

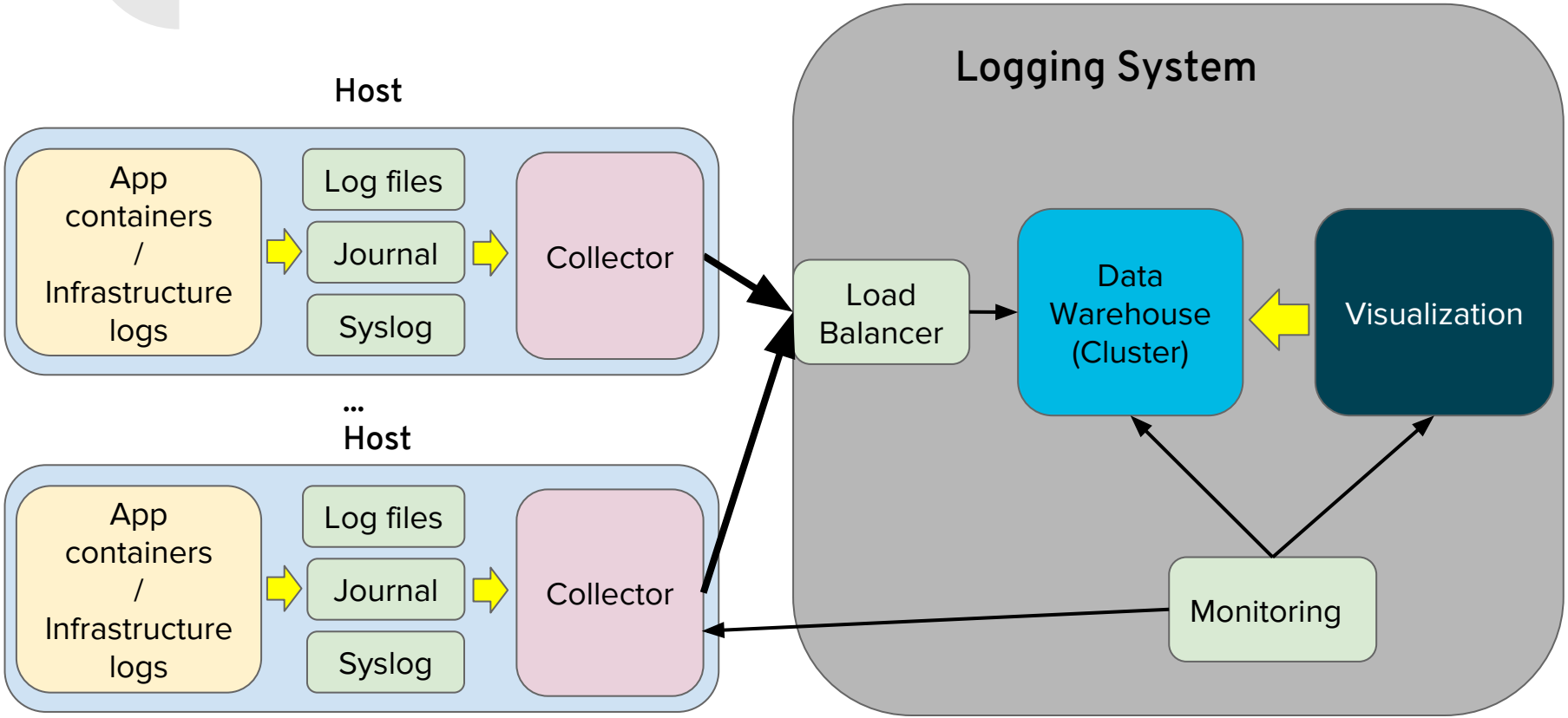
Integrate distributed Logging with Kafka



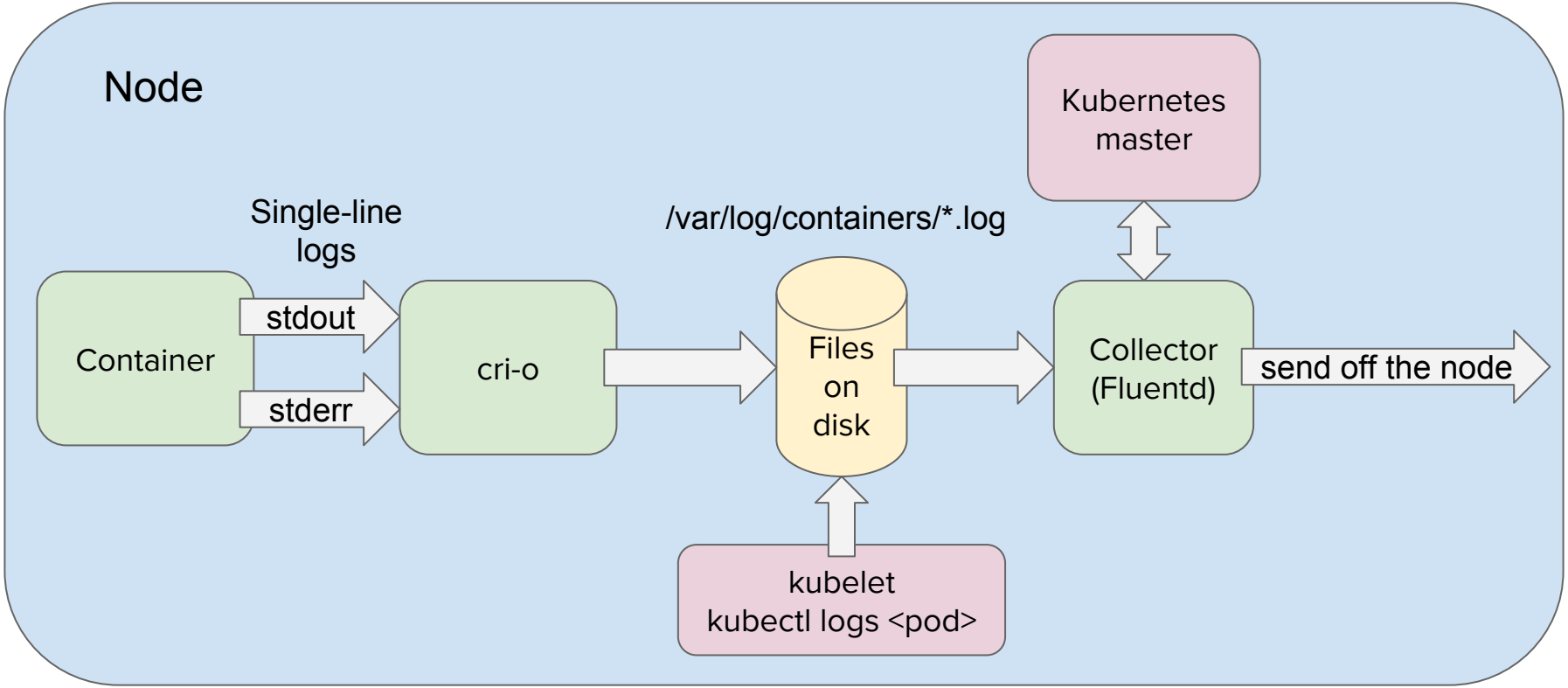
Origin Aggregated Logging

- Part of OKD
- Based on Elasticsearch, Fluentd and Kibana
- Collecting distributed logs
- Common data model
- Security model - multi-tenancy
- All open source

Logging Architecture

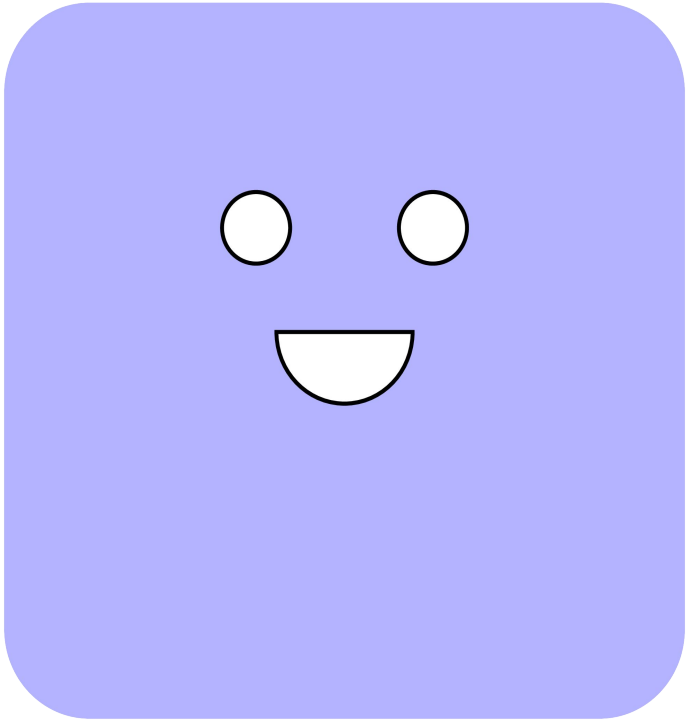


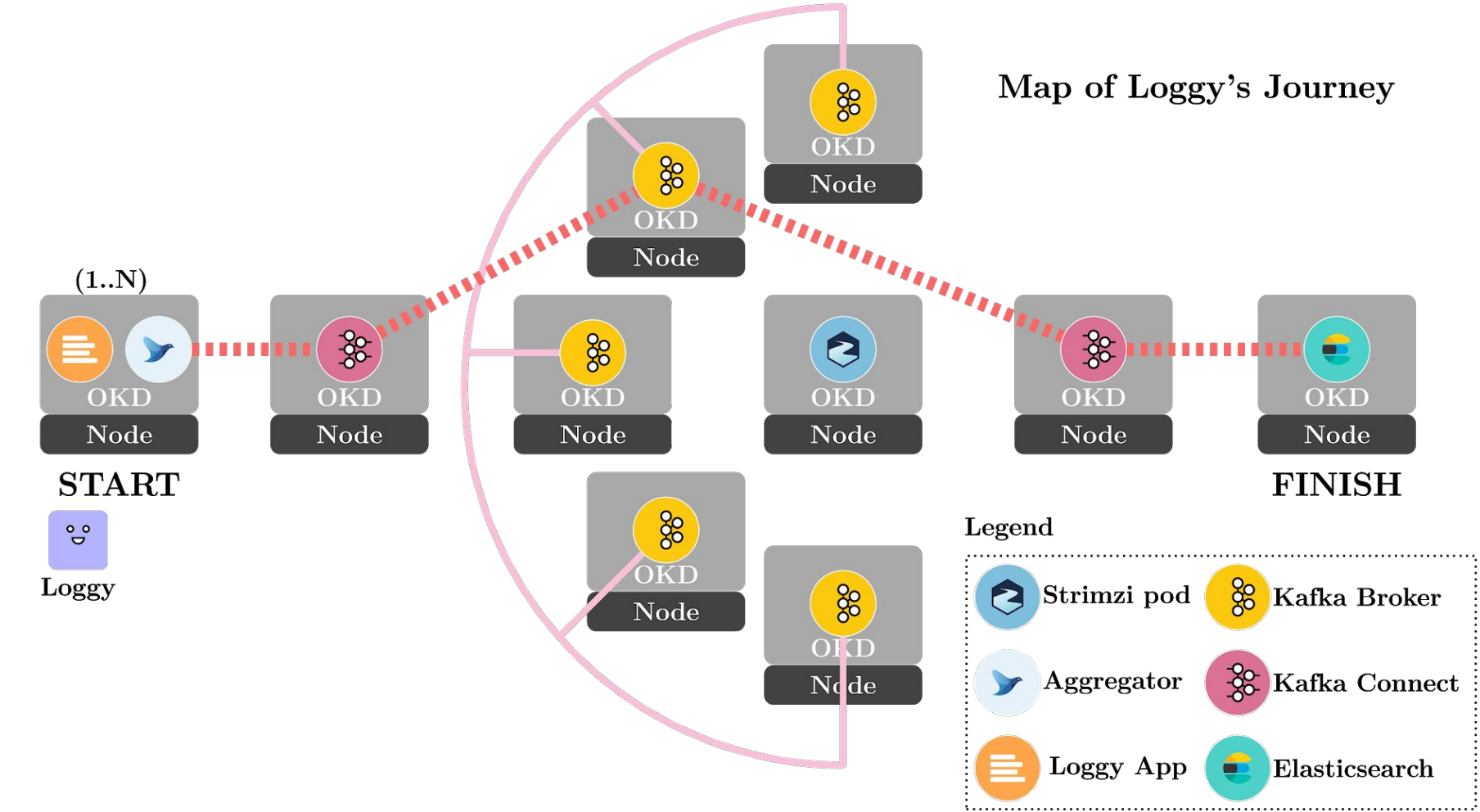
Container Logging



Loggy

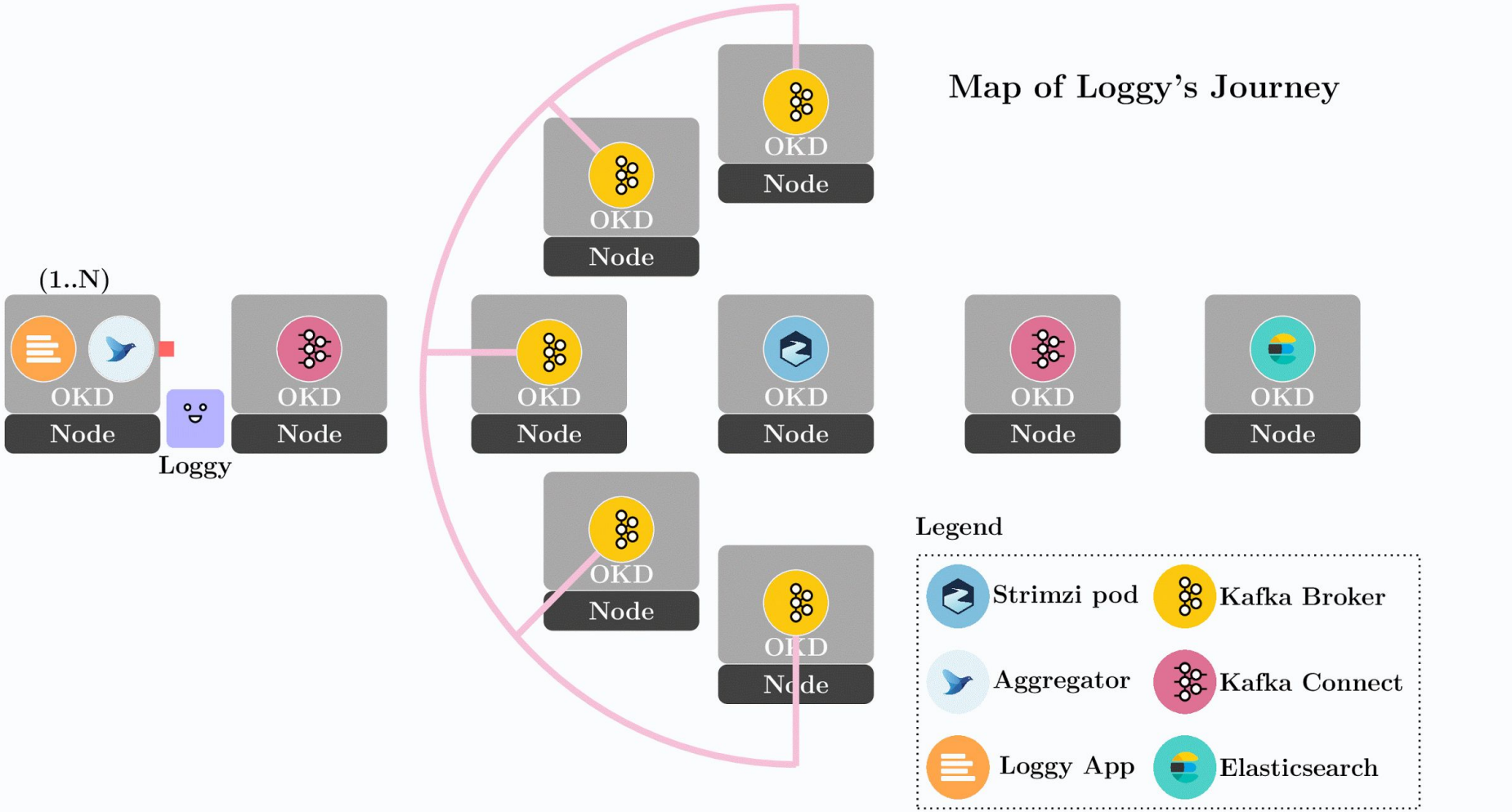
- Loggy is a log message
- Needs to get to work safely and reliably




Map of Loggy's Journey
(1..N)
OKD
Node
START
Loggy
FINISH
Legend
Strimzi pod
Kafka Broker
Aggregator
Kafka Connect
Loggy App
Elasticsearch

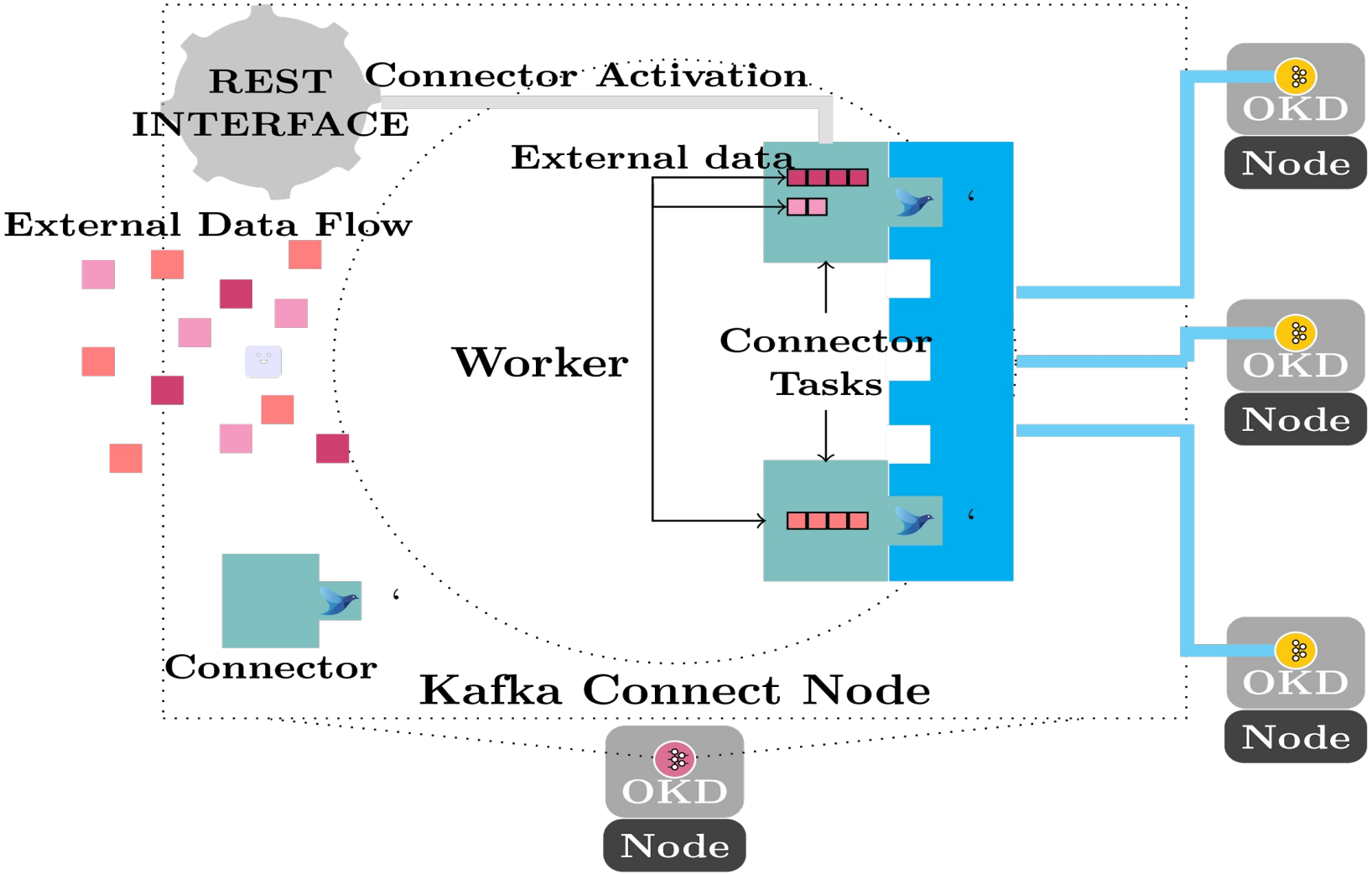
Log Collector

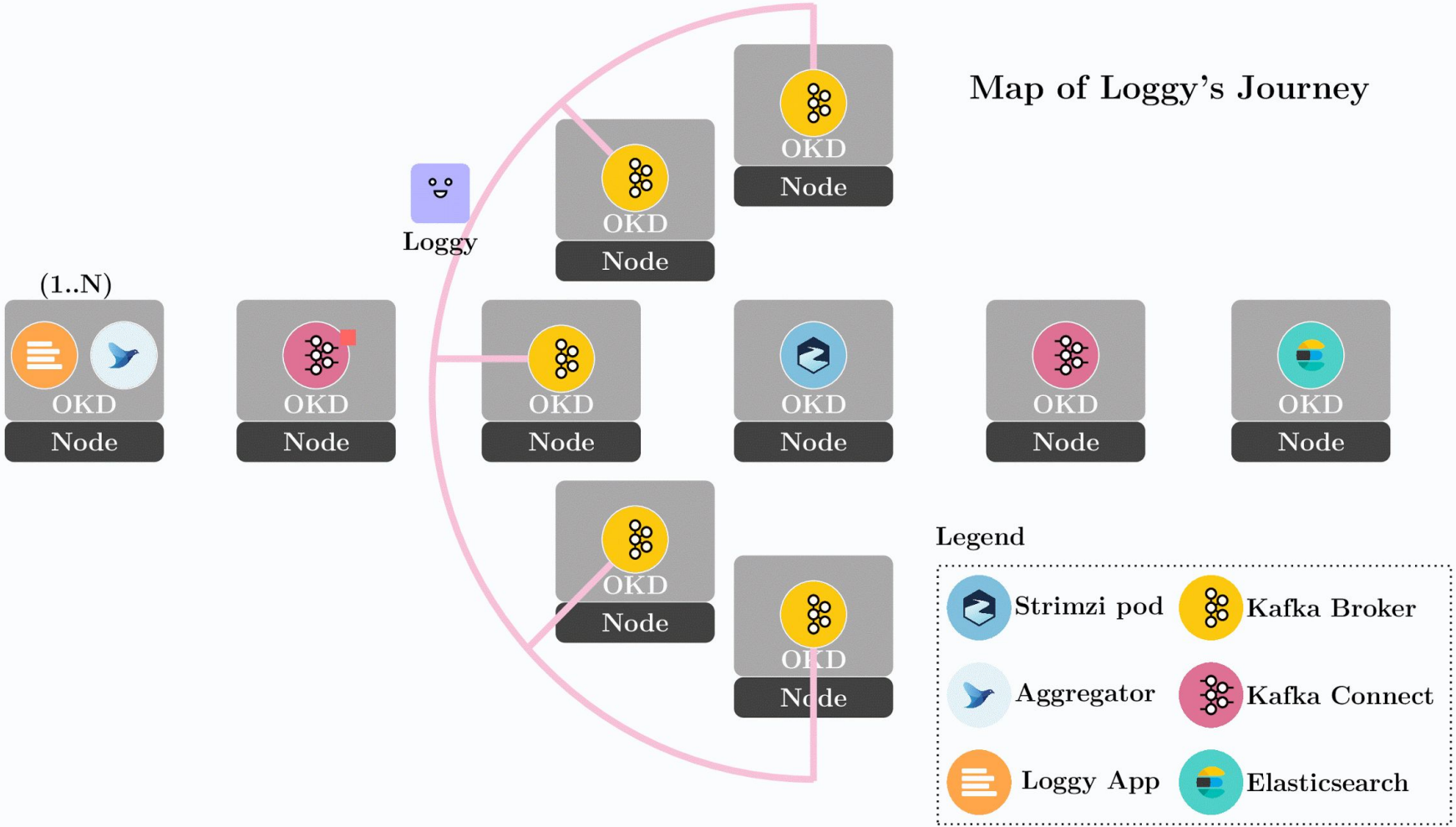
- Collects all container logs from node’s filesystem
- Ability to tag, filter and enrich logs for export
    - Add Kubernetes metadata to every log line


Map of Loggy's Journey
(1..N)
OKD
Node
Loggy
OKD
Node
OKD
Node
OKD
Node
OKD
Node
OKD
Node
OKD
Node
OKD
Node
OKD
Node
OKD
Node
Legend
Strimzi pod
Kafka Broker
Aggregator
Kafka Connect
Loggy App
Elasticsearch

Kafka Connect Source

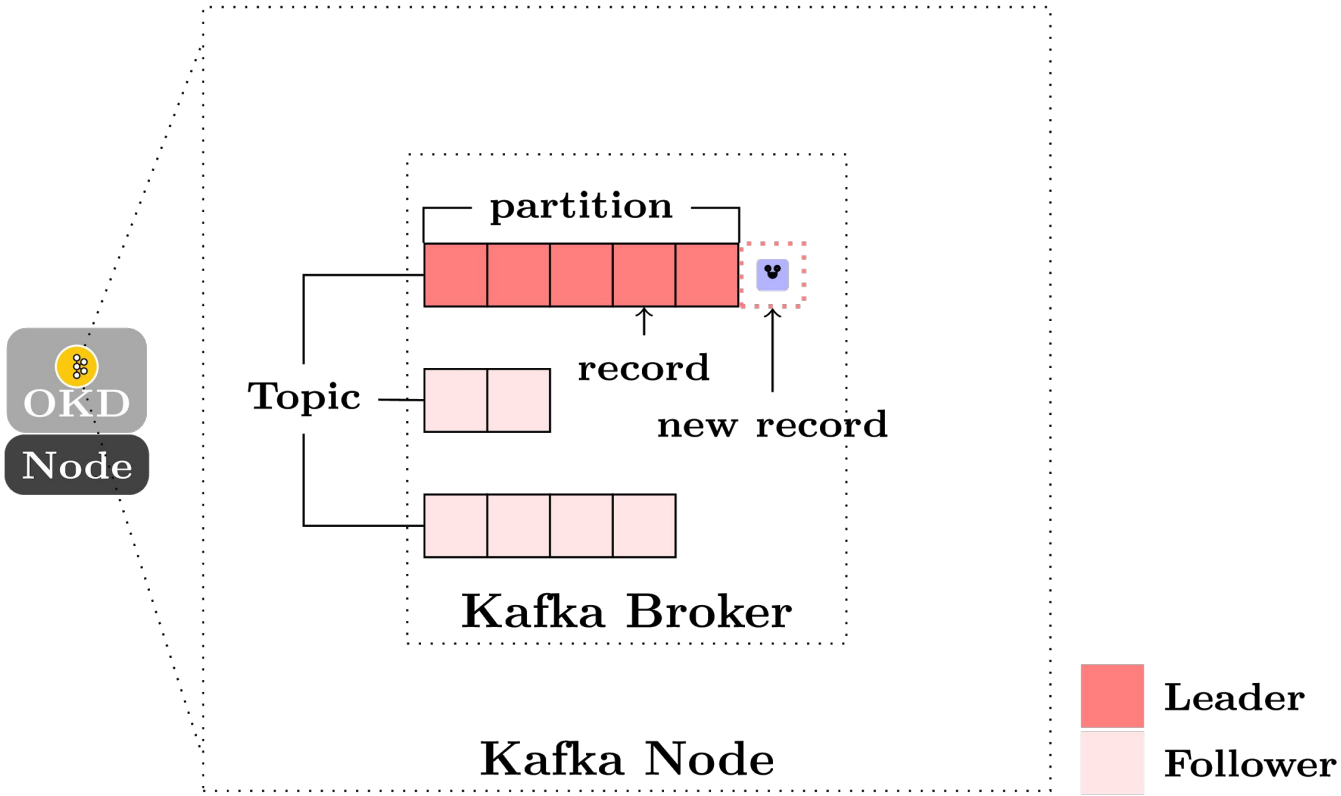
- Imports data from external systems into Kafka brokers
- Pluggable Connectors
- Rest Interface, Tasks and Workers


REST
INTERFACE
Connector Activation
External data
External Data Flow
Worker
Connector
Tasks
Connector
Kafka Connect Node
OKD
Node
OKD
Node
OKD
Node
OKD
Node


Map of Loggy's Journey
Loggy
(1..N)
OKD
Node
Legend
Strimzi pod
Kafka Broker
Aggregator
Kafka Connect
Loggy App
Elasticsearch

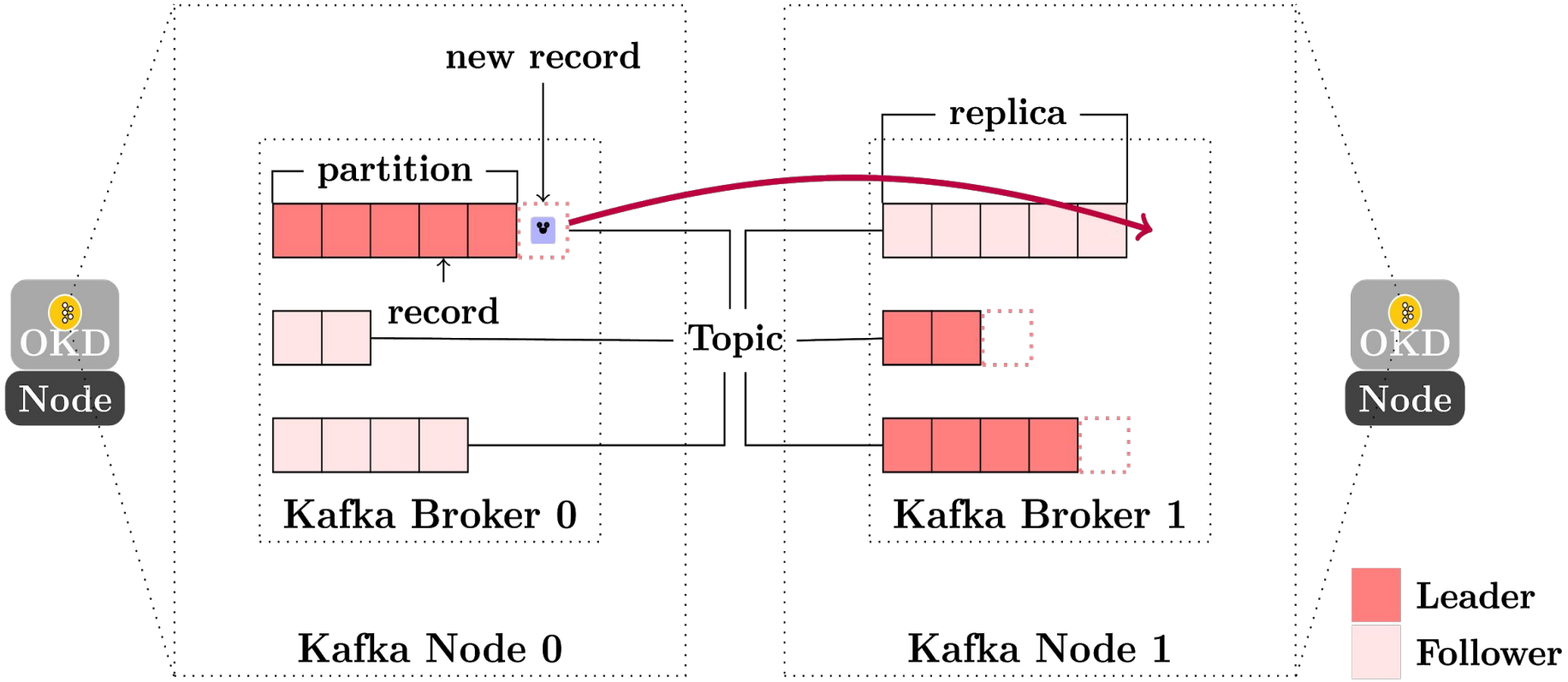
Kafka Broker

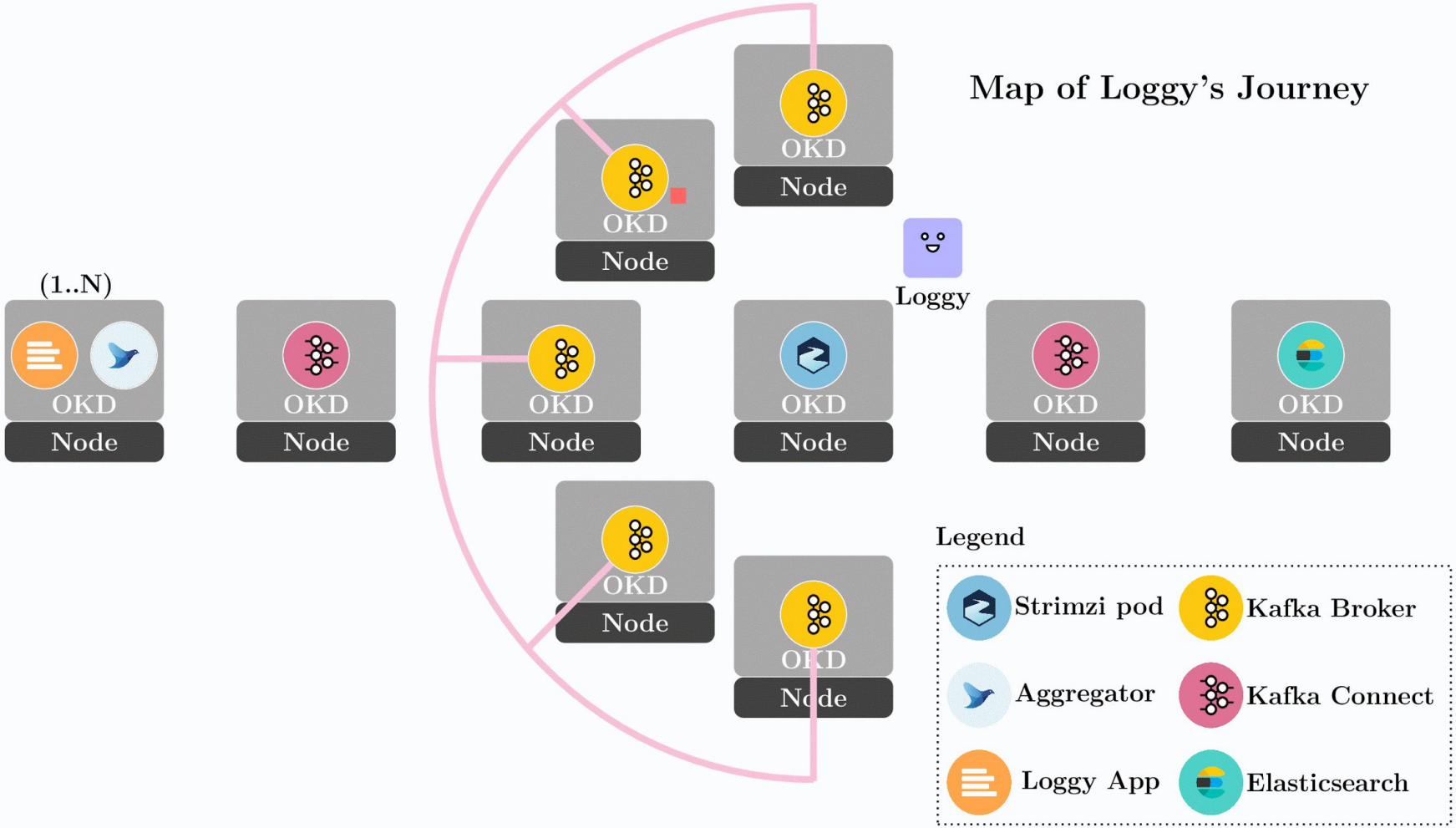
- Like a post box
- Topics, partitions and the distribution of load
- Smart clients / dumb brokers


OKD
Node
partition
record
new record
Topic
Kafka Broker
Kafka Node
Leader
Follower

Broker Reliability

- Replication
- Partition Leadership
- Performance/Reliability trade offs

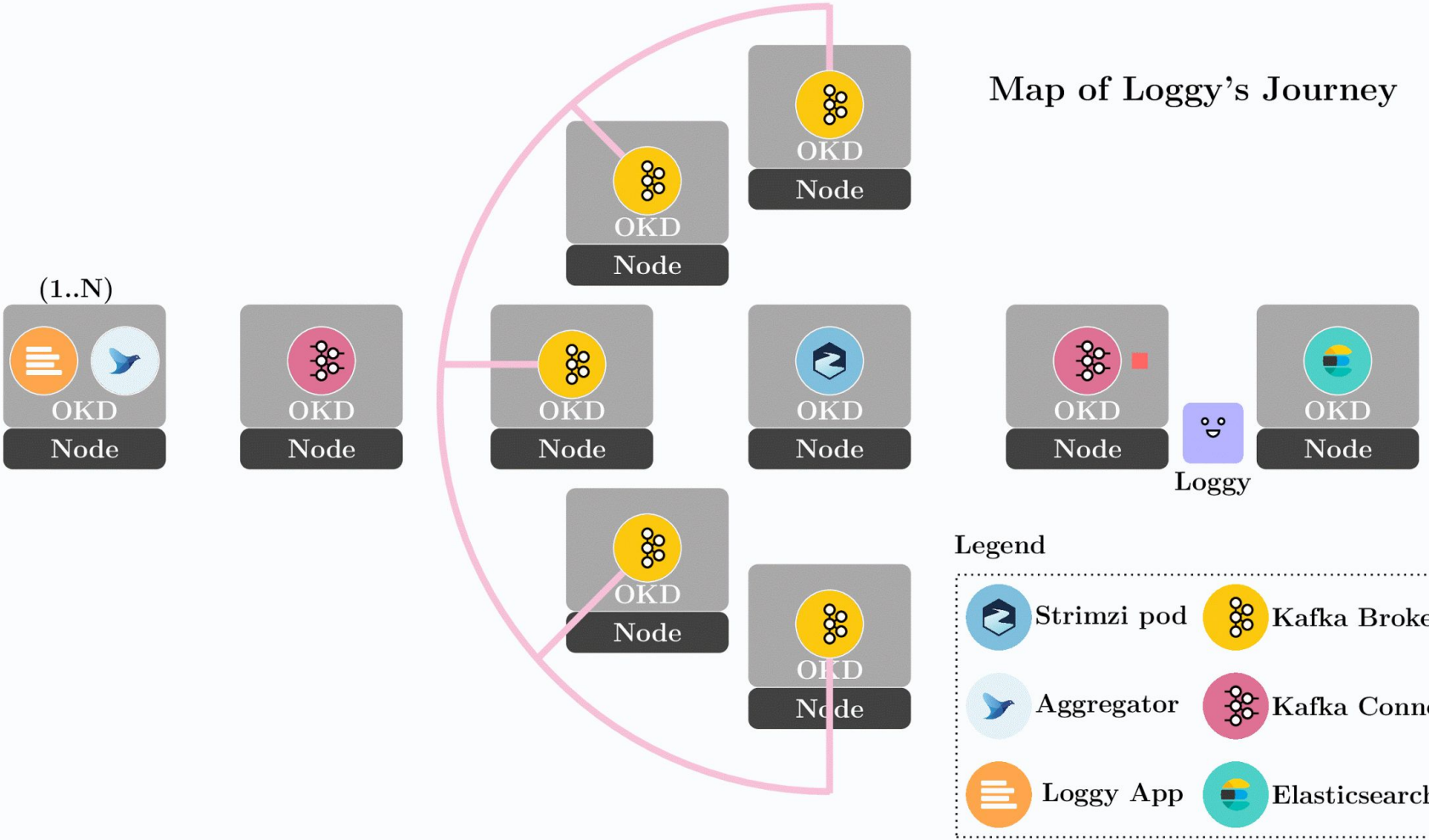

new record
replica
partition
record
Topic
OKD
Node
OKD
Node
Kafka Broker 0
Kafka Broker 1
Kafka Node 0
Kafka Node 1
Leader
Follower

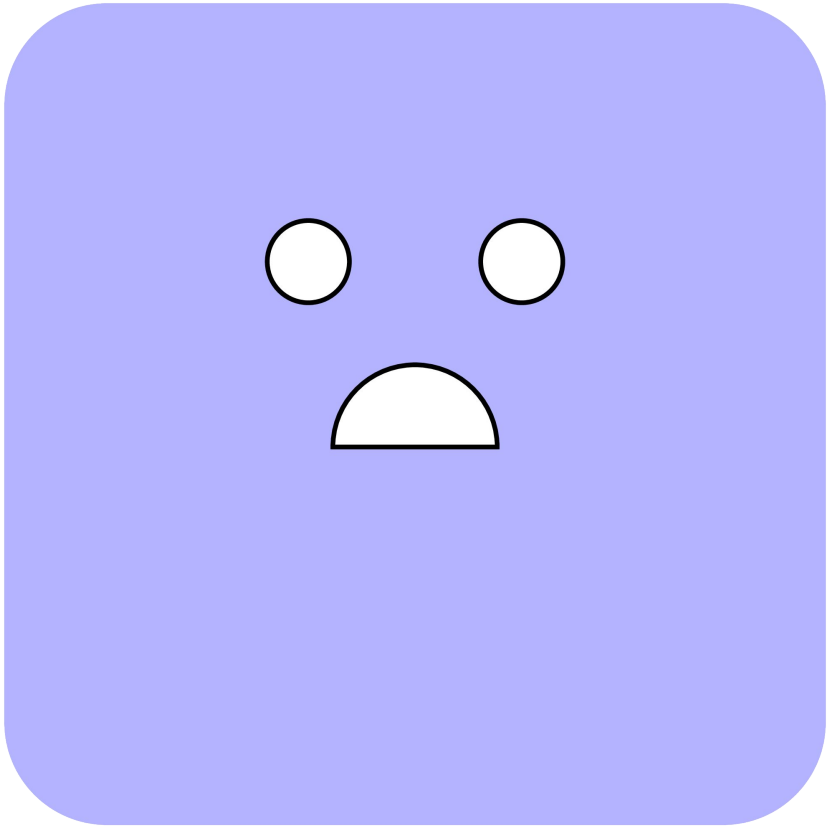

Map of Loggy's Journey
(1..N)
OKD
Node
Loggy
Legend
Strimzi pod
Kafka Broker
Aggregator
Kafka Connect
Loggy App
Elasticsearch

Kafka Connect Sink

- Exports Kafka broker data to external systems
- Distributed mode
- Convenience of Connect Framework

Map of Loggy's Journey



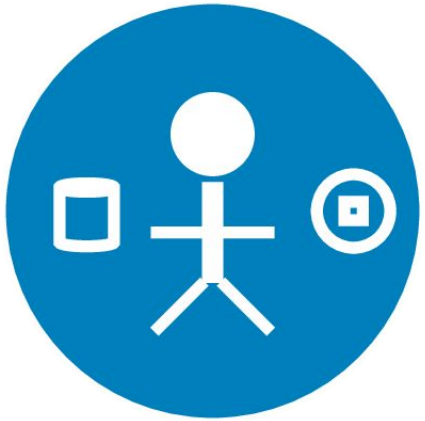


Strimzi Operators

- OKD/Kafka Integration
- Automates and manages Kafka deployment
- Operator Pattern

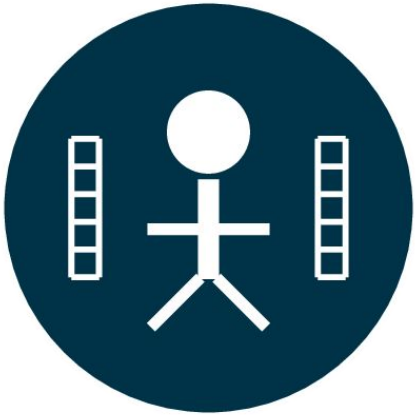
Cluster Operator

- Manages clusters:
  - Kafka
  - Kafka Connect
  - Zookeeper
  - Mirror Maker
- Advanced integration features



Entity Operator

- Manages different Kafka objects
- Currently consists of two Operators:
  - Topic Operator
  - User Operator



Custom Resources

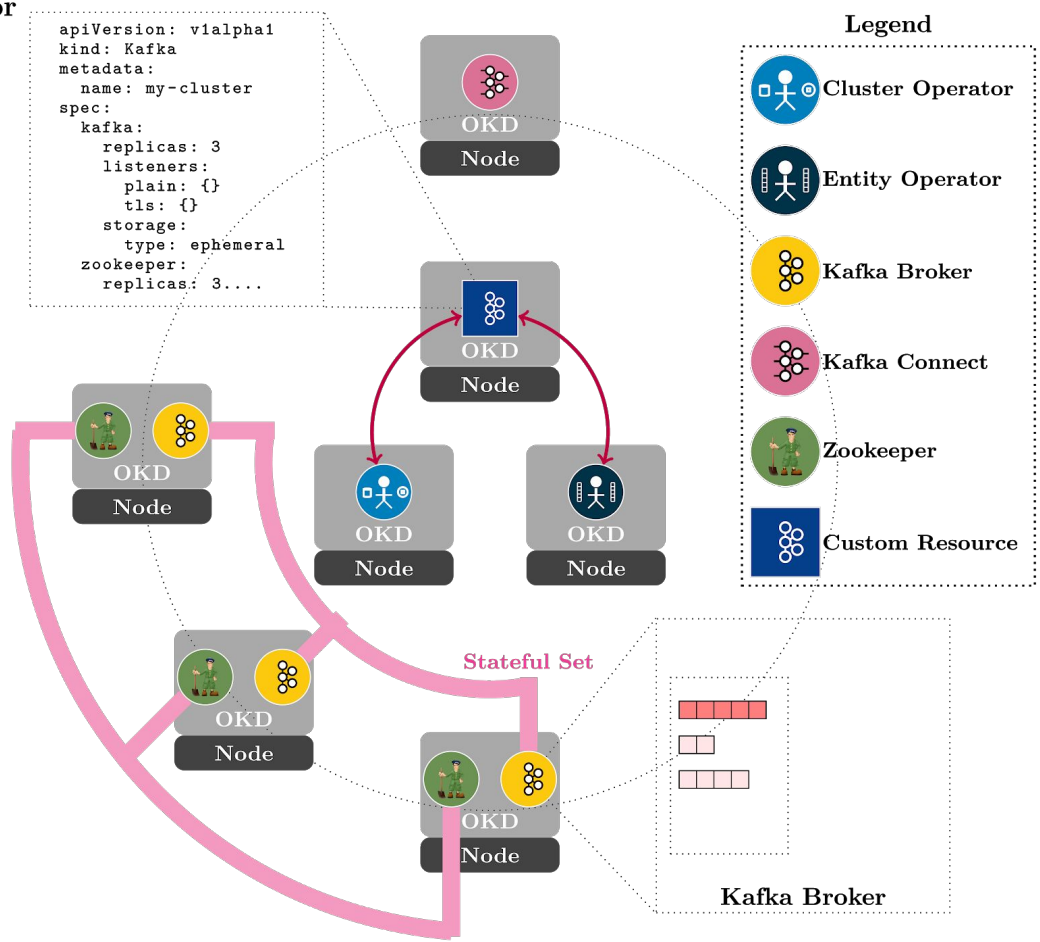
- Blueprint to describe Kafka cluster components
- Operators monitor these blueprints, matching cluster state with what is described

```
apiVersion: kafka.strimzi.io/v1alpha1
kind: Kafka
metadata:
  name: my-cluster
spec:
  kafka:
    replicas: 3
    listeners:
        plain: {}
        tls: {}
    config:
        offsets.topic.replication.factor: 3
        transaction.state.log.replication.factor: 3
        transaction.state.log.min.isr: 2
    storage:
        type: ephemeral
  zookeeper:
        replicas: 3
        storage:
          type: ephemeral
  entityOperator:
        topicOperator: {}
        userOperator: {}
```

Strimzi

Kafka Operator Architecture

```
apiVersion: v1alpha1
kind: Kafka
metadata:
  name: my-cluster
spec:
  kafka:
    replicas: 3
    listeners:
      plain: {}
      tls: {}
    storage:
      type: ephemeral
  zookeeper:
    replicas: 3....
```



DEMO

Image Credits

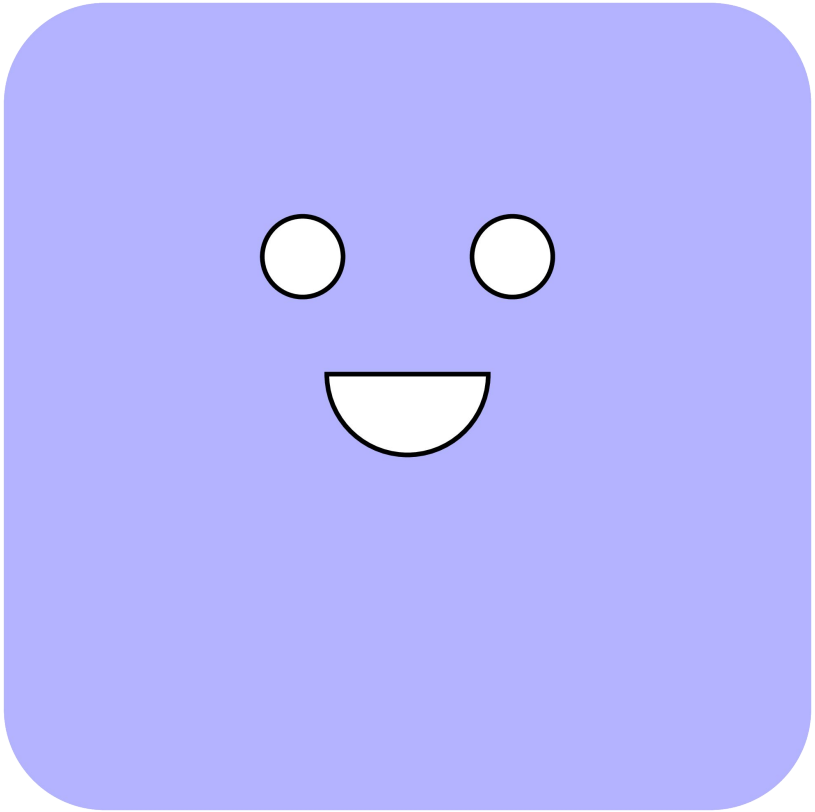
In order of appearance:

1. Maxmillion. (2013, January 4). NYC vs Boston [Digital image]. Retrieved October 15, 2018, from https://i.imgur.com/uffaY.jpg

Content Credit

- Narkhede, N., Shapira, G., & Palino, T. (2017). *Kafka - the definitive guide: Real-time data and stream processing at scale*. Sebastopol, Kalifornien: O'Reilly.
- Strimzi authors. (2018, October 15). Strimzi Documentation (Master). Retrieved October 15, 2018, from http://strimzi.io/docs/master/
- OKD authors. (2018, October 15). OKD Latest Documentation. Retrieved October 15, 2018, from https://docs.okd.io/latest/welcome/index.html

THANK YOU



EXTRAS

Cluster-level logging multi-tenancy